

## Contents

Introduction .....	3
Online Availability .....	4
Download Locally .....	5
Install R and RStudio .....	5
Download BioModME from GitHub.....	5
Install R Repositories.....	6
Start Application .....	7
Pull Docker Image .....	8
Download Docker.....	8
Pull/Run From Docker.....	10
Mathematical Derivations .....	14
Equations .....	14
Chemical Based Equations .....	14
Input/Output.....	19
Units .....	25
Unit Conversion .....	25
Converting Compound Units .....	26
Unit Definitions .....	26
Creating A Custom Law .....	30
Creating a Custom Equation .....	34
SBML Import .....	35
Export Model .....	36
Coding Languages .....	36
MATLAB.....	36
R .....	37
Storage Languages .....	37
rds .....	38
SBML .....	38
Model Information Tables .....	39
LaTeX Document .....	39
Differential Equations .....	40

Basic Tutorial - Introduction .....	42
Constructing Model .....	43
Define Variables.....	43
Create Reactions .....	46
Equation 1 - Law of Mass Action, Reversible.....	46
Equation 2 - Michaelis-Menten Kinetics.....	47
Equation 3 - Law of Mass Action, Unidirectional.....	48
Equation 4 - Law of Mass Action, Unidirectional.....	49
Equation 5 - Synthesis by Factor.....	49
Equation 6 - Degradation by Rate.....	51
Compartment Input/Output .....	52
Parameters.....	53
Differential Equations .....	55
Solve Model .....	55
Visualize Model.....	57
Export.....	60
Latex.....	60
Multicompartment Tutorial .....	62
Constructing Model .....	64
Create Compartments .....	64
Define Variables.....	65
Add Equations .....	67
Add Compartment Inputs/Outputs .....	68
Parameters.....	80
Differential Equations .....	82
Results.....	83
Solve Model .....	83
Visualization .....	85
Parameter Estimation Tutorial .....	87
Build Model.....	88
Visualize Model.....	91
Perform Parameter Estimation.....	93

## Introduction

BioModME is an R based web application built using the Shiny framework for expedient biological system modeling and simulations. It provides the users with an all-encompassing and code-free computational model building tool to define reaction networks, develop and simulate computational models, and analyze the behavior of complex biological systems. This web application was developed with modules that allow the users to systematically build their own models for a biological system using the graphical user interface (GUI), without requiring the knowledge of a computer programming language. The program converts the reaction network to its appropriate mathematical equations (differential and algebraic equations; DAEs), allowing the non-expert users, including bench biologists and clinicians, to avoid all tedious mathematical derivations of the equations needed to describe the biological system. R visualization packages are used to present the simulation results in meaningful ways in the GUI with options to export the model information and results for other applications. The model can also be exported to other programming languages, such as MATLAB, R and Python.

BioModME was built using the R programming language and Shiny framework. Specific elements of the application have been developed with the use of customized HTML, CSS, JavaScript, and Python. As such, the application is provided through a variety of methods for the user's convenience including online deployed versions, source code to be locally used, and as a docker image for local use avoiding dependencies.

### Note

Using the online server is a good option that requires no download of application or knowledge of running R. The server, currently, does not offer the ability to save models/load models from a user's repository in the cloud. It does offer the ability to download model save files locally and upload local files.

## Online Availability

BioModME (stable version) is hosted on webservers at the Medical College of Wisconsin (MCW) by the Clinical & Translational Science Institute (CTSI) of Southeast Wisconsin. It can be accessed at:

<https://biomodme.ctsi.mcw.edu/>

Additionally, a backup version (developer version) of the application can be accessed through the following:

[https://jwomack7512.shinyapps.io/Model\\_Builder/](https://jwomack7512.shinyapps.io/Model_Builder/)

The developer version is the first author's personal shinyapps account and will be active but has a 15-minute timeout period that users should be aware of.

## Download Locally

BioModME can be run on your local system using R.

### Install R and RStudio

R and RStudio will need to be installed (tested locally with R 4.1.1 and RStudio 1.4.1717). Please check CRAN for the installation of R (<https://cran.r-project.org>).

For the installation of RStudio, go to and follow instructions from <https://www.rstudio.com>.

