Analysis of Calcium Dynamics: Parameter Estimation Using Genetic Algorithm

N Ajith Kumar

(CH14MTECH11009)

A Dissertation Submitted to

Indian Institute of Technology Hyderabad

In Partial Fulfillment of the Requirements for

The Degree of Master of Technology

भारतीय प्रौद्योगिकी संस्थान हैदराबाद

Indian Institute of Technology Hyderabad

Department of Chemical Engineering

June, 2016

Declaration

I declare that this written submission represents my ideas in my own words, and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

N Ajith Kumar

CH14MTECH11009

Approval Sheet

This thesis entitled **Analysis of Calcium Dynamics: Parameter estimation using Genetic Algorithm** by N Ajith Kumar is approved for the degree of Master of Technology from IIT Hyderabad.

(Dr. Kishlay Mitra) Examiner
Dept. of Chemical Engineering
IIT-Hyderabad

(Dr. Saptarshi Majumdar) Examiner
Dept. of Chemical Engineering
IIT-Hyderabad

(Dr. Balaji Iyer) Examiner
Dept. of Chemical Engineering
IIT-Hyderabad

(Dr. Pankaj Kumar Kolhe) Examiner
Dept. of Mechanical and Aerospace Engineering
IIT-Hyderabad

(Dr. Lopamudra Giri) Adviser
Dept. of Chemical Engineering
IIT-Hyderabad

Acknowledgements

I thank almighty whose blessing has enabled me to accomplish my project work successfully.

It is my pride privilege to express my sincere thanks and deep sense of gratitude towards **Dr. Lopamudra Giri**, for her advice, supervision and patience through which this work was able to take a shape in which it has been presented. It was her valuable discussions and endless endeavours through which I gained a lot. Her constant encouragement has always been a moral support for me. Again I am very thankful to her for her time and engagement, and for sharing her knowledge with me.

A special word of thanks to my thesis evaluation committee Dr Kishlay Mitra, Dr. Saptarshi. Majumdar, Dr. Balaji Iyer, Dr. Pankaj Kumar Kolhe for their valuable suggestions.

It is hard for me to express enough gratitude to my seniors **Mr. Soumitri Srinivas** (Ph.D. Scholar) and **Mr. Santosh Kumar Varanasi** (Ph.D. Scholar) who have always been a source of inspiration for me and stood by my side at the toughest times.

On a more personal note, the whole credit of my achievements goes to my parents, who were always there for me in my difficulties. It was their unshakable faith in me that has always helped me to proceed further.

N Ajith Kumar

Dedicated to

My parents

Abstract

Live cell imaging of intracellular calcium is a cutting-edge tool used in drug design, delivery and screening. The aim of this study was to develop the mathematical model for the drug mediated (G protein coupled receptor targeting drug) intracellular calcium responses in fibroblasts and estimate the kinetic parameters. The current work proposes a computational framework for classification of heterogeneous data, model selection and parameter estimation using genetic algorithm (GA). Since the data is heterogeneous and large in size, we performed (1) reduction of the dimension using principal component analysis (PCA) and (2) classification of the calcium dynamics using K-means algorithm. Using PCA and K-means, the cell-to-cell variability was modeled as a mixture of three subpopulations (a) low amplitude (b) high amplitude-immediate (c) high amplitude-delayed responses. For model selection we formulated a series of models having various product formation kinetics, substrate inhibition and product inhibition kinetics. Then we used the hybrid algorithm for model selection through minimization of the error between the experimental and the simulated calcium profiles. Hybrid of two optimization techniques, GA and gradient-based method was used, which takes advantage of both the techniques. In this hybrid algorithm, GA provides the initial guess values for gradient-based method. Using this method we found that the Michaelis Menten kinetic model provides a satisfactory agreement with the experimental data, whereas the adoption of detailed models leads to negligible improvements of the fit. Moreover we found that any of the model having Michaelis Menten kinetics, Hill kinetics, inclusion of substrate inhibition, product inhibition or exponential delay mechanisms were not able to capture the delayed response.

Contents

Chapter 1

Introduction

G-protein coupled receptors are the target receptors for almost 45% of the drug in current drug market. For drug screening using cell-based assays, fluorescent imaging of calcium dynamics cell-population can be used to obtain the dose-response profile. However, construction of a dose-response function based on single cell responses is rather challenging as the cells in a population respond heterogeneously to the drug. Here we develop a computational framework for identification of the mathematical model based on kinetic mechanism and estimate kinetic parameters corresponding to the heterogeneous calcium responses in HeLa cells. The temporal dynamics of cytosolic calcium was measured through time-lapse imaging using confocal microscopy for various drug doses (GPCR targeting drug).

1.1 Calcium Imaging Experiment

Figure 1.1 shows the calcium responses in a cell population and in this experiment, 400 ng drug is added to each and every cells. As the calcium response cannot be measured directly, we added Fluo-4 reagent (fluorescent sensor that binds to calcium). Intensity of the Fluo-4 with respect to time at different cell locations were noted. Since calcium concentration is proportional to the intensity of the Fluo-4, we have plotted intensity with respect to time at different cell locations.

As we can see from the Figure 1.1, some cells brighter (higher intensity) indicating that the cells are responding to the drug. The Figure 1.2 shows the experimental plot of calcium concentration vs. time of the cells for 400 ng drug dose.

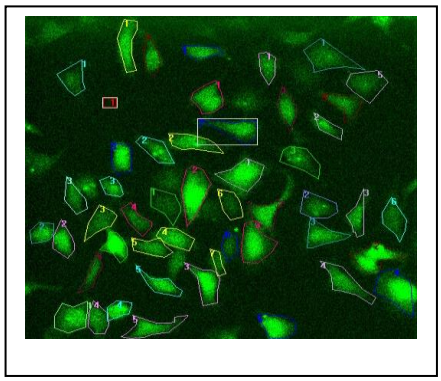


Fig 1.1: Confocal imaging of intracellular calcium in HeLa cell population (400ng/mL drug)

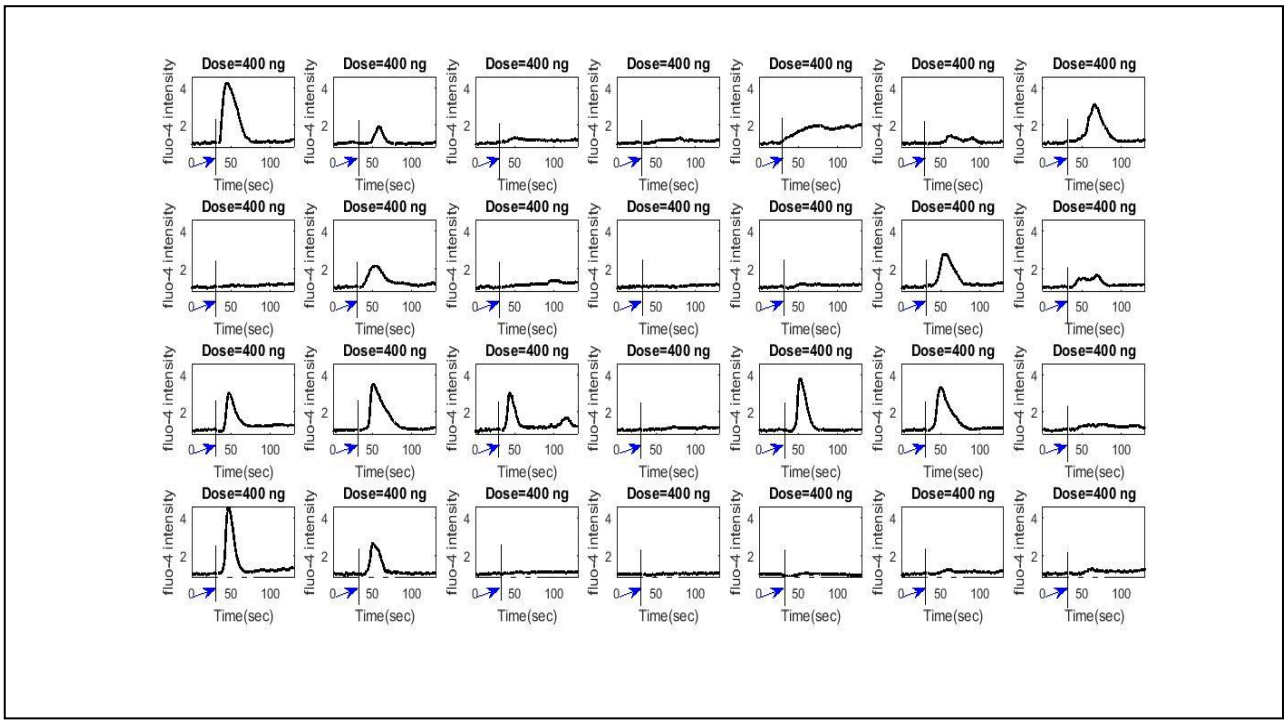


Fig 1.2: Fluo-4 intensity (proportional to calcium concentration in cells) vs. time in a cell population (Drug dose=400 ng/mL drug dose)

1.2 Existing challenges:

The specific challenges in analysis of intracellular calcium dynamics are as follows

1. To handle large amount of dynamic data from the imaging experiments (handling large amount of videos)
2. To identify a mathematical model for calcium responses for each and every cell which can predict the time course of calcium concentration (since very little information is available on the kinetic mechanisms of the intracellular reactions and reaction network)
3. To propose a general model for the cells with various drug doses
4. To estimate the kinetic parameters of the proposed mathematical model (it is challenging to obtain these kinetic parameters experimentally)

Instead of proposing a general model for the whole population or finding a kinetic model for each and every cell, the models can be proposed for specific type of cell responses. In order to perform the classification, we performed dimension reduction by principal component analysis. And we used K-means clustering to divide the population in three subpopulations.

The estimation of the kinetic parameters in the biological models is often formulated as an optimization problem. Hence, we defined the objective function of the problem as the difference between the model outputs/response produced from the simulation (using kinetic parameters) and the respective experimental measurements. As a result, the solution, which was formed from the combination of kinetic parameter sets, generate the model output that closely fit the experimental measurements.

Chapter 2

Literature Review

Biological systems typically consist of large numbers of interacting components and involve complex processes at a variety of spatial, temporal and biological scales [1]. The key part of the systems biology approach is mathematical modelling and it can be used to produce composite models which describe systems across multiple scales.

To understand the kinetic mechanism of drug-cell interactions, computational modeling will play an important role. Generally the biological kinetic models are constructed using a set of coupled differential equations, mostly by using ordinary differential equations (ODEs), to signify the reactions in a specific range of time intervals. The models heavily rely on a set of parameters such as reaction rates and transportation rates that characterize the physiological behaviors of system. It is challenging to find these kinetic parameters through experimental analyses. Hence, these kinetic parameters are rather approximated based on the given experimental measurements. In most cases, the nonlinear least squares techniques (in finding root mean square error, RMSE) are used to find optimal parameters that may produce model outputs which fit closely to our experiment measurements. This task is usually hampered by the nonlinearity of the systems as well as the incompleteness of the available experimental measurements [2].

2.1 Reaction Mechanism and kinetic model

The model-building process generally starts with (1) expert proposing a model (2) fitting the model to the data and (3) changing the model if the predictions are not satisfactory. This is a knowledge-intensive, time-consuming and iterative process [3]. Fitting the model to experimental data involves searching for the model parameters that accurately describe the data as defined by an error criterion. The error is generally defined as the sum of the squares of the differences between the model predictions and the experimental data. This step in modelling, known as parameter estimation, is critical, because wrong conclusions may be reached if a set of model parameters that are able to describe the data are available but are

not identified. Parameter estimation is complicated by the complexity and non-linearity of the model, quality of the data, the lack of tight bounds on the parameter search space and the lack of generic tools that can cater to a wide range of models [3].

We have assumed a reaction mechanism based on our experimental data as shown in Figure 2.1 and proposed a kinetic model. From Figure 2.1 we conclude that the experimental data and the simulated model (blue curve) looks similar. Hence we plan to set a framework for parameter estimation (kinetic parameters) using optimization techniques.

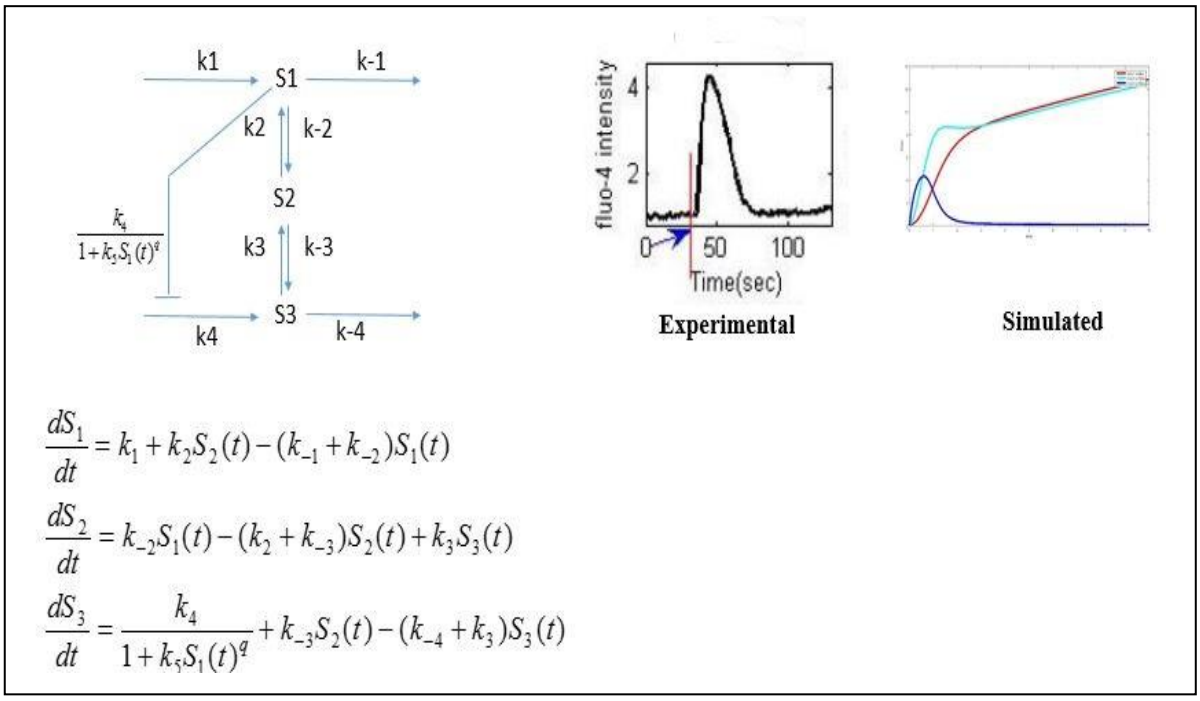


Fig 2.1: Reaction Mechanism and kinetic model for simulation of calcium response

.

2.2 Optimization methods

2.2.1 Genetic Algorithm:

Genetic algorithms (GA) were first proposed by John Holland in 1975[4]. GA's are quite promising as a stochastic global optimization method. These algorithms are based on the evolutionary ideas of natural selection and genetics. It is the most popular method used in the parameter estimation problem. GA requires only information concerning the quality of the solution and does not need linearity of the parameters.

The Genetic algorithm selects individuals at each step in a random manner from the current population of species parents and uses them to produce new generations for the next generation. Over successive generations, the population evolves toward an optimal solution. In each step, genetic algorithm uses three types of operators to create the next generation of the population: selection, crossover, and mutation.

Selection operator: The selection is based on the adaptation of individuals. It is carried out by choosing pairs of individuals from one generation to another and those involved in the reproduction process of the future population. A certain percentage of the population size is maintained from one generation to another called elitism. Elitism involves copying a small proportion of the fittest candidates, unchanged, into the next generation. This can sometimes have a dramatic impact on performance. Candidate solutions that are preserved unchanged through elitism remain eligible for selection as parents when breeding the remainder of the next generation.

Crossover operator: Crossover (like selection) is a convergence operation which is intended to pull the population towards a local minimum/maximum. The crossing is applicable to two individuals drawn randomly from a population above the current population. These two individuals are mated to give birth to two other offspring's. Despite the randomness, this exchange of information gives genetic algorithms power in their work: sometimes "good" genes from one parent will replace the "bad" genes and create another offspring better adapted to the environment. Generally the crossover probability will be 0.7-0.85.