### Download BioModME from GitHub

#### Option 1 - Using Git Bash:

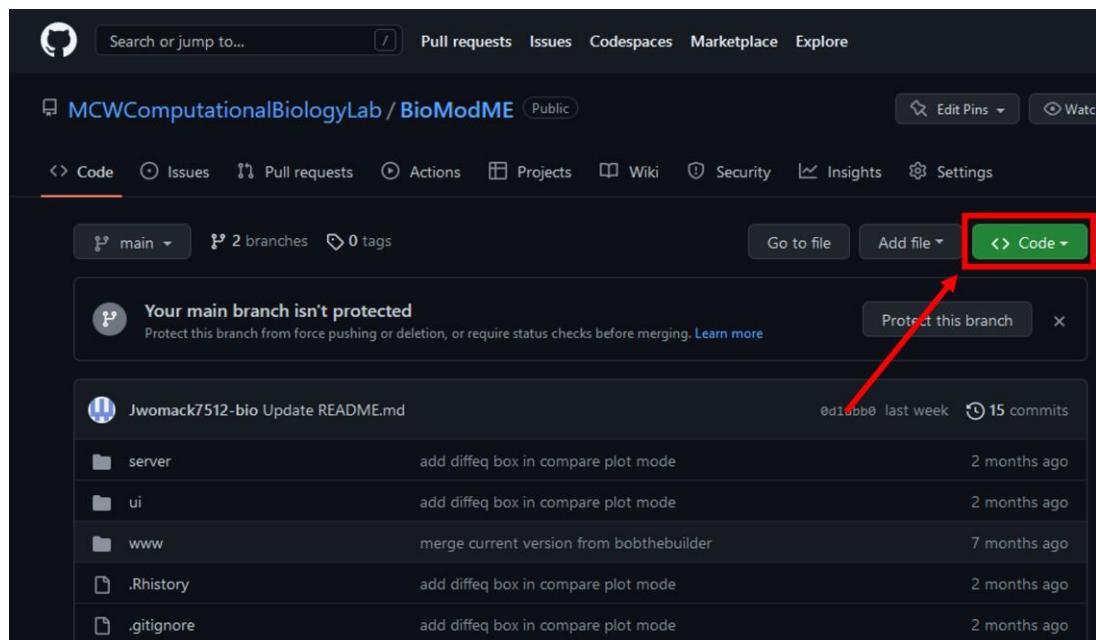
1. Open Git Bash
2. Change the current working directory to the location where you want the cloned directory.
3. Enter the following command:

```
git clone https://github.com/MCWComputationalBiologyLab/BioModME.git
```

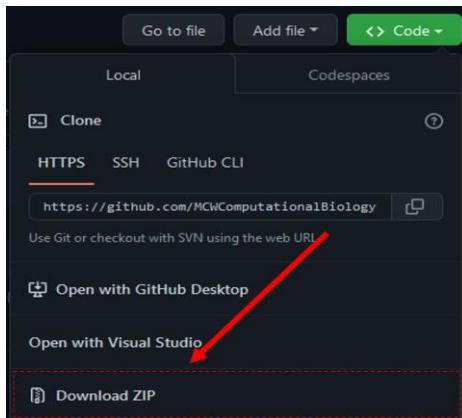
4. Press **Enter** to create your local clone of the application.

#### Option 2 - Direct Download:

1. Navigate to <https://github.com/MCWComputationalBiologyLab/BioModME>.
2. Click the green button "<> Code".



3. Press download ZIP



4. Unzip folder in desired location.

## Install R Repositories

BioModME uses a vast number of standard R repositories that will need to be downloaded. Open up the UI.R file from the downloaded file. This script loads all the needed libraries for the application to properly run. To install these repositories, you can run the following:

```
# Vector of package names to install
load.lib <- c("shinydashboard", "bs4Dash", "shiny", "ggplot2", "gridExtra",
"shinythemes", "shinyWidgets", "shinyjs", "DT", "tidyverse", "dplyr",
"rhandsontable", "data.table", "ggpmisc", "colourpicker", "shinyBS", "shinyjqui",
"bsplus", "deSolve", "plotly", "Deriv", "viridis", "ggpubr", "shinycssloaders",
"waiter", "fresh", "readxl", "minpack.lm", "measurements", "qdapRegex", "XML",
"xml2", "katex", "reshape2", "clipr", "jsonlite")
# Remove any packages from list that are already installed
install.lib <- load.lib[!load.lib %in% installed.packages()]

# Install all packages in install list
for (lib in install.lib) install.packages(lib, dependencies = TRUE)
sapply(load.lib, require, character = TRUE)
```

All the required packages should now be installed.