Mutation operator: Mutation is a divergence operation. It is intended to occasionally break one or more members of a population out of a local minimum/maximum space and potentially discover a better minimum/maximum space .The mutation operator for all these individuals was generated in the new population.. It serves to emulate the natural phenomenon whereby offspring sometimes happen to have genes with totally different characteristics from their parent [5] because of errors due to various factors.

Advantages of GA:

- No prior information is needed about search space.
- Excellent global search capability (able to do multi-prolonged population based search)

Disadvantages of GA:

- Weak local search capability
- Suffer from slow convergence speed

The flowsheet of GA is shown in Figure 2.2.

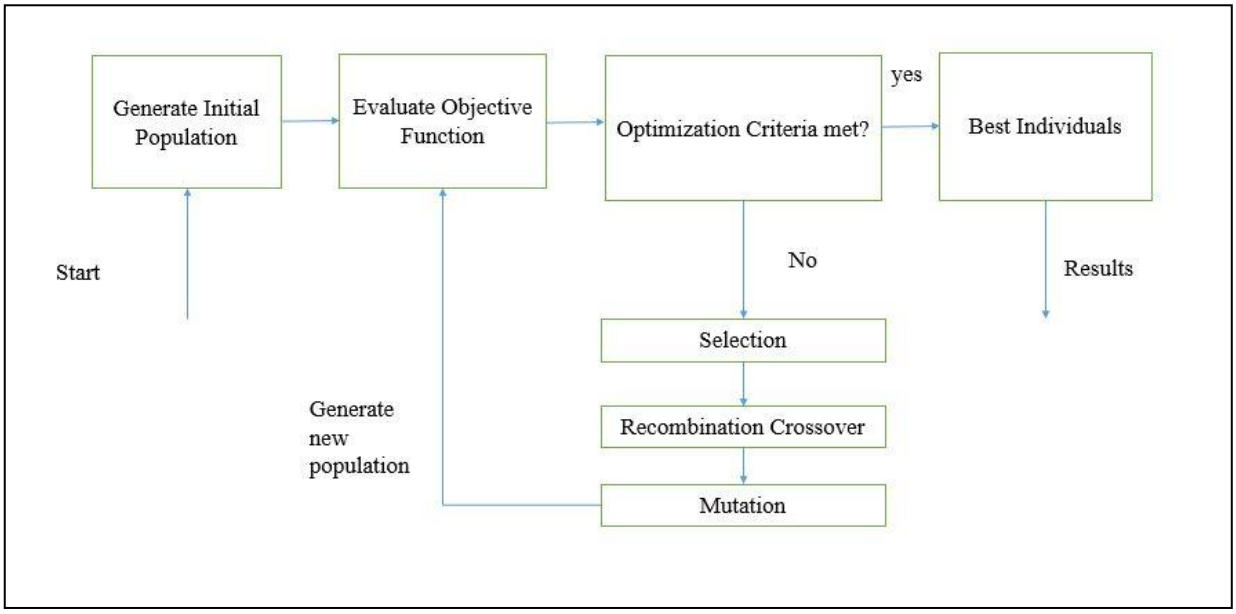


Fig 2.2: Flowsheet of genetic algorithm

2.3 Hybrid Method for optimization

A hybrid algorithm for optimization takes advantages of both genetic algorithm and gradient based search method. In this GA is used to get local minima and by using this one we can use gradient based method to ensure that global minima is reached. The GA based hybrid procedure identifies the most promising regions of the parameter search space [3]. The best solutions from GA in every generation and also the members of the final generation are then used as the initial guess values for the local optimizer based on gradient-based method (fmincon in MATLAB, combining steep descent method and Newton method).

To overcome the drawbacks of GA, based on the mechanism of the biological DNA, RNA genetic algorithm was proposed to estimate the parameters in chemical engineering processes [6]. They encoded the chromosomes with nucleotide bases and GA operators are modified with RNA molecular operations. In this algorithm, they first encode each individual with a strand of nucleotide bases, RNA strand. Then instead of cross over operators to improve the performance of GA, RNA-recoding operator and protein-folding operators are designed in RNA-GA. Apart from encoding procedure, RNA strands are first translated into amino acids ones, protein strands, according to triplet codons in decoding procedure [6].

2.4 RNA-GA:

RNA contains 4 kinds of nucleotides: Adenine (A), Uracil (U), Guanine (G) and Cytosine (C)[6]. In genetic code, a three-letter codes, triplet codon, decides an amino acid, i.e., three nucleotides in RNA strand decide an amino acid by the translating operator. On the decoding procedure of the PIRGA (Protein inspired RNA GA), RNA strands are first translated into amino acid strands, protein ones [6].

Procedure of the RNA-GA

Step1:.Initilizing the population (N individuals)

Step2: Calculating objective function (fitness value) at each population divide population into two groups

Step3: Select N individuals to combine mating pool with the selection strategy.

Step4: Check whether or not meets the condition of the RNA-recoding operator. If yes then execute the RNA-recoding operator and go to Step 6. Otherwise, go to Step 5.

Step5: Check whether or not meets the condition of the protein mutual-folding operator. If yes then execute the protein mutual-folding operator and go to Step 6. Otherwise, carry out protein self-folding operator and go to Step 6.
Step6: Executing mutation operator (adaptive probability).
Step7: Repeat the Steps from 2–6 until termination conditions are met and the final solution is found.

Advantages of RNA-GA

- Can improve diversity of the population
- Can able to overcome fraudulence compared with GA[6].

Disadvantages of RNA-GA:

- Sacrifices the speed of convergence to obtain diversity in population.
- Not applicable for high-dimensional optimization problems.

To overcome the deficiencies of RNA-GA, a DNA based GA (DNA-GA) was proposed[7]. In this algorithm, they encoded each individual with a sequence of nucleotide bases. Then, inspired by the operations of DNA molecular, they designed genetic operators to enhance the global searching ability of the DNA-GA. Simulation studies on six benchmark functions [7],varying from two-dimensional to high dimensional, show the superiority of the DNA-GA in contrast to other two algorithms, RNA-GA and GA as shown in Figure 2.3a and Figure 2.3b.

2.5 The DNA Genetic Algorithm

Binary data is encoded with 0 and 1, DNA is encoded with nucleotides which is of four type's adenine (A), guanine (G), cytosine (C), and thymine (T)[7].

Genetic operators

There are three types of genetic operators for DNA-GA. They are crossover operator, selection operator and three mutation operator consisting of inverse-anticodon operator, maximum-minimum operator, and normal-mutation operator [7]as shown in Figure 2.4.

Test functions.

Test functions	Optimal solution	Optimal value
$\min f(x) = 1 + \left((x_1 - 100)^2 + (x_2 - 100)^2\right)/4000 - \cos(x_1 - 100)\cos\left((x_2 - 100)/\sqrt{2}\right),\ x_i \in [-600, 600]$	(100, 100)	0
$\min f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2,\ x_i \in [-5.12, 5.12]$	(1, 1)	0
$\max f(x) = \left(\frac{a}{b+(x_1^2+x_2^2)}\right)^2 + (x_1^2 + x_2^2)^2,\ a = 3,\ b = 0.05,\ x_i \in [-5.12, 5.12]$	(0,0)	3600
$\min f(x) = 0.5 + \frac{\left(\sin\sqrt{x_1^2+x_2^2}\right)^2 - 0.5}{\left[1+a(x_1^2+x_2^2)\right]^2},\quad x_i \in [-10, 10]$	(0,0)	0
$\min f(x) = -c_1 \exp\left(-c_2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\cos\left(\sum_{i=1}^{D} c_3 x_i\right)\right) + c_1 + \exp(1),$ $x_i \in [-32.768, 32.768], i = 1 : D, D = 10, c_1 = 20, c_2 = 0.2, c_3 = 2\pi$	(0,0...0)	0
$\min f(x) = \sum_{i=1}^{n} x_i^2,\ n = 10,\ x_i \in [-5.12, 5.12]$	(0,0...0)	0

Comparison of efficiency and reliability ability of three algorithms.

Test function	DNA-GA		RNA-GA		GA	
	CPU time(s)	Suc (%)	CPU time(s)	Suc (%)	CPU time(s)	Suc (%)
f_1	0.61	100	3.75	72	5.04	44
f_2	1.8	100	3.01	100	5.61	76
f_3	1.14	100	4.23	100	3.37	74
f_4	0.54	100	6.58	62	3.3	46
f_5	3.1	100	26.24	0	7.46	88
f_6	1.21	100	19.97	100	2.02	100

Fig 2.3: a) Six Benchmark Test Functions (objective functions) to test the performance of optimization algorithm b) Comparison of efficiency of three algorithms DNA-GA, RNA-GA and GA

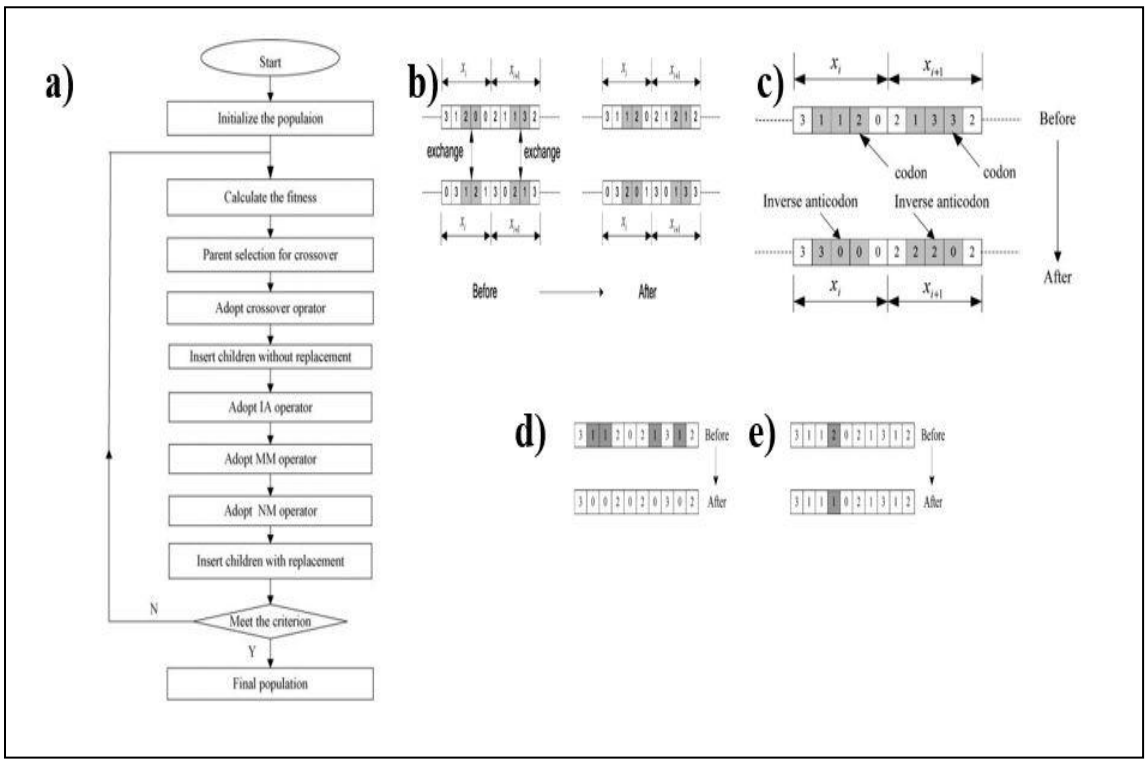


Fig 2.4 : **a)** Flowsheet of DNA-GA algorithm **b)** crossover operator **c)** IA operator **d)** MM operator **e)** NM operator

Procedure of the DNA-GA

Step 1: Initializing the population (N individuals)

Step 2: Calculating objective function (fitness value) at each population

Step 3: Select two individuals from the population randomly as the parents and adopt crossover operator over the parents to generate new individuals. Repeat this step until N/2 new individuals are created.

Step 4: Insert all the new individuals generated in step 3 into the population without deleting old individuals.

Step 5: Adopt three mutation operators orderly over each individual, and generate 3/2N new individuals.

Step 6: Replace all the original individuals with the new ones produced in step 5.

Step7: Apply elitism in conjunction with tournament selection to choose N individuals from the population for advancing into the next generation.

Step 8: Repeating steps 2–7 until the stop criteria are met, and the final solution is found.

Advantages of DNA-GA:

- Convergence speed is superior compared to GA, RNA-GA.
- Probability to converge to global minima is high.
- Applicable to high dimensional optimization functions.

Chapter 3

Model Selection

3.1 Data Reduction Technique

We need to analyze the time series calcium response for a large number of cells and the dimension of our experimental data is large. So we have reduced the dimension through implementation of principal component analysis (PCA). The flowchart for PCA is shown in Figure 3.1.

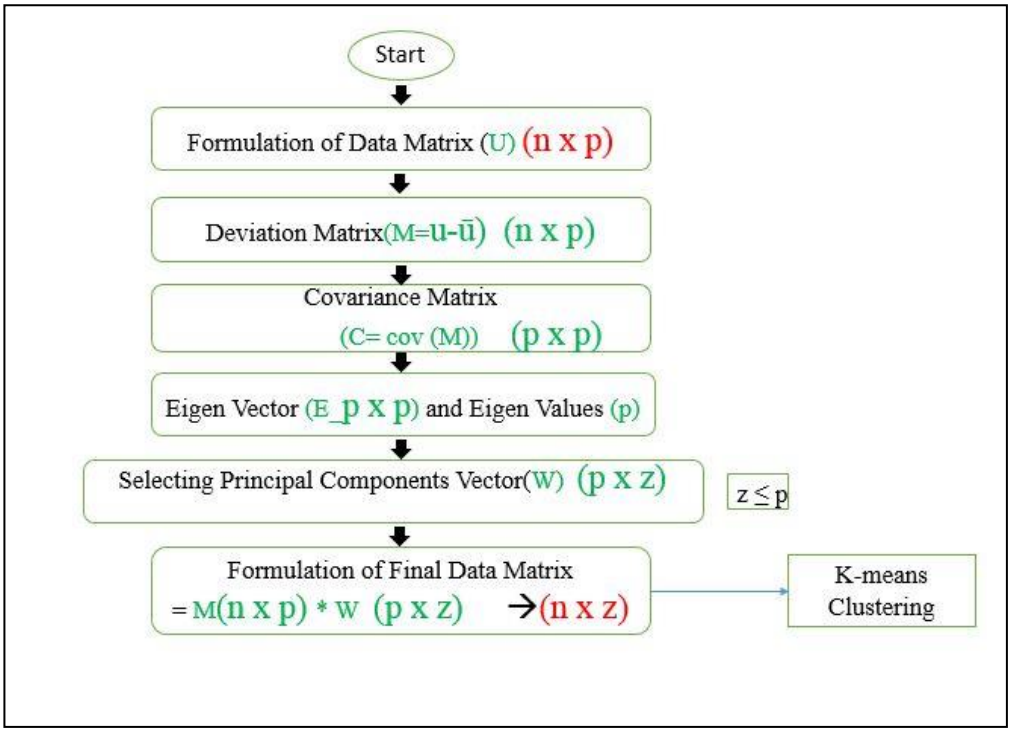


Fig 3.1: Flowsheet for principal component analysis

Principal component analysis: This method is used for reduction of data dimension and feature extraction through creating a new set of variables called principal components. Each of the new variables is a linear combination of the original variables. Each of the principal components is chosen so that it would describe most of the data features. All the principal components are orthogonal to each other [8]; and hence there is no redundant information. The first principal component has the maximum variance among all possible choices.

Our experimental data is of 119*28 dimension (time*cell).We have reduced the dimension in time direction. First we find the deviation matrix i.e. how much the data is deviating is from the mean [8]. Then we find the covariance matrix which is of 119*119 dimension. Next we find the Eigen values and corresponding Eigen vectors. Now the maximum variance data is selected by selecting Eigen vectors corresponding to highest Eigen values (2). Now the deviation matrix is multiplied by Eigen vector matrix to form new data matrix of 28*2 dimension. So finally the data matrix of dimension 28*119 is reduced to 28*2. The overall flowsheet of my project is shown in Figure 3.2:

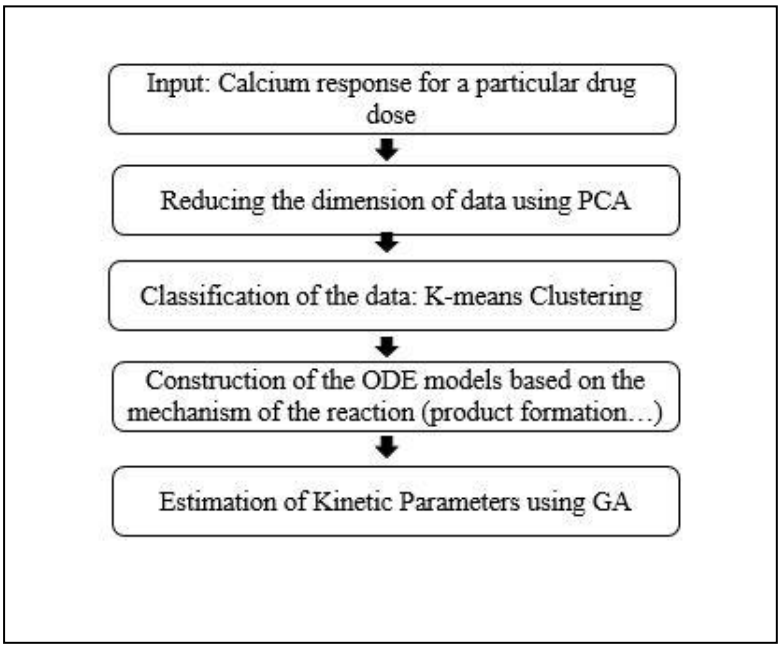


Fig 3.2: Workflow for data classification, model selection and parameter estimation

3.2 K-means Clustering

K-means is a technique for classification of the objects based on attributes/features into K number of groups where K is positive integer number. The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid. Thus, the purpose of K-mean clustering is to classify the data. The output data from PCA is given as input to the K-means algorithm. The input to the k-means is given as to classify the data into three types of clusters (K=3). After clustering, the data was divided into three types of responses a) fast response, cluster 1 (green) b) flat response, cluster 2 (blue) and c) delayed response, cluster 3 (red). Hence we plan to fit three models for each cluster so that a combination of three models with various parameters can be used as the predictive framework.

3.3 Model Selection

We selected one dataset for calcium response from cluster 1 and assumed a mechanism [9] to describe that cell data as shown in Figs 3.3 and 3.4 : In the Figure 3.3, X represents drug, after addition of the drug it binds to the receptor to form S1. Where, S1 acts as catalyst to the next reaction to form S2 which in turn acts as a catalyst to form S3. S3 is assumed to be calcium concentration. We proposed a set of models assuming (1) first order kinetics, 3 ode model (Model 1) (2) Michalis Menten kinetics, 3 ode model (Model 2) [4] and (3) first order kinetics, 4 ode model (Model 3). We have used genetic algorithm to estimate the parameters for all these models and selected the model having minimum error using the estimated parameter.

3.4 Parameter estimation using hybrid algorithm

The flowsheet of Genetic Algorithm used for parameter estimation is shown in Figure 3.5. We have used the uniform crossover and the mutation operators with probabilities 0.8 and 0.01, respectively. As a rule of thumb, the crossover probability is generally greater than 0.75 so as to encourage better exploration of the search space. The top 5% of the solutions in every generation are preserved in the next generation. This makes sure that the best solutions are passed on even if the GA does not find a better solution during the search. A higher value of this elitism operator typically leads to premature convergence and a lower

value might result in the loss of good solutions identified by the GA. The number of generations are given as 100 and the population size is given as 100.

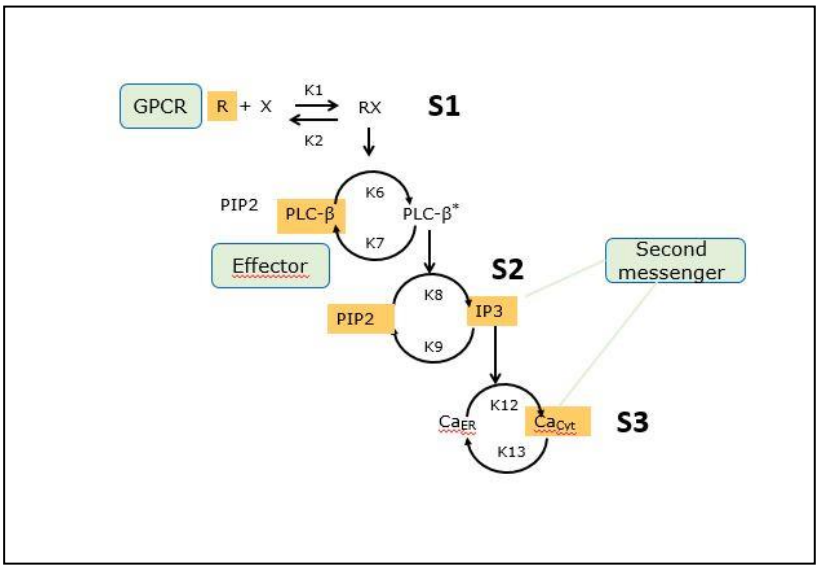


Fig 3.3: Biomolecules present in the signaling pathway

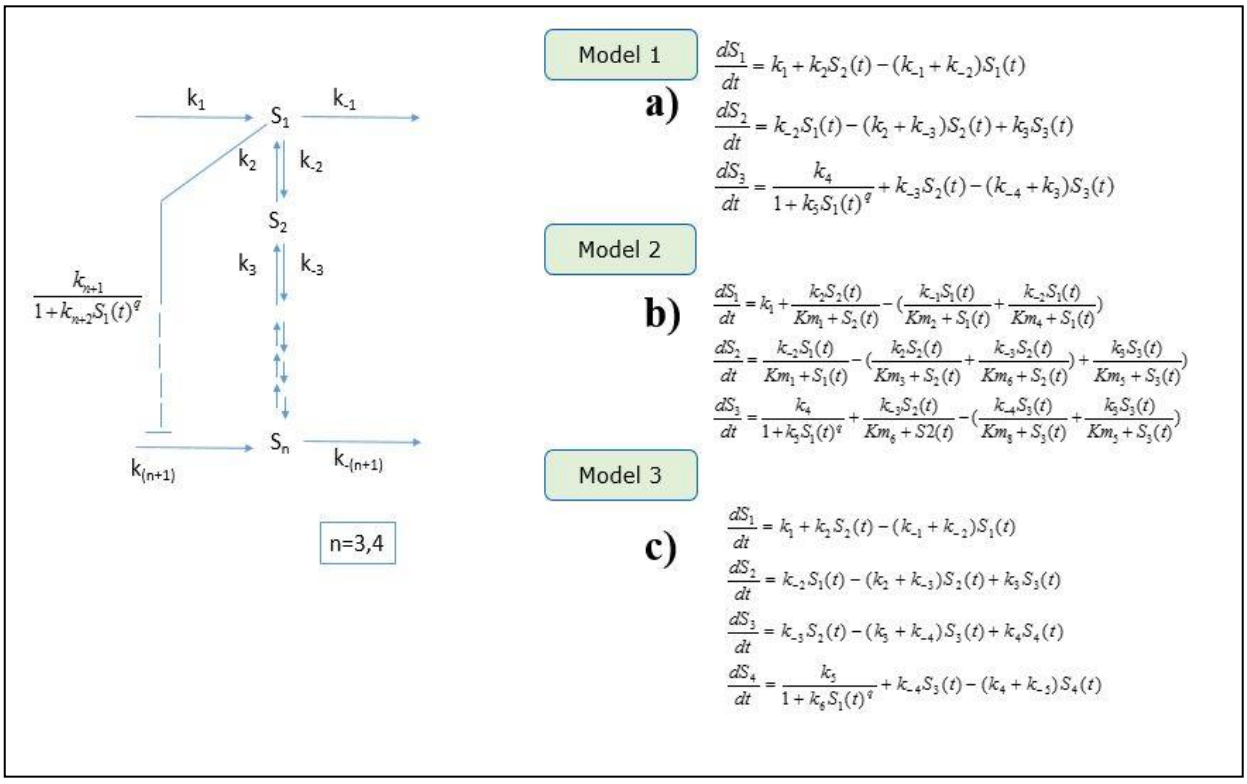


Fig 3.4: Reaction mechanism and the corresponding model equations, (a) 1st order kinetics b) Michalis Menten model c) Four equations 1st order ODE model

3.5 Objective function-Problem formulation :

Model 1 and Model 2 are having 10 parameters while model 3 is having 12 parameters as shown in Table 3.1 The description of the parameters is also given in Table 3.1. The parameters (rate constants) are generated by GA and are given as input to an ODE solver (ode45) for solving coupled differential equations using Runge-Kutta-Method (ode23 in MATLAB).

The objective function is presented as the difference (error to be minimized) measured between the experimental calcium response and the simulated values of intracellular calcium responses from the ode solver.

$$RMSE = \left[\sqrt{\frac{\sum (\exp - sim)^2}{n}} \right]^2$$

This error function is given as input function to the Genetic Algorithm. The GA evaluates the estimated parameter after our criteria meets i.e. either error is < 0.01 or the number of generations exceeds 100. After estimation of parameters from GA, these parameters are given as input to the gradient based method (fmincon in MATLAB).

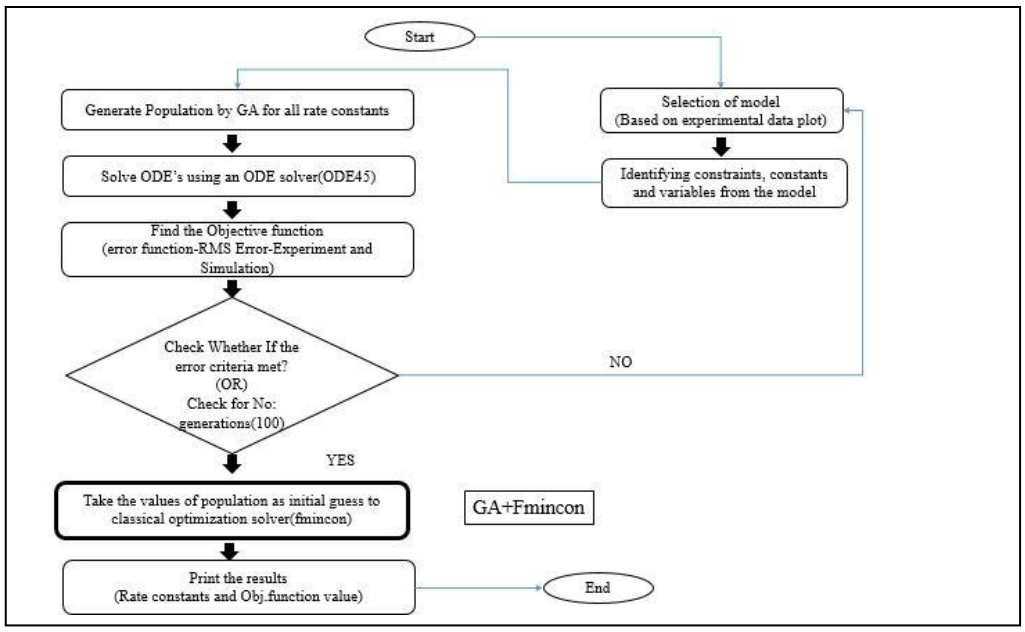


Fig 3.5: Flowsheet for model selection and parameter estimation

Table 4.1: Description of kinetic parameters and their ranges for model 1, 2&3.

Parameter	Description	Range	Model 1	Model 2	Model 3
k_1	rate constant for formation of S1	[0-30]	√	√	√
k_{-1}	rate constant for degradation of S1	[0-30]	√	√	√
k_2	rate constant for formation of S1 and degradation of S2	[0-30]	√	√	√
k_{-2}	rate constant for formation of S2 and degradation of S1	[0-30]	√	√	√
k_3	rate constant for formation of S2 and degradation of S3	[0-30]	√	√	√
k_{-3}	rate constant for formation of S3 and degradation of S2	[0-30]	√	√	√
k_4	rate constant for formation of S3 ,degradation of S4	[0-30]	√	√	√
k_{-4}	rate constant for degradation of S3, formation of S4	[0-30]	√	√	√
k_5	rate constant for degradation of S3 , formation of S4	[0-30]	√	√	√
k_{-5}	rate constant for degradation of S4	[0-30]			√
k_6	rate constant for degradation of S4	[0-30]			√
q	exponential for degradation of S3,S4	[0-10]	√	√	√

Chapter 4

Mathematical modeling of calcium response

In this study, the intracellular calcium concentration of 28 cells were measured through time lapse imaging (119 intervals of time, 28x119 matrix). Using principal component analysis we have reduced the data to 28x2 matrix (taking two principal components into consideration) as shown in Figure 4.1a.

Since the main challenge is to identify the model that explain the whole dataset, we divided these 28 cells into 3 clusters using K-means clustering. From Figure 4.1b we can see that the data is divided into three clusters. Each cluster is responding differently, green cluster indicates fast response, red cluster indicates delayed response and blue cluster indicates flat response. The cells in each cluster is responding in a similar manner with respect to other members in the same cluster. Hence we plan to identify three models for each of these three clusters. Figure 4.2 gives the experimental cell data plot after clustering the cell data.

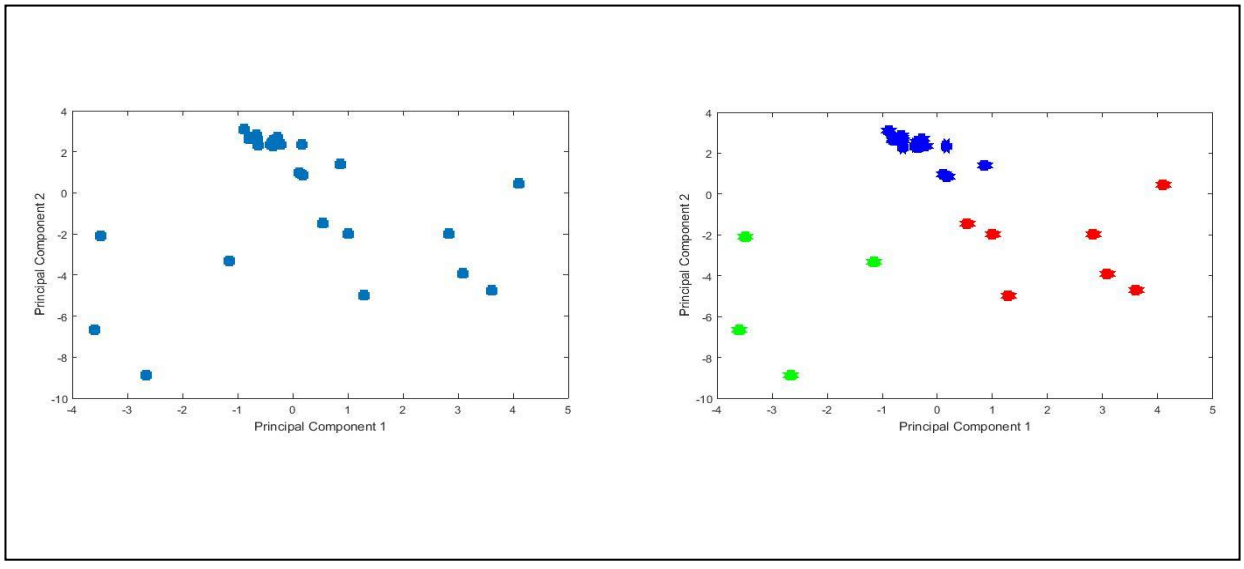


Fig 4.1: The scatter plot of two principal components for **400ng drug dose data a)** Before Clustering **b)** After Clustering using K-Means (K=3)

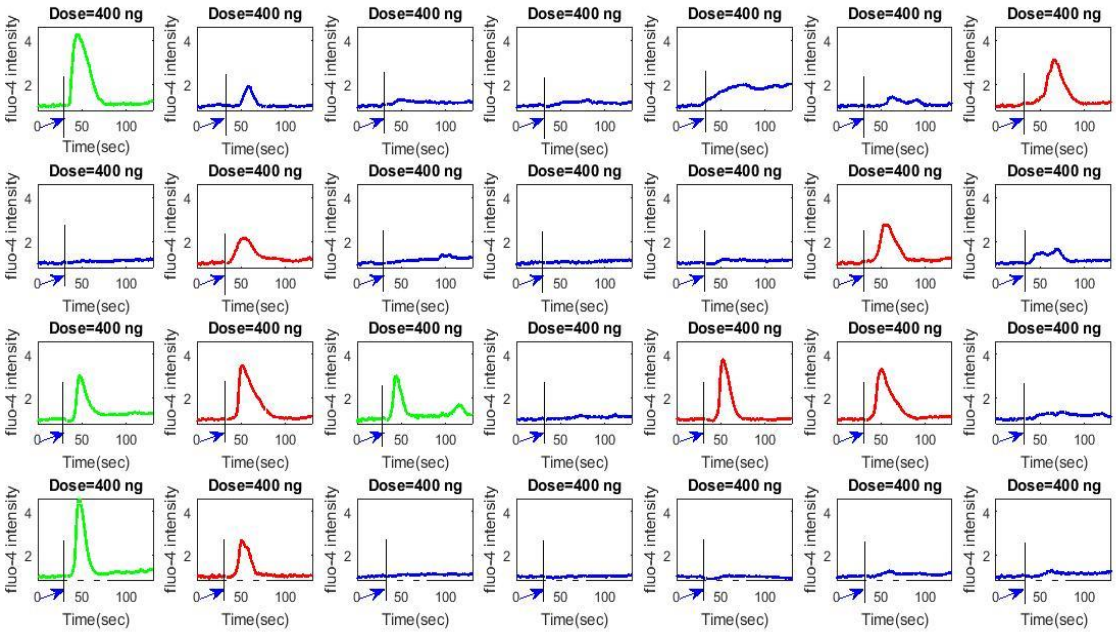


Fig 4.2: Three types of calcium response obtained through K-means clustering of calcium response in a cell population, Green-Immediate response, Red-Delayed response, Blue-Flat response (Low amplitude)

Cluster 1:

We took time series data from one cell (cluster 1) for model fitting and tried to fit with three models, model 1, model 2 and model 3. Figure 4.3, shows the plot of simulated calcium response (blue) vs experimental calcium response (pink) along with the error with each time point. The next panel shows the scatter plot of simulated response vs experimental response for various models. Table 4.1 shows the values of estimated parameters and objective function value/error by Genetic Algorithm. The results clearly shows that Model 2 having the Michaelis Menten model is yields comparatively less root mean square error (RMSE = 0.1129). Hence our framework can be used for selection of the best fitting model for the dynamic data.

Cluster 2:

Similarly we took another representative data (calcium response in a cell) from cluster 2 for model fitting and we investigated three models. The kinetic parameters are estimated by Genetic Algorithm. Figure 4.4 shows the plot of simulated calcium response (blue) vs experimental calcium response (pink) along with the error at each time point. The other panel shows the scatter plot for the simulated vs experimental response for various models. Table 4.2 shows the values of estimated parameters and objective function value by genetic Algorithm. We observed that all models are giving RMSE of approximately ~ 0.047 with the minimum error for model 3 (Simple 1^{st} order ODE with 4 equations).

Cluster 3:

Similarly we took another representative data (calcium response in a cell) from cluster 3 for model fitting and we investigated three models. The kinetic parameters are estimated by Genetic Algorithm. The Figure 4.5, shows the plot of simulated response (blue) vs experimental response (pink) along with the error. The other panel shows the simulated vs experimental response for various models. Table 4.3 shows the values of parameters and objective function value by genetic Algorithm. We observed that Model 1 (1^{st} order model) is yields comparatively less error (RMSE = 0.2422).

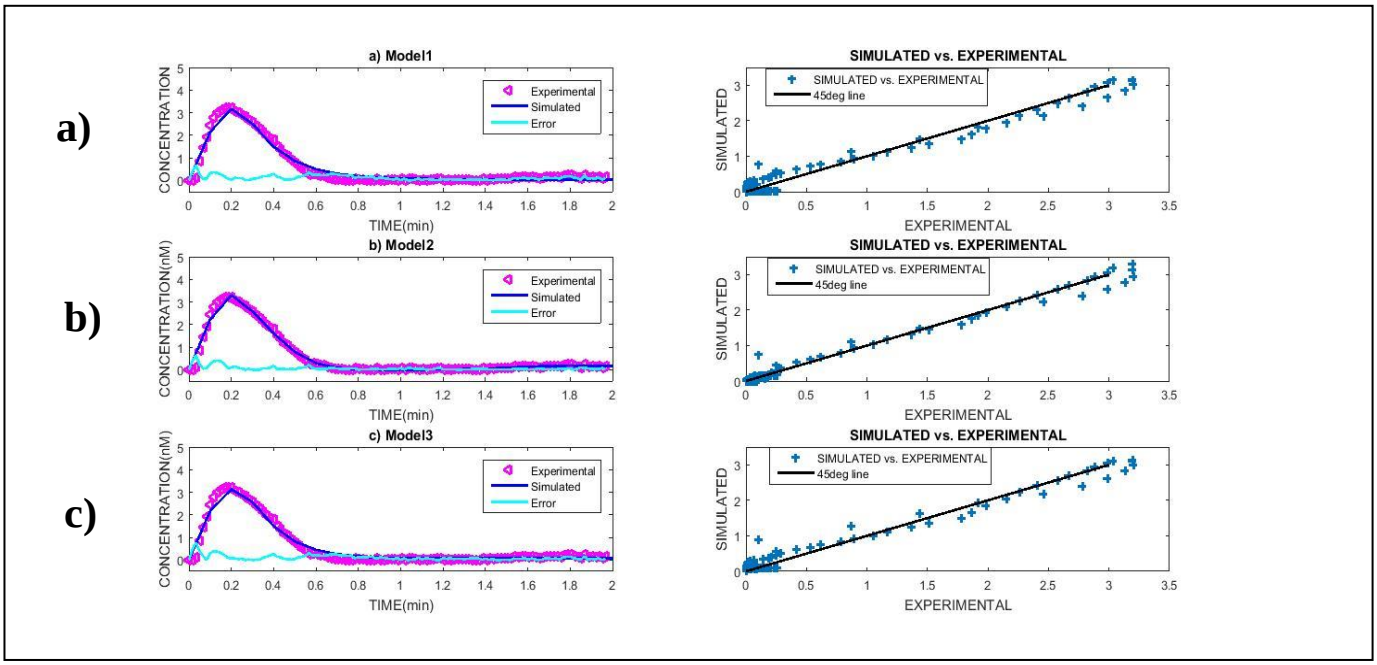


Fig 4.3: Panel 1: Time course of calcium response from experiments (cluster 1) and simulation, Panel 2: Scatter plot of simulation vs experimental data for cluster 1, a**)** Model 1 **b)** Model 2 **c)** Model 3

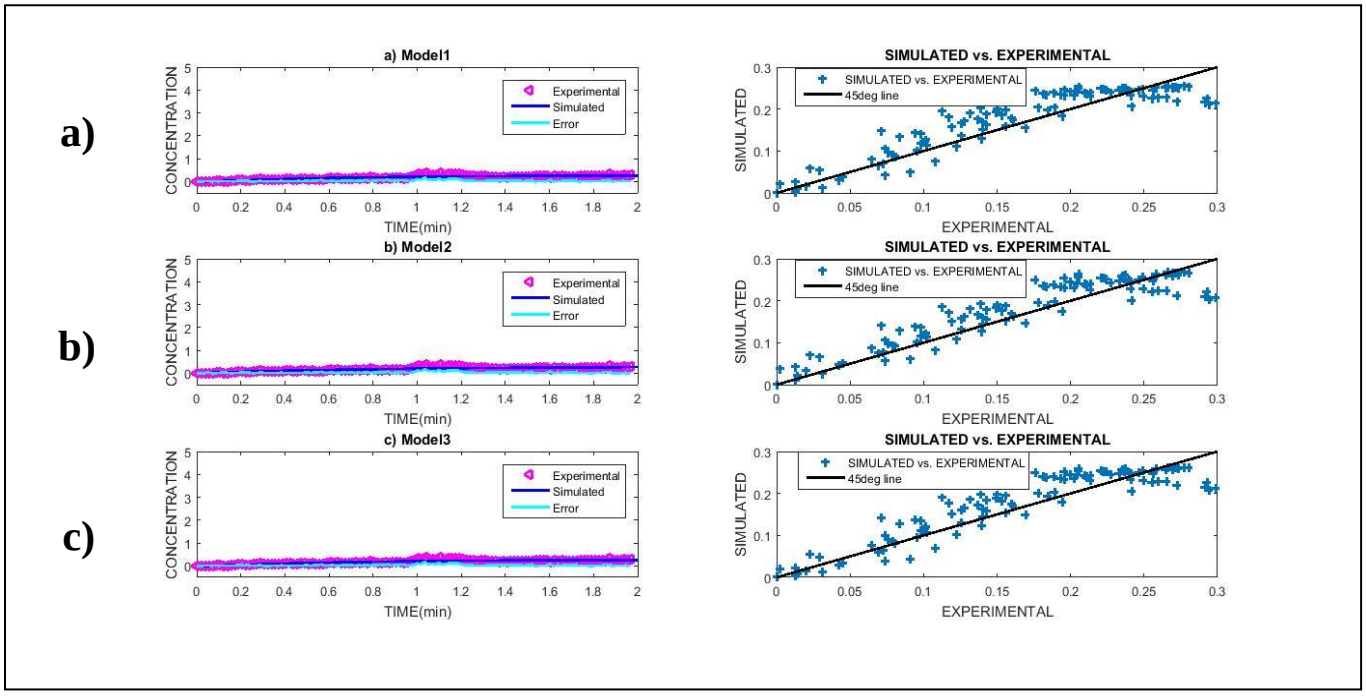


Fig 4.4: Panel 1: Time course of calcium response from experiments (cluster 2) and simulation, Panel 2: Scatter plot of simulation vs experimental data for cluster 2, **a)** Model 1 **b)** Model 2 **c)** Model 3

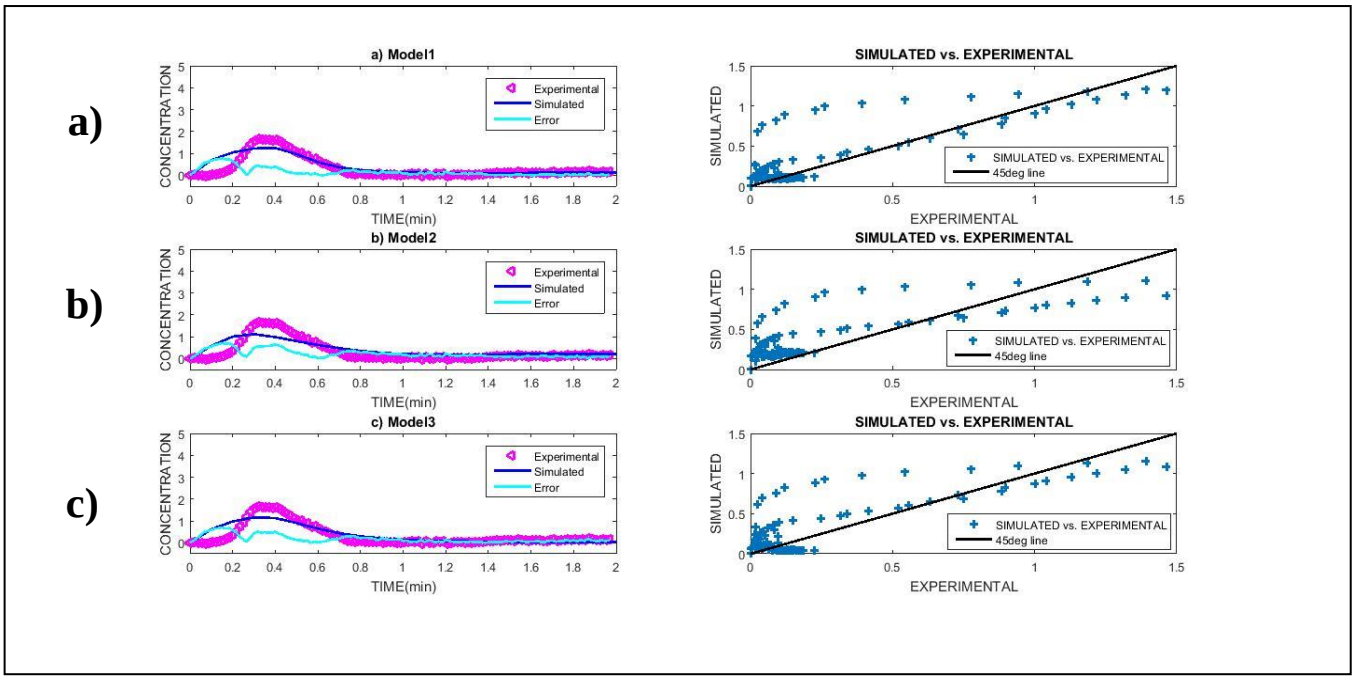


Fig 4.6: Panel 1: Time course of calcium response from experiments (cluster 3) and simulation, Panel 2: Scatter plot of simulation vs experimental data for cluster 3, a**)** Model 1 **b)** Model 2 **c)** Model 3

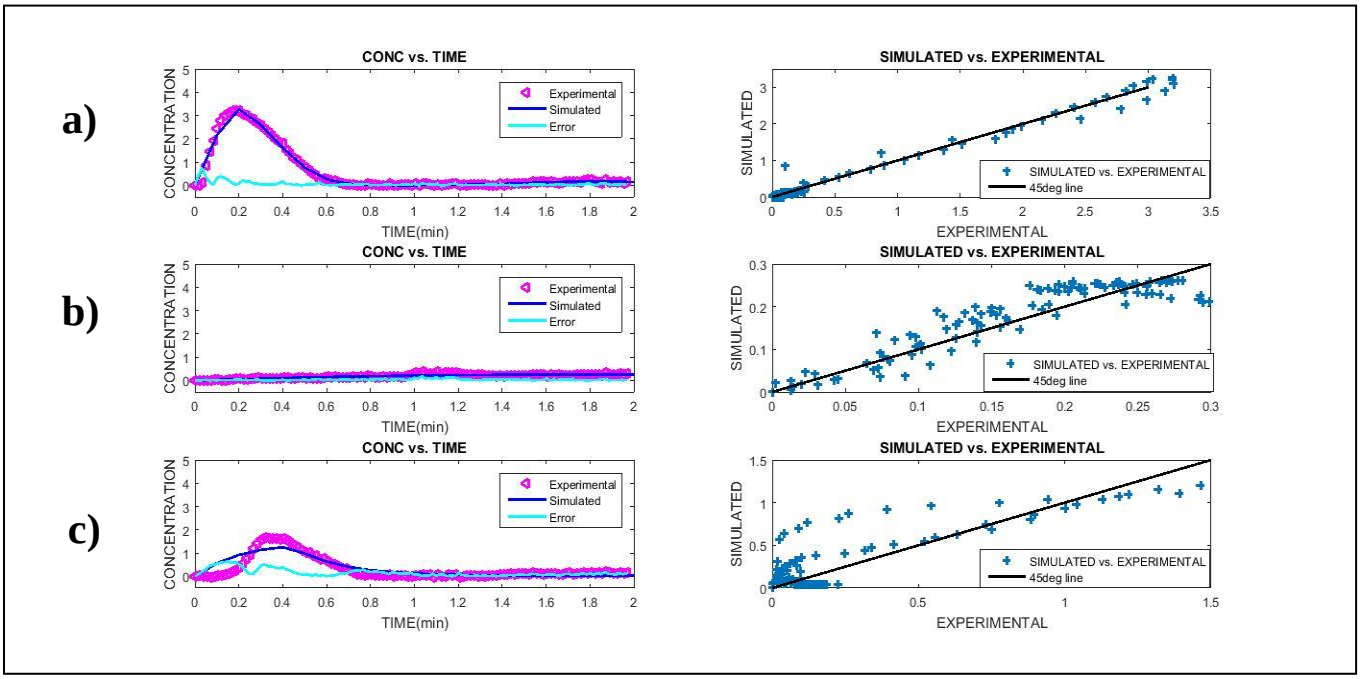


Fig 4.5: Panel 1: Time course of calcium response from experiments and simulation, Panel 2: Scatter plot of simulation vs experimental data, a**)** Cluster 1 **b)** Cluster 2 **c)** Cluster 3

Hybrid Method (GA+Fmincon) Results:

Estimated parameters from Genetic Algorithm corresponding to best fit models for each cluster has been given as the input for the gradient search. Figure 4.6, shows the fitting results from the hybrid method (GA+Fmincon method). Table 4.4 shows the values of parameters and objective function values estimated by hybrid method and Table 4.5 shows the performance of several methods.

Table 4.1: Estimated kinetic parameters by GA and corresponding error for data in cluster 1 (for various models: Model **1:10 parameters, Model 2, 10 parameters, Model 3: 12 parameters)**

	k1	k-1	k2	k-2	k3	k-3	k4	k-4	k5	k-5	k6	q	Error
Model 1	4.5173	3.0493	7.6822	0.0376	2.4952	29.9899	29.9853	5.921	2.2221			9.993	0.1592
Model 2	8.2734	18.5615	8.2389	1.7819	11.7065	0.022	29.9976	3.8641	7.0948			9.9786	0.1124
Model 3	11.7155	11.1817	13.1449	29.7656	14.3503	0.0625	1.6875	0.0404	29.9845	5.1562	24.7968	9.9938	0.1548

Table 4.2: Estimated kinetic parameters by GA and corresponding error for data in cluster 2 (for various models: **Model 1:10 parameters, Model 2, 10 parameters, Model 3: 12 parameters)**

	k1	k-1	k2	k-2	k3	k-3	k4	k-4	k5	k-5	k6	q	Error
Model 1	26.283	1.7135	15.1203	4.6709	12.2392	2.5191	1.4451	19.992	14.7232			2.4424	0.0471
Model 2	4.15	2.5924	14.7742	13.5444	26.7765	13.4107	0.0164	6.4107	29.8615			0.0299	0.0481
Model 3	18.8928	19.595	0.43	8.0887	28.0067	7.4106	16.5513	14.0516	1.3928	16.6344	29.6957	1.7919	0.0469

Table 4.3: Estimated kinetic parameters by GA and the corresponding error for cluster 3 for various models **(Model 1:10 parameters, Model 2, 10 parameters, Model 3: 12 parameters)**

	k1	k-1	k2	k-2	k3	k-3	k4	k-4	k5	k-5	k6	q		Error
Model 1	3.5465	0.2283	5.5191	20.812	4.0636	0.1333	9.4745	2.7514	22.1572			9.9742		0.2422
Model 2	12.8555	22.4235	22.7948	0.0129	1.4145	26.4374	8.8997	7.1491	0.433			9.9648		0.2824
Model 3	8.5719	6.6864	15.2718	29.9246	29.545	17.5238	1.076	0.1217	7.9826	4.2966	25.4907	9.9885		0.2507

Table 4.4: Estimated kinetic parameters by hybrid method (GA+Fmincon) and the corresponding error for the data for three clusters (for the selected models)

			k1	k-1	k2	k-2	k3	k-3	k4	k-4	k5	k-5	k6	q		Error
Cluster 1	GA		8.2734	18.5615	8.2389	1.7819	11.7065	0.022	29.9976	3.8641	7.0948			9.9786		0.1124
	GA+Fmincon		9.2012	19.8172	7.4072	2.1193	8.5806	0.0107	29.6265	7.7122	8.3452			9.9779		0.1172
Cluster 2	GA		26.283	1.17135	15.1203	4.6709	12.2392	2.5191	1.4451	19.992	14.7232			2.4424		0.0471
	GA+Fmincon		26.4606	1.1078	13.5055	3.6502	14.2483	2.1777	0.4879	22.083	14.8104			2.0166		0.0478
Cluster 3	GA		3.5465	0.2283	5.5191	20.812	4.0636	0.1333	9.4745	2.7514	22.1572			9.9742		0.2422
	GA+Fmincon		0.0013	0.0047	7.2185	5.8105	4.7656	0.1209	7.1163	0.0352	21.9957			9.9909		0.2223

Table 4.5: Performance of the method: Estimated time of computation for Genetic Algorithm and Gradient based search

	Genetic Algorithm	Fmincon
Model 1	829 sec	70 sec
Model 2	623 sec	46sec
Model 3	934 sec	97sec

Chapter 5

Sensitivity Analysis

One of the major difficulties in constructing mathematical models of biological system is the lack of precise parameter values which are often associated with a high degree of uncertainty. This uncertainty in parameter values can be incorporated into the modelling process using sensitivity analysis, the systematic investigation of the relationship between uncertain model inputs and the resulting variation in the model outputs.

Here we performed sensitivity analysis to identify the parameters that has significant effect on the model output. We performed sensitivity analysis through change in one of the kinetic parameters by $\pm$50% of the estimated value keeping other parameters constant. Figure 5.1-5.6 shows the sensitivity analysis with respect to various kinetic parameters (Here we show the results for 6 parameters k_1, k_{-1}, k_2, k_{-2}, k3, and q respectively, for other parameters, the analysis was performed but the result is not shown) for Model 1 fitted to cluster 1.

Here we show that the cell response is sensitive to most of the kinetic parameters. Similar analysis was performed for cluster 1 by model 2 and model 3.The results are tabulated in Table 5.1 .The result shows that for Model 2, the simulated calcium response is not sensitive to k_{-2} and k_{-3} and for Model 3 the simulated calcium is not sensitive with respect to k_3, k_{-3}, k_{-4}, and k_{-6}.

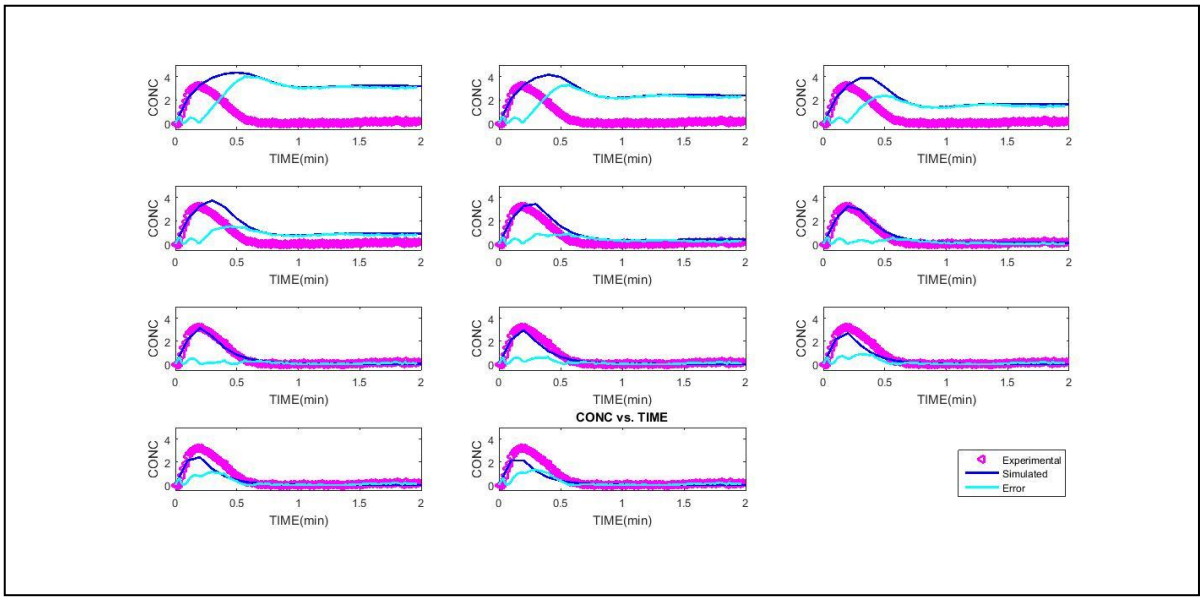


Fig 5.1: Sensitivity analysis for k_1: Plot of Simulated results, Experimental results, and Error with time (for cluster 1, model 1 when parameter 'k_1' is varied from [1-7]).

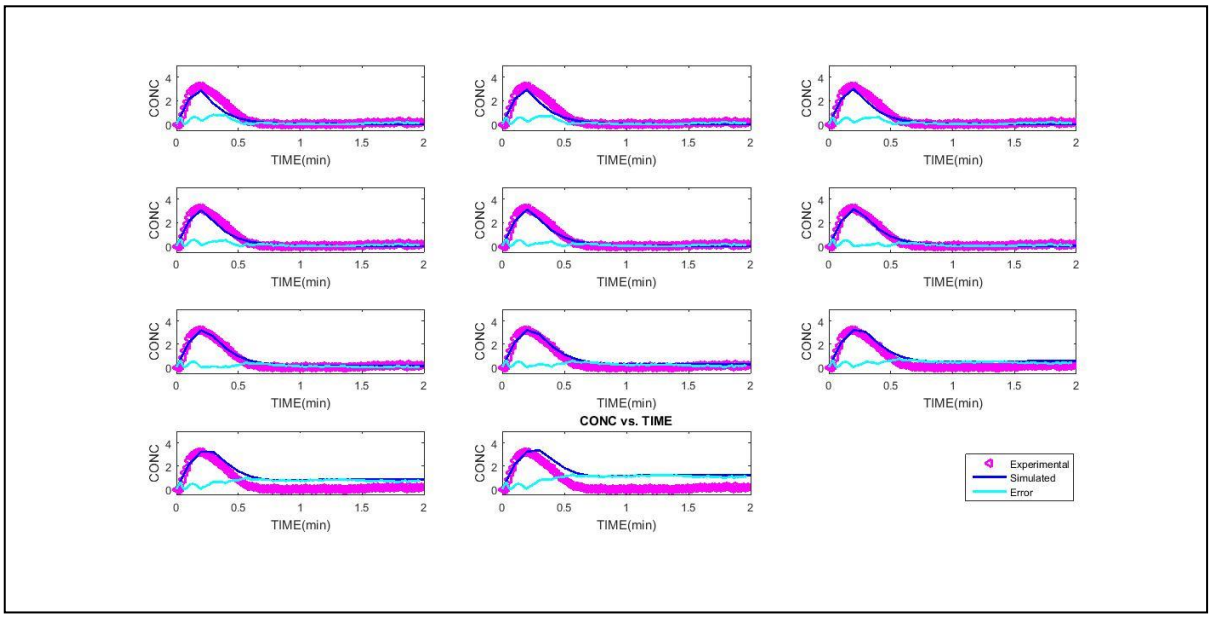


Fig 5.2: Sensitivity analysis for k_{-1}: Plot of Simulated results, Experimental results, and Error with time (for cluster 1, model 1 when parameter 'k_{-1}' is varied from [1-5])

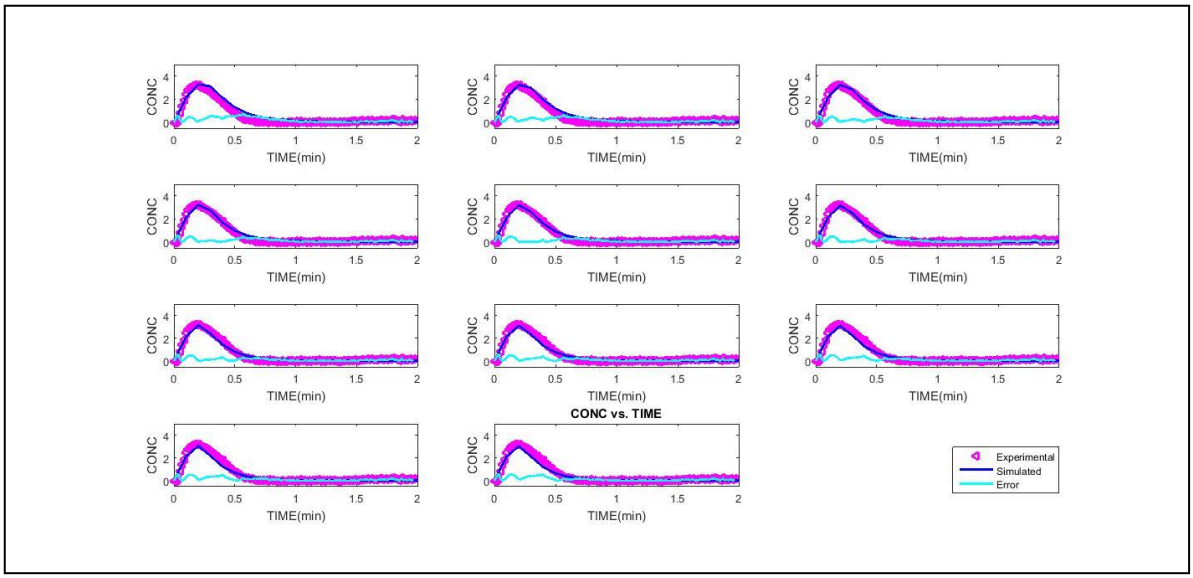


Fig 5.3: Sensitivity analysis for k_2: Plot of Simulated results, Experimental results, and Error with time (for cluster 1, model 1 when parameter 'k_2' is varied from [3-12]).

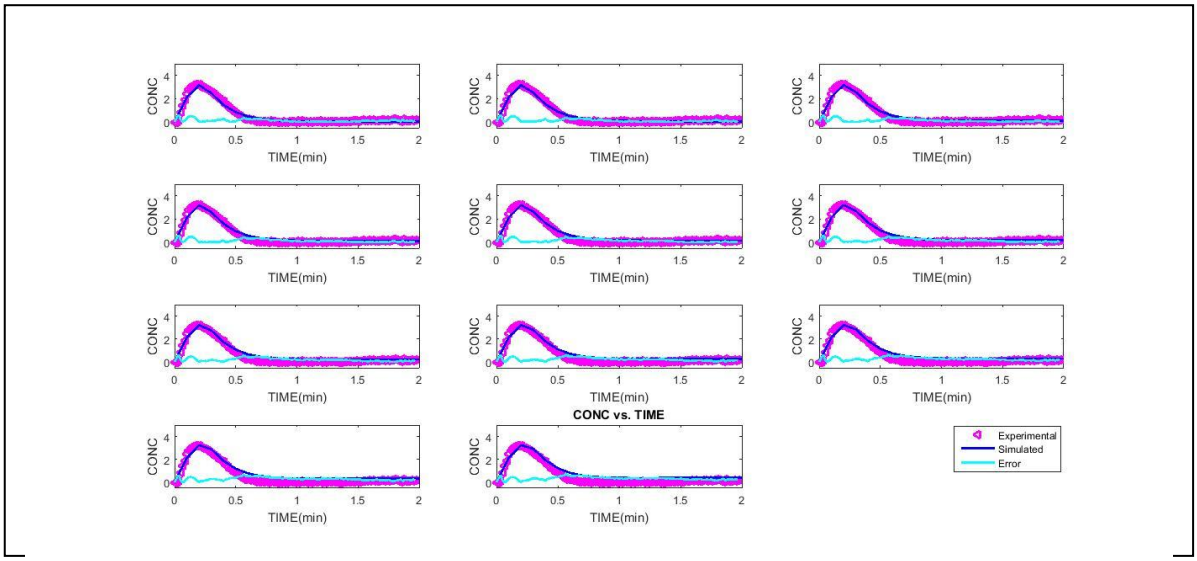


Fig 5.4: Sensitivity analysis for k_{-2}: Plot of Simulated results, Experimental results, and Error with time (for cluster 1, model 1 when parameter 'k_{-2}' is varied from [0-1]).

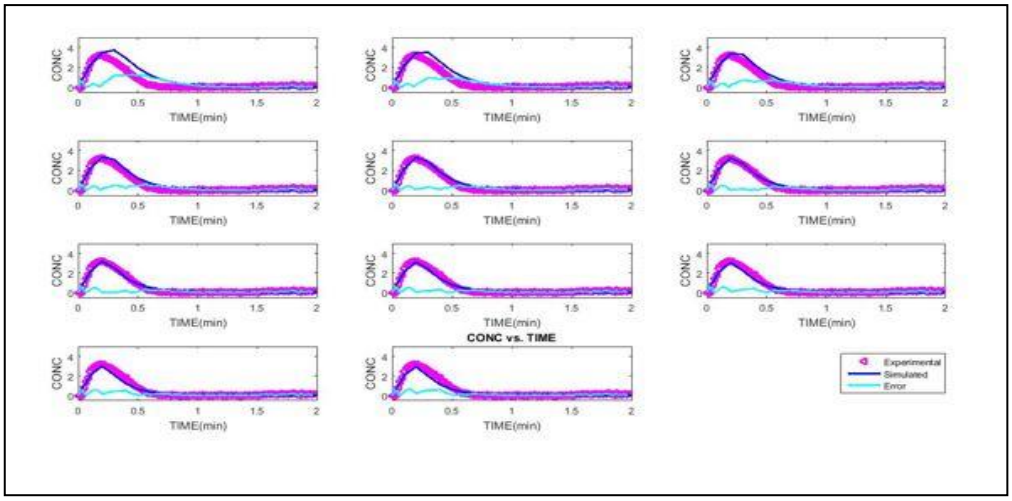


Fig 5.5: Sensitivity analysis for k_3: Plot of simulated results, experimental results, and error with time (for cluster 1, model 1 when parameter 'k_3' is varied from [0-4]).

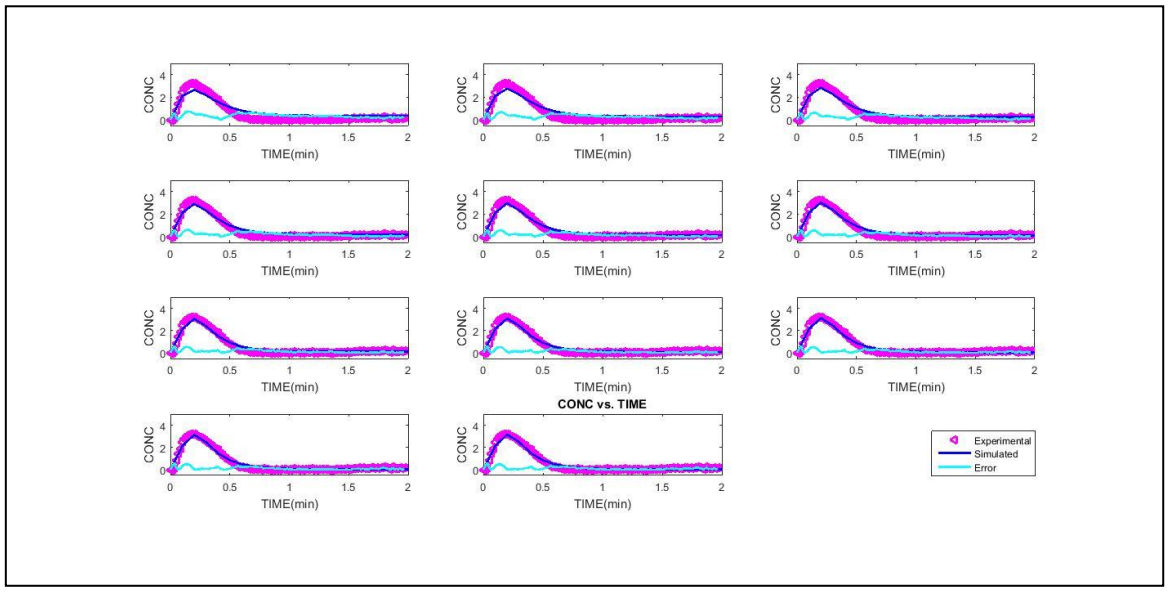


Fig 5.6: Sensitivity analysis for q: Plot of Simulated results, experimental results, and error with time (for cluster 1, model 1 when parameter 'q' is varied from [4-10]).

Table 5.1: Summary of sensitivity analysis results for a) model 1 b) model 2 c) model 3 for cluster 1

Model 1	Range	
k1	[1-7]	√
k_1	[1-5]	√
k2	[3-12]	√
k_2	[0-1]	√
k3	[0-4]	√
k_3	[14-30]	√
k4	[14-30]	√
k_4	[2-9]	√
k5	[0-4]	√
q	[4-10]	√

Model 2	Range	
k1	[3-13]	√
k_1	[8-28]	√
k2	[3-13]	√
k_2	[0-3]	X
k3	[5-18]	√
k_3	[0-1]	X
k4	[14-30]	√
k_4	[1-6]	√
k5	[3-11]	√
q	[4-10]	√

Model 3	Range	
k1	[5-18]	√
k_1	[5-17]	√
k2	[6-20]	√
k_2	[14-30]	√
k3	[6-22]	X
k_3	[0-1]	X
k4	[0-3]	√
k_4	[0-1]	X
k5	[14-30]	√
k_5	[2-8]	√
k6	[11-30]	X
q	[4-10]	√

Chapter 6

Analysis of delayed calcium response

Three models chosen based on various kinetic mechanisms were not able capture the trend in calcium responses under cluster 3 (delayed response). In order to explain the delay in the response we formulated a series of models corresponding to other reaction mechanism as shown in Figure 6.1.We proposed three different network structures/motifs based on various topologies. The first structure contains only substrate inhibition (Figure 6.1a), the second structure contains only product inhibition (Figure 6.1b), and the third one contains a combination of substrate and product inhibition (Figure 6.1c). For each structure we proposed five different types of kinetic mechanisms for substrate and product inhibition, exponential terms in transportation of calcium as shown in Figure 6.2. a) Product inhibition of type 1 ($\frac{k_1}{k_2 + k_3 S(t)^q}$) b) Product inhibition of type 2 ($\frac{k_1}{k_2 + k_3 S(t) + k_4 S(t)^2}$) [1010] c) Gaussian substrate inhibition of type 3 d) One exponential term for transportation of calcium concentration (S3) e) Two exponential terms for transportation of calcium concentration S3. Using various combinations of these models, we constructed 23 possible models.

The parameters were estimated by using Genetic Algorithm and the model with minimum error was chosen as the best possible model. For optimization, we used number of generations as 100, population size as 100 and the probability of Genetic operators like cross over, mutation and elitism probabilities as 0.8, 0.01 ,0.05 respectively. The models were solved by using an ODE solver (Ode 45 in MATLAB).

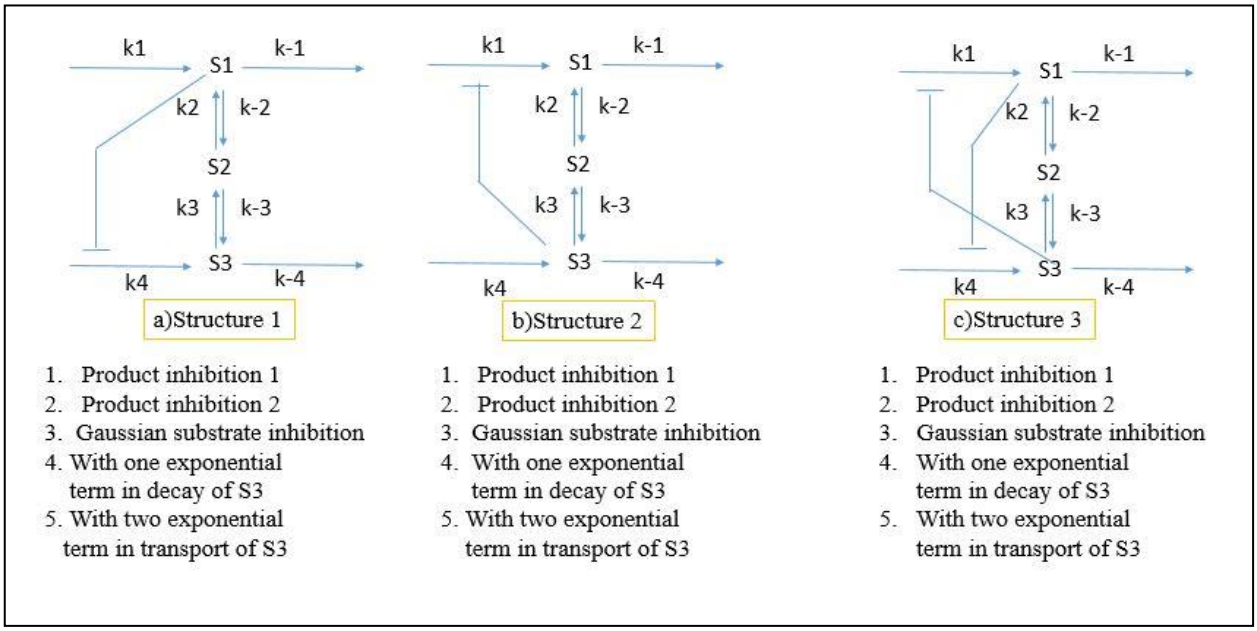


Fig 6.1: Three possible network structures/motifs and corresponding mechanisms for cluster 3: **a)** with substrate inhibition **b)** product inhibition **c)** both substrate and product inhibition

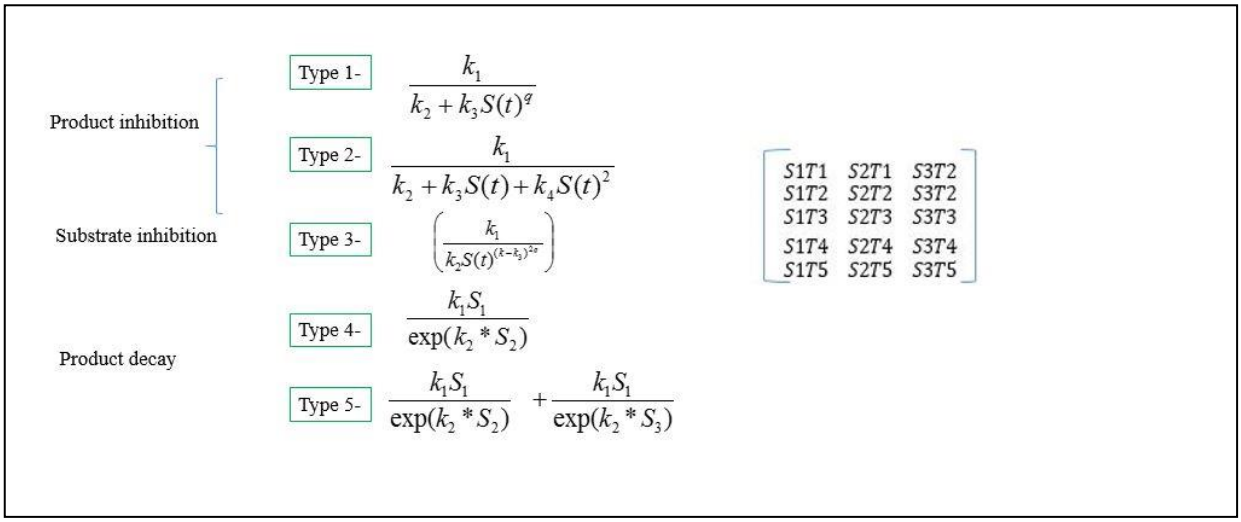


Fig 6.2: Possible kinetic mechanisms for substrate and product inhibition and calcium transportation

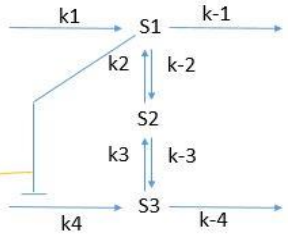
Model :S1T3

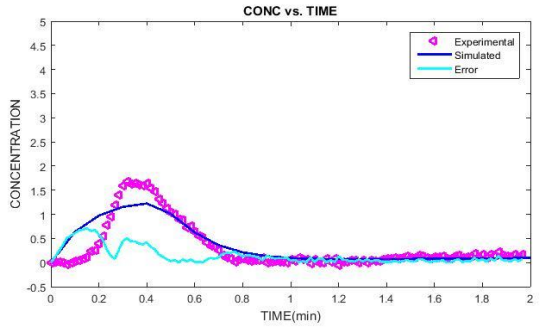
$$\frac{dS_1}{dt} = k_1 + \frac{k_2 S_2(t)}{Km_1 + S_2(t)} - \left(\frac{k_{-1} S_1(t)}{Km_2 + S_1(t)} + \frac{k_{-2} S_1(t)}{Km_4 + S_1(t)}\right)$$

$$\frac{dS_2}{dt} = \frac{k_{-2} S_1(t)}{Km_1 + S_1(t)} - \left(\frac{k_2 S_2(t)}{Km_3 + S_2(t)} + \frac{k_{-3} S_2(t)}{Km_6 + S_2(t)}\right) + \frac{k_3 S_3(t)}{Km_5 + S_3(t)}$$

$$\frac{dS_3}{dt} = \frac{k_4}{1 + k_5 S_1(t)^{(0.4 - k_4)^{2q}}} + \frac{k_{-3} S_2(t)}{Km_6 + S_2(t)} - \left(\frac{k_{-4} S_3(t)}{Km_8 + S_3(t)} + \frac{k_3 S_3(t)}{Km_5 + S_3(t)}\right)$$

S1 inhibits the formation of S3





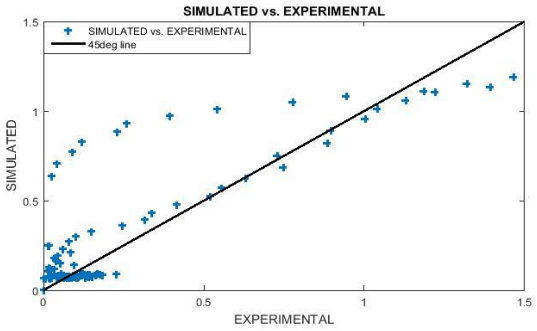
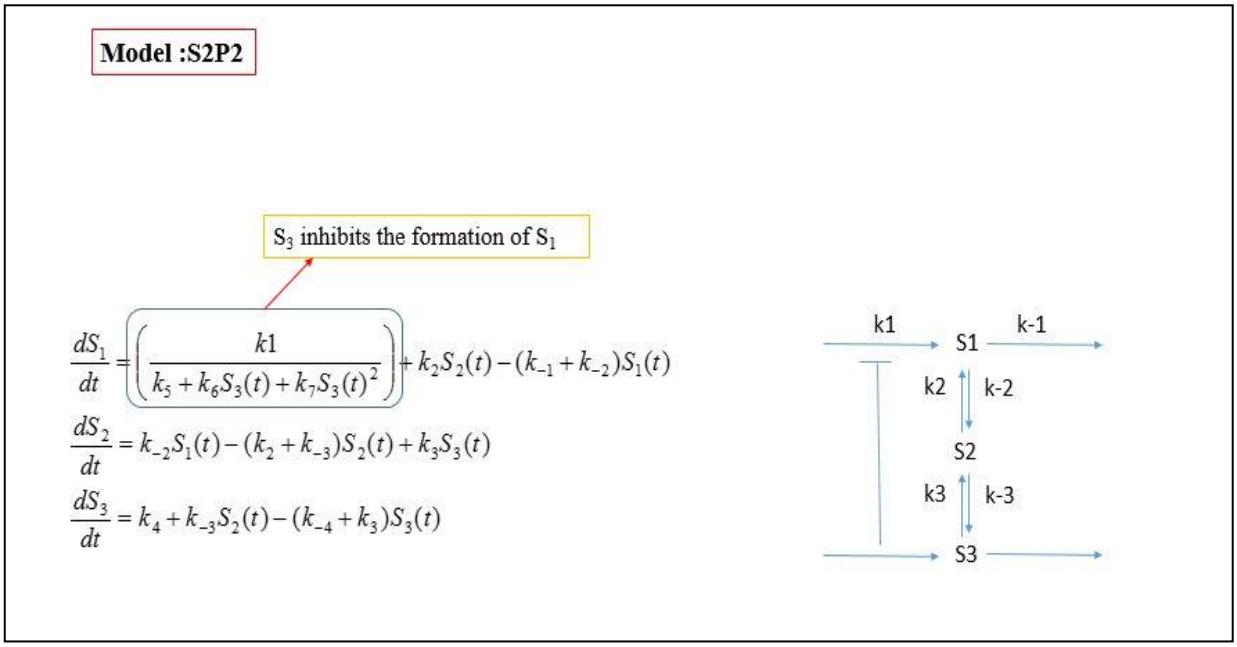


Fig6.3: **a)** Reaction mechanism and kinetic model with inhibition of type 3:**S1T3 b)** Panel 1: Time course of calcium response from experiments (cluster 3) and simulation, Panel 2: Scatter plot of simulation vs experimental data for cluster 3, Model with inhibition of type 3:**S1T3**



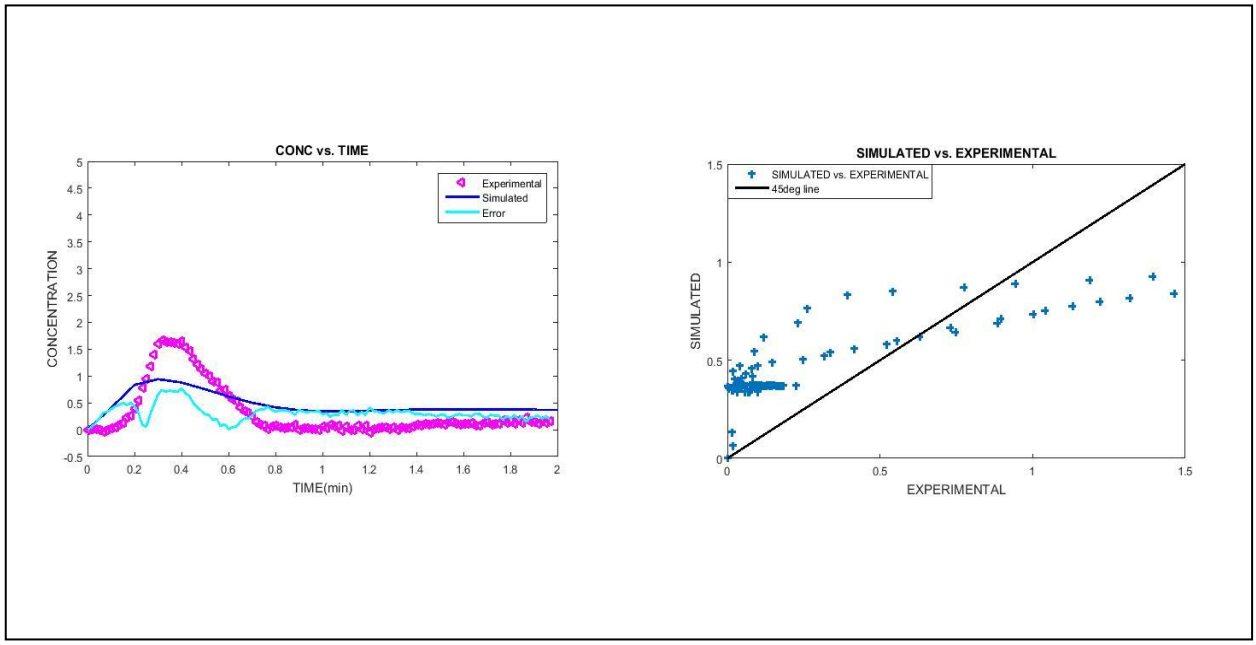
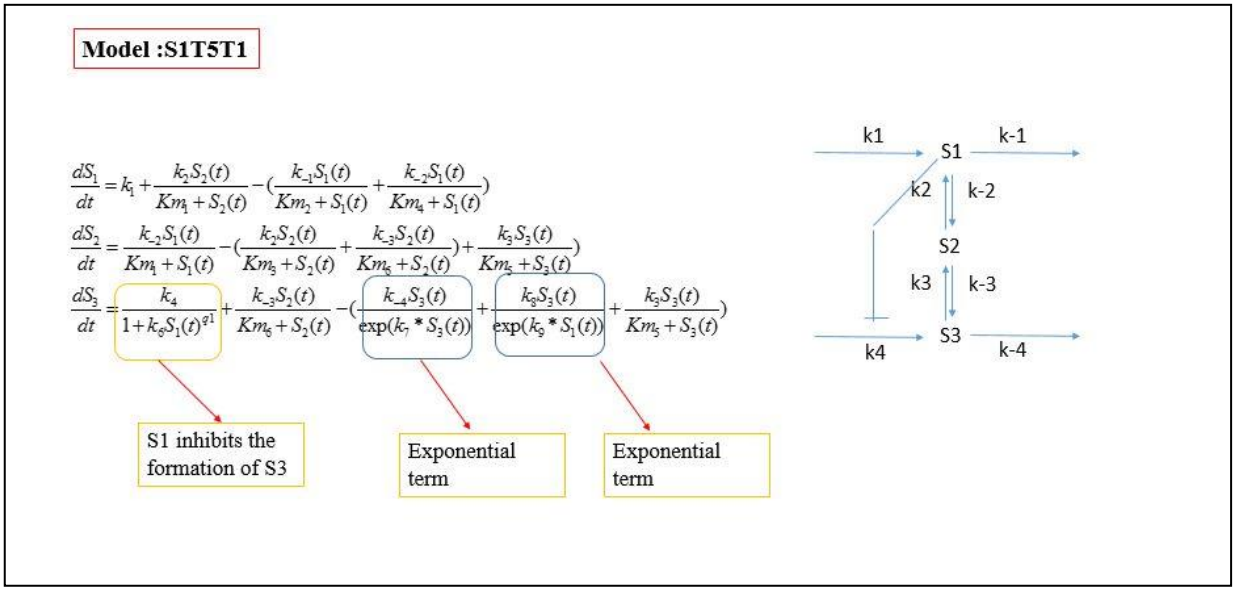


Fig 6.4: **a)** Reaction mechanism and kinetic Model with inhibition of type 2 **S2P2** **b)** Panel 1: Time course of calcium response from experiments (cluster 3) and simulation, Panel 2: Scatter plot of simulation vs experimental data for cluster 3, Model with inhibition of type 2:**S2P2**



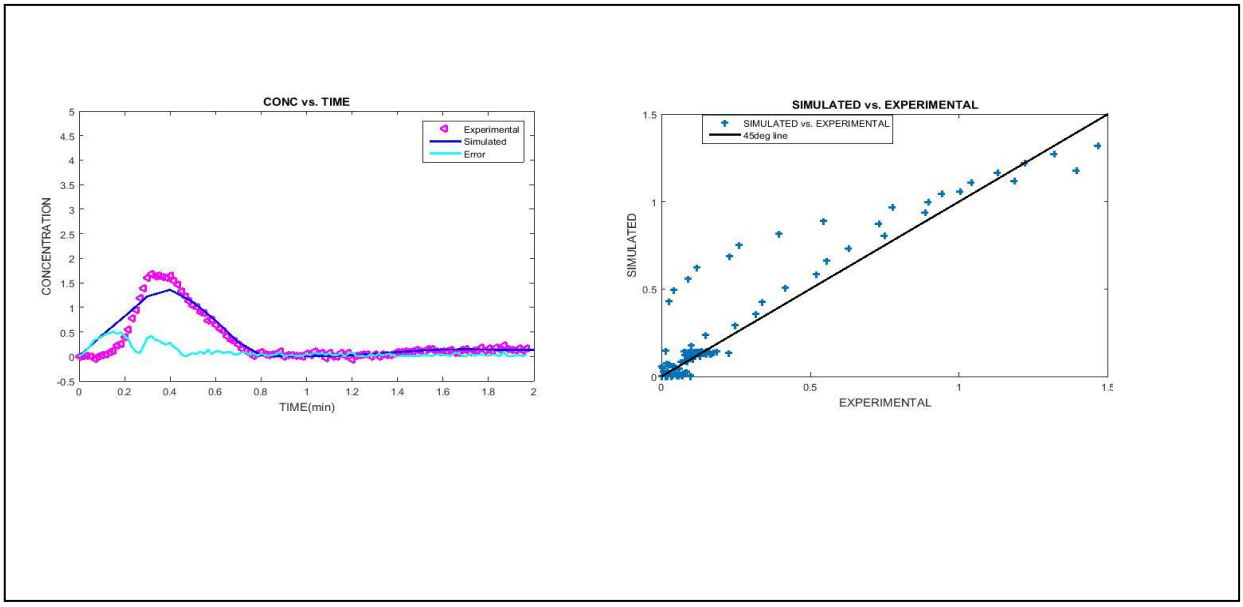


Fig 6.5: a) Reaction mechanism and kinetic Model with inhibition of type 1 and type 5**(case 2) S1T5T1**: **b)** Panel 1: Time course of calcium response from experiments (cluster 3) and simulation, Panel 2: Scatter plot of simulation vs experimental data for cluster 3, Model with inhibition of type 1&5 **(case 2): S1T5T1**

Table 6.1: Error analysis for three types of network structure/reaction mechanisms proposed for cluster 3, Model with exponential transportation shows minimum error.

	Error(Rmse)
Model S1T1	------
Model S1T2	------
Model S1T3	0.2739
Model S1T4	0.2819
Model S1T5	0.3917
Model S1T5T1	0.1596

Structure 1

	Error(Rmse)
Model S2T1	0.4759
Model S2T2	0.4757
Model S2T3	-------
Model S2T4	0.4756
Model S2T5	0.4762

Structure 2

	Error(Rmse)
Model S3T1	0.4764
Model S3T2	0.4716
Model S3T3	0.4798
Model S3T4	0.4762
Model S3T5	0.4757

Structure 3

We investigated various combinations of inhibition terms in the models and found that on including inhibition terms of type 1($\frac{k_1}{k_2 + k_3 S(t)^q}$) and type 4($\frac{k_1 S_1}{\exp(k_2 * S_2)}$) yields comparatively less RMSE (0.1596) [See Figure 6.5a and 6.5b]. From the error analysis as shown in Table 6.1, it can be concluded that the minimum error is 0.2739 and the most suitable model contains the exponential terms for calcium transportation. Here we show the results on investigations for various reaction mechanism. Figure 6.3 to 6.5 shows the simulated calcium response vs experimental calcium response along with the error. The second panel of these figures show the scatter plot of simulated vs experimental responses.

Chapter 7
Conclusions & Discussion

Conclusions: The present work provides a framework for finding the most suitable model corresponding the intracellular calcium response from a list of models. From Figure 4.3, and from Table 4.2, we can conclude that the model having Michalis Menten kinetics can capture the immediate calcium response (cluster 1) well compared to other models. From Figure 4.4 and from Table 4.3, we can conclude that a particular parameter set for most of the models can capture the flat responses or low amplitude responses (cluster 2). From Figure 4.5 and from Table 4.4, we can conclude that simple 1^{st} order model can capture the cluster 3 best compared to other possible models.

For all the kinetic mechanisms used, GA may not yield the estimation of parameters as upon reaching a global basin, it takes time to find the optimum there as the nature of search in GA is stochastic in nature. On the other hand, the disadvantage of the classical method is that they progress based on the gradient information at the current point and prone to get structure to a local optima very frequently. However, Global optima can be found very fast by the classical method if the initial guess has been provided in the global basin. Using this basis, GA is used to find the global basin first and then use the classical gradient based search technique to find the local minima. The assumption here is that the GA yields the global basin for the given optimization problem.

Based on the current work and computational framework, further improvements are possible as follows:

- ***Detailed validation of the model***: All the data in one cluster (for various drug doses) can be fitted to the selected model for that cluster. We need to choose multiple videos and cluster the data so that the number of data in one cluster will be twice that of the number of kinetic parameters in the model.
- ***Advanced clustering techniques***: Advanced clustering method such as support vector machine can be used for clustering the cells in contrast to Kmeans clusters (which clusters the data in a linear manner).
- ***Updating model formulation***: We can update our model database through inclusion of more complex mechanisms having positive and negative feedbacks, mutual

inhibition, co-operative binding sites. Additionally we can have a more exhaustive set of models having combinations of substrate inhibition, product inhibition, mutual inhibitions, and exponential terms in reactions and transports.

- ***Optimization techniques***: In future, we may also perform the parameter estimation using techniques such as improved versions of GA such as DNA-GA, RNA-GA, VA-DNA GA, and techniques other than GA such as particle swarm optimization, simulated annealing etc.

References

1. Sumner, T., Sensitivity analysis in systems biology modelling and its application to a multi-scale model of blood glucose homeostasis, 2010, UCL (University College London).
2. Abdullah, A., et al., A new particle swarm evolutionary optimization for parameter estimation of biological models. Int J Comput Inf Syst Ind Manag Appl, 2013. **5**: p. 571-580.
3. Katare, S., et al., A hybrid genetic algorithm for efficient parameter estimation of large kinetic models. Computers & chemical engineering, 2004. **28**(12): p. 2569-2581.
4. Ghovvati, M., et al., Comparison across growth kinetic models of alkaline protease production in batch and fed-batch fermentation using hybrid genetic algorithm and particle swarm optimization. Biotechnology & Biotechnological Equipment, 2015. **29**(6): p. 1216-1225.
5. Nougués, J.M.a., M.D. Grau, and L. Puigjaner, *Parameter estimation with genetic algorithm in control of fed-batch reactors.* Chemical Engineering and Processing: Process Intensification, 2002. **41**(4): p. 303-309.
6. Tao, J. and N. Wang, DNA computing based RNA genetic algorithm with applications in parameter estimation of chemical engineering processes. Computers & Chemical Engineering, 2007. **31**(12): p. 1602-1618.
7. Chen, X. and N. Wang, A DNA based genetic algorithm for parameter estimation in the hydrogenation reaction. Chemical Engineering Journal, 2009. **150**(2): p. 527-535.
8. Ehsani, M.R., H. Bateni, and G.R. Parchikolaei, Modeling the oxidative coupling of methane using artificial neural network and optimizing of its operational conditions using genetic algorithm. Korean Journal of Chemical Engineering, 2012. **29**(7): p. 855-861.
9. Iglesias, P.A. and B.P. Ingalls, *Control theory and systems biology*. 2010: MIT Press.
10. Nweke, C., G. Okpokwasili, and P. Harcourt, Kinetics of growth and phenol degradation by Pseudomonas species isolated from petroleum refinery wastewater. Int. J. Biosci, 2014. **4**: p. 28-37.

Appendix:

1) GA main File (calling the function for calculation of error between simulated and experimental response)

```
    clear all
    clc
    close all
 options=gaoptimset('Generations',100,'PopulationSize',100);%% GA
INPUT
[parmest,fval1]=ga(@funcdeff1,10,[],[],[],[],[0 0 0 0 0 0 0 0 0 0
0],[30 30 30 30 30 30 30 30 30 10],[],[],options);  %% NUMBER OF
PARAMETERS-10,RANGE [0-30][0-10].

tim=0:0.1:10;
p=1;

a=parmest;  %% FINAL OPTIMIZED PARAMETERS
ansss=fval1;  %% FINAL RMSE

b=[1;1;1;1;1;1;1;1;1;1];  %% MICHALIS MENTEN KINETIC CONSTANTS
%% TIME (EXPERIMENTAL) %%
time0p=[0   593 1585    2581    3584    4592    5600    6604
7599    8598    9604    10602   11604   12610   13615   14608
15612   16614   17617   18618   19623   20625   21617   22610
23614   24615   25619   26613   27619   28618   33246   34244
35244   36247   37244   38240   39236   40238   41242   42250
43254   44254   45255   46251   47248   48243   49240   50236
51237   52236   53237   54240   55245   56243   57240   58230
59230   60224   61230   62230   63230   64231   65235   66239
67231   68229   69222   70270   71268   72222   73227   74227
75224   76226   77226   78235   79245   80241   81239   82237
83237   84244   85247   86247   87248   88240   89281   90273
91267   92263   93260   94257   95258   96261   97254   98252
99255   100253  101264  102268  103268  104264  105266  106263
107265  108263  109256  110251  111255  112260  113264  114270
115283  116275  117278  118274  119279  120282  121287  122281
123276  124275  125271  126266  127264  128265  129270  130273
131280  132286  133295  134296  135289  136282  137290  138293
139297  140293  141284  142280  143278  144272  145284  146283
147278  148276  149277  150280  151292  152288];

time0=time0p';

%% EXPERIMENTAL DATA %%

runningfilep=[2.0292    1.9588  2.1023  2.0052  1.9076  2.1103
2.1672  2.2059  2.1334  2.0402  2.0017  2.0646  1.9439  2.111
2.1061  2.0599  2.102   2.1789  2.1617  2.007   2.0008  2.0575
2.0292  2.0042  2.0283  2.0492  2.0079  2.0449  2.0801  2.2619
2.1349  2.2735  2.3093  2.3024  2.2702  2.1782  2.273   2.3253
2.3583  2.4576  2.5205  2.7467  2.8105  3.0847  3.392   3.8749
4.218   4.7197  5.1458  5.5666  5.7204  5.6265  5.6524  5.5894
5.5601  5.6597  5.4443  5.294   4.9939  4.7888  4.6036  4.4199
```

```
4.3426  4.1206  4.099   3.7794  3.8185  3.5738  3.418   3.3465
3.1317  2.9703  2.9262  2.7815  2.5785  2.4807  2.4383  2.3095
2.3975  2.4463  2.3658  2.3437  2.3518  2.3792  2.472   2.2511
2.3146  2.3406  2.3083  2.3049  2.2528  2.3895  2.3419  2.4758
2.3901  2.4221  2.2158  2.4121  2.2536  2.2548  2.427   2.311
2.4366  2.1515  2.2847  2.3155  2.3661  2.3697  2.2797  2.3599
2.2761  2.3582  2.3262  2.3162  2.4061  2.4523  2.4411  2.4938
2.4529  2.4856  2.5407  2.4909  2.4679  2.5859  2.5657  2.4855
2.4545  2.5002  2.5012  2.542   2.4291  2.4354  2.5512  2.5835
2.4666  2.5122  2.5122  2.6541  2.5217  2.6065  2.4878  2.6352
2.5355  2.7398  2.5716  2.5563  2.5064  2.6139  2.5281
2.6194];%cluster 3
 runningfile=runningfilep';
         time1=time0-time0(1);
        time = time1/(1000*60);
%         %% NORMALIZING THE DATA FOR CALCIUM RESPONSE
%              for i0=4:2:4;
                 runningbasemeanst = runningfile;
                 runningbasemean = sum(runningbasemeanst(1:25))/25;
                 runningcalnorm = runningfile/runningbasemean;
                 runningcalnormnew = runningcalnorm;
%
%              end
        exp = runningcalnormnew;
        expp=exp(32:end);
        expp1=expp-expp(1);
        timep=time(32:end);
        timep1=timep-timep(1);
        intval=[0;0;0];          %% INITIAL CONDITIONS
        %intval=[0;0;0;0];
        [tt,fval]=ode45(@(t,x) model5(t,x,a,b),tim,intval); %% ODE
SOLVER

 xap=fval(:,3);              %% SIMULATED VALUE
 calsim = interp1(tim,xap,timep1,'pchip');   %% SIMULATED
(INTERPOLATED VALUE)

error = abs(calsim-expp1);

err1=rms(error)          % FINAL RMSE
%
%% PLOTTING %%
       l=0:3.5;
       figure(2)
       subplot(1,2,1)
       %
        plot(timep1,expp1,'<m','linewidth',2)
        hold on
        plot(tt,fval(:,3),'b','linewidth',2)
        hold on
        plot(timep1,error,'c','linewidth',2)
        hold on
        title('CONC vs. TIME','fontsize',26)
        xlabel('TIME(min)','fontsize',26);
        ylabel('CONCENTRATION','fontsize',26);
         axis([0 2 -0.5 5])
        legend('Experimental','Simulated','Error','fontsize',26);
```

```
%          figure(2)
           subplot(1,2,2)
         plot(expp1,calsim,'+','linewidth',2)
         hold on
         %axis ([0 3.5 0 3.5])
         %axis([0 0.3 0 0.3])
         axis([0 1.5 0 1.5])
%          axis tight
         plot(l,l,'k','linewidth',2)
         xlabel('EXPERIMENTAL','fontsize',26)
         ylabel('SIMULATED','fontsize',26)
         legend('SIMULATED vs. EXPERIMENTAL','45deg
line','fontsize',26)
         title('SIMULATED vs. EXPERIMENTAL','fontsize',26)
%
         figure(3)
         plot(timep1,expp1,'<m','linewidth',2)
         hold on
         plot(tt,fval(:,3),'b','linewidth',2)
         hold on
         plot(timep1,error,'c','linewidth',2)
         hold on
         title('CONC vs. TIME','fontsize',26)
         xlabel('TIME(min)','fontsize',26);
         ylabel('CONCENTRATION','fontsize',26);
         axis([0 2 -0.5 5])
         legend('Experimental','Simulated','Error','fontsize',26);
         hold on
display(a,'optimum parameters are /n');
display(ansss,'Final Error /n');
```

2) Funcdeff1 file (Function File), calculation of error

```
function err=funcdeff1(Pact)        %% INPUT PARAMETERS FROM GA
     %% TIME
 time00p=[0  593 1585    2581    3584    4592    5600    6604
7599    8598    9604    10602   11604   12610   13615   14608
15612   16614   17617   18618   19623   20625   21617   22610
23614   24615   25619   26613   27619   28618   33246   34244
35244   36247   37244   38240   39236   40238   41242   42250
43254   44254   45255   46251   47248   48243   49240   50236
51237   52236   53237   54240   55245   56243   57240   58230
59230   60224   61230   62230   63230   64231   65235   66239
67231   68229   69222   70270   71268   72222   73227   74227
75224   76226   77226   78235   79245   80241   81239   82237
83237   84244   85247   86247   87248   88240   89281   90273
91267   92263   93260   94257   95258   96261   97254   98252
99255   100253  101264  102268  103268  104264  105266  106263
107265  108263  109256  110251  111255  112260  113264  114270
115283  116275  117278  118274  119279  120282  121287  122281
123276  124275  125271  126266  127264  128265  129270  130273
131280  132286  133295  134296  135289  136282  137290  138293
139297  140293  141284  142280  143278  144272  145284  146283
147278  148276  149277  150280  151292  152288];

time000=time00p';
```

```
%% CLUSTER DATA-EXPERMENTAL
  runningfileep=[2.0292   1.9588  2.1023  2.0052  1.9076  2.1103
2.1672  2.2059  2.1334  2.0402  2.0017  2.0646  1.9439  2.111
2.1061  2.0599  2.102   2.1789  2.1617  2.007   2.0008  2.0575
2.0292  2.0042  2.0283  2.0492  2.0079  2.0449  2.0801  2.2619
2.1349  2.2735  2.3093  2.3024  2.2702  2.1782  2.273   2.3253
2.3583  2.4576  2.5205  2.7467  2.8105  3.0847  3.392   3.8749
4.218   4.7197  5.1458  5.5666  5.7204  5.6265  5.6524  5.5894
5.5601  5.6597  5.4443  5.294   4.9939  4.7888  4.6036  4.4199
4.3426  4.1206  4.099   3.7794  3.8185  3.5738  3.418   3.3465
3.1317  2.9703  2.9262  2.7815  2.5785  2.4807  2.4383  2.3095
2.3975  2.4463  2.3658  2.3437  2.3518  2.3792  2.472   2.2511
2.3146  2.3406  2.3083  2.3049  2.2528  2.3895  2.3419  2.4758
2.3901  2.4221  2.2158  2.4121  2.2536  2.2548  2.427   2.311
2.4366  2.1515  2.2847  2.3155  2.3661  2.3697  2.2797  2.3599
2.2761  2.3582  2.3262  2.3162  2.4061  2.4523  2.4411  2.4938
2.4529  2.4856  2.5407  2.4909  2.4679  2.5859  2.5657  2.4855
2.4545  2.5002  2.5012  2.542   2.4291  2.4354  2.5512  2.5835
2.4666  2.5122  2.5122  2.6541  2.5217  2.6065  2.4878  2.6352
2.5355  2.7398  2.5716  2.5563  2.5064  2.6139  2.5281  2.6194];

runningfileee=runningfileep';

        time111=time000-time000(1);
        timeee = time111/(1000*60);

         %% NORMALIZING THE DATA FOR CALCIUM RESPONSE
        %              for i0=3:2:5;
 runningbasemeansttt = runningfileee;
runningbasemeannn = sum(runningbasemeansttt(1:25))/25;
runningcalnormmm =    runningfileee/runningbasemeannn;
runningcalnormnewww = runningcalnormmm;
 %              end
         exppp = runningcalnormnewww;
        expppp=exppp(32:end);
        expp111=expppp-expppp(1);
        timeppp=timeee(32:end);
        timep111=timeppp-timeppp(1);
        intvalll=[0;0;0];
        %intvalll=[0;0;0;0];
        %Pest=[1;1;1;1;1;1;1;1];
        Pest=[1;1;1;1;1;1;1;1];
         timmm=0:0.1:10;
%          timmm=timee-Pact(11);

[~,Yvalll]=ode45(@(t,x)model5(t,x,Pact,Pest),timmm,intvalll);
xappp=Yvalll(:,3);
calsimmm = interp1(timmm,xappp,timep111,'pchip');
errorrr = abs(calsimmm-expp111);
         %% RMS ERROR %%
        err=rms(errorrr);
```

3) Model file (function), solution of ODE models having kinetic mechanisms :

```
function f = model5( t,x,p,pb)

% MONOD MODEL

f=zeros(3,1);

f(1)=p(1)+(p(3)*x(2)/(pb(3)+x(2)))-(p(2)*x(1)/(pb(2)+x(1)))-
(p(4)*x(1)/(pb(4)+x(1)));
f(2)=(p(4)*x(1)/(pb(4)+x(1)))+(p(5)*x(3)/(pb(5)+x(3)))-
(p(3)*x(2)/(pb(3)+x(2)))-(p(6)*x(2)/(pb(6)+x(2)));
f(3)=(p(7)/(1+p(9)*power(x(1),p(10))))+(p(6)*x(2)/(pb(6)+x(2)))-
(p(8)*x(3)/(pb(8)+x(3)))-(p(5)*x(3)/(pb(5)+x(3)));
```

4) Sensitive Analysis:

```
clear all
clc
close all

%k1=4.5173;k_1=3.0493;k2=7.6822;k_2=0.0376;k3=2.4952;k_3=29.9899;k4
=29.9853;k_4=5.9210;k5=2.2221;q=9.993;
%% MODEL 1 DATA FROM GA
%k1=8.2734;k_1=18.5615;k2=8.2389;k_2=1.7819;k3=11.7065;k_3=0.022;k4
=29.9976;k_4=3.8641;k5=7.0948;q=9.9786;
%% MODEL 2 DATA FROM GA
k1=11.7155;k_1=11.1817;k2=13.1449;k_2=29.7656;k3=14.2503;k_3=0.0625
;k4=1.6875;k_4=0.0404;k5=29.9845;k_5=5.1562;k6=24.7968;q=9.9938;
%% MODEL 3 DATA FROM GA
tim=0:0.1:10;
a1=floor(q-ceil(0.5*q));
a2=ceil(1.5*q);

if(a1<0)
    a1=0;
end
if(a2>10)
    a2=10;
end
h=(a2-a1)/10;
% intvals=[0;0;0];
% b=[6.0399;2.6456];                    %% VARYING CONSTANTS [K3,Q]
% ap=parmest;
% [a1,b1]=size(parmest);
 %%%[LOWERBOUND,STEPSIZE,UPPERBOUND]=
range=[a1,h,a2]

p=1;
for i=a1:h:a2
% a=parmest;%(1:a1,1:b1-1) i];%0.6715];
% ansss=fval1;
%tim=t-a(11);
```

```
%a=[22.1008    2.5975    0.0735   40.1613   44.8795   35.9338
37.5373   16.8255    1.5766   24.9824];%    0.1618

%a=[i 5 5 5 5 5 5 5 5];
%a=[4.4217  3.2649  3.6485  3.6074  5.8833  0.0036  29.9982 0.8049
0.2708  i];

%a=[k1 k_1 k2 k_2 k3 k_3 k4 k_4 k5 q]; %%% FOR MODEL 1&2
a=[k1 k_1 k2 k_2 k3 k_3 k4 k_4 k5 k_5 k6 i];  %%% FOR MODEL 3

%b=[1;1;1;1;1;1;1;1];
%         % SAVING THE TIME DATA (TIMING DATA FOR X AXIS)
%         time0 = runningfile(:,2);
time0p=[0   593 1585    2581    3584    4592    5600    6604
7599    8598    9604    10602   11604   12610   13615   14608
15612   16614   17617   18618   19623   20625   21617   22610
23614   24615   25619   26613   27619   28618   33246   34244
35244   36247   37244   38240   39236   40238   41242   42250
43254   44254   45255   46251   47248   48243   49240   50236
51237   52236   53237   54240   55245   56243   57240   58230
59230   60224   61230   62230   63230   64231   65235   66239
67231   68229   69222   70270   71268   72222   73227   74227
75224   76226   77226   78235   79245   80241   81239   82237
83237   84244   85247   86247   87248   88240   89281   90273
91267   92263   93260   94257   95258   96261   97254   98252
99255   100253  101264  102268  103268  104264  105266  106263
107265  108263  109256  110251  111255  112260  113264  114270
115283  116275  117278  118274  119279  120282  121287  122281
123276  124275  125271  126266  127264  128265  129270  130273
131280  132286  133295  134296  135289  136282  137290  138293
139297  140293  141284  142280  143278  144272  145284  146283
147278  148276  149277  150280  151292  152288];

time0=time0p';

runningfilep=[1.7947    1.7066  1.795   1.7653  1.6533  1.7034
1.7993  1.8415  1.8215  1.7106  1.6945  1.7219  1.6706  1.7545
1.7825  1.6893  1.8103  1.8798  1.8723  1.673   1.7675  1.7778
1.7967  1.7977  1.7476  1.8274  1.8982  1.8357  1.8104  1.852
1.7904  1.8549  1.8217  2.0381  3.3923  4.3831  5.2303  6.1928
6.7607  7.1215  7.3803  7.4975  7.4826  7.4902  7.2023  7.1245
6.9317  6.8244  6.5528  6.3874  6.1029  5.8338  5.6522  5.3505
5.1337  4.9923  4.5175  4.26    3.9181  3.7117  3.4205  3.2369
2.947   2.7876  2.5925  2.2879  2.3618  2.2923  2.2255  2.1878
2.1046  1.966   1.9223  1.8908  1.9208  1.8792  1.8789  1.8218
2.0128  1.8937  1.9125  1.8636  1.8468  1.8123  1.9642  1.8496
1.8571  1.9047  1.9319  1.9432  1.8433  1.9973  1.8651  2.0078
1.9786  1.9334  1.8098  1.9474  1.8606  1.8842  1.9102  1.9863
2.0104  1.8516  1.8998  1.8432  1.915   1.8872  1.8912  1.9287
1.8878  1.8659  1.8735  1.8711  1.977   1.8822  1.9809  1.9356
1.9965  1.9994  2.0321  1.9748  2.0042  2.128   2.0995  2.1155
2.1673  2.0542  2.1394  2.0851  2.0182  2.0487  2.173   2.1385
2.1122  2.1519  2.0667  2.2132  2.2644  2.267   2.1653  2.1092
1.9778  2.3012  2.0454  2.1146  2.0543  2.1796  2.0096  2.1384];

runningfile=runningfilep';
```

```
time1=time0-time0(1);
time = time1/(1000*60);
 %          %% NORMALIZING THE DATA FOR CALCIUM RESPONSE
%
%               for i0=4:2:4;
                   runningbasemeanst = runningfile;
                  runningbasemean = sum(runningbasemeanst(1:25))/25;
                  runningcalnorm = runningfile/runningbasemean;
                  runningcalnormnew = runningcalnorm;
%
%               end
         exp = runningcalnormnew;
        expp=exp(32:end);
        expp1=expp-expp(1);
        timep=time(32:end);
        timep1=timep-timep(1);
         %intval=[0;0;0];
        intval=[0;0;0;0];
        [tt,fval]=ode45(@(t,x) model9(t,x,a),tim,intval);
         xap=fval(:,4);
        calsim = interp1(tim,xap,timep1,'linear');
         error = abs(calsim-expp1);
        err1=rms(error);
        errvalv(p)=err1;
        parm1(p)=i;
        l=0:3.5;
%
         subplot(4,3,p)
         plot(timep1,expp1,'<m','linewidth',2)
                   hold on
         plot(tt,fval(:,4),'b','linewidth',2)
         hold on
         plot(timep1,error,'c','linewidth',2)
         hold on
         hold on
        xlabel('TIME(min)')%,'fontsize',26);
        ylabel('CONC')%,'fontsize',26);
        axis([0 2 -0.5 5])
        p=p+1;
end
%
         title('CONC vs. TIME')%,'fontsize',26)
         legend('Experimental','Simulated','Error')%,'fontsize',26);
         [parm1' errvalv']
%xlswrite('e.xls',err1');
% display(a,'optimum parameters are /n');
% display(ansss,'Final Error /n');
```