### Note

Current versions of RStudio will have a dropdown header telling you that you do not have all the packages installed and will provide you a button to click to install them all.

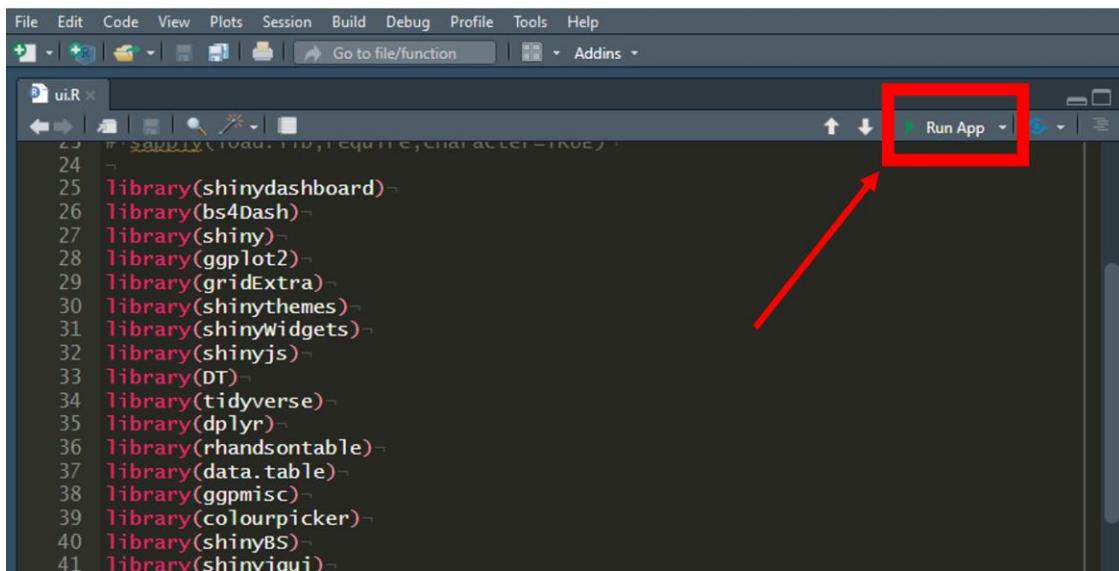
## Start Application

The application can be started from base R by navigating to the application home folder and running the following:

```
shiny::runApp()
```

A more common way to run the app is to run it directly from RStudio. This is done by:

1. Open either “ui.R” or “server.R” in RStudio
2. Press bottom in top right of script that says “Run App”.



## Pull Docker Image

The application has been created as a docker image for user convenience if they wish to run the app locally without installing R or worrying about other possible dependencies.

## Download Docker

In this section, we will download docker to our computer. This will install the desktop app but what we really need is the docker terminal commands to be added. If you have a working version of Docker on your system, skip to the next section.

1. Go to <http://www.docker.com/products/docker-desktop>
2. Download docker for your respective system. This tutorial follows the windows download.



### The Docker Subscription Service Agreement has been updated.

3. Go to your downloads folder and begin the system installation of docker. This download will take a few minutes.
4. A configuration popup will appear. Keep “Install required Windows components for WSL 2” checked. You can uncheck the desktop shortcut option if you want.



## Configuration

- Install required Windows components for WSL 2
- Add shortcut to desktop

Ok

5. Once download is complete a message will appear to restart your system. Perform computer restart.
6. On restart you may get a notification that “WSL 2 installation is incomplete”. If this is the case, you will need to install this update or the docker image will not run. Follow the link on the popup, it takes you to the following link: <https://docs.microsoft.com/en-us/windows/wsl/install-manual#step-4--download-the-linux-kernel-update-package>. Download and install the kernel from this link.

**Step 4 - Download the Linux kernel update package**

1. Download the latest package:

- WSL2 Linux kernel update package for x64 machines

**Note**

If you're using an ARM64 machine, please download the ARM64 package instead. If you're not sure what kind of machine you have, open Command Prompt or PowerShell and enter: `systeminfo | find "System Type"`. **Caveat:** On non-English Windows versions, you might have to modify the search text, translating the "System Type" string. You may also need to escape the quotations for the find command. For example, in German `systeminfo | find "Systemtyp"`.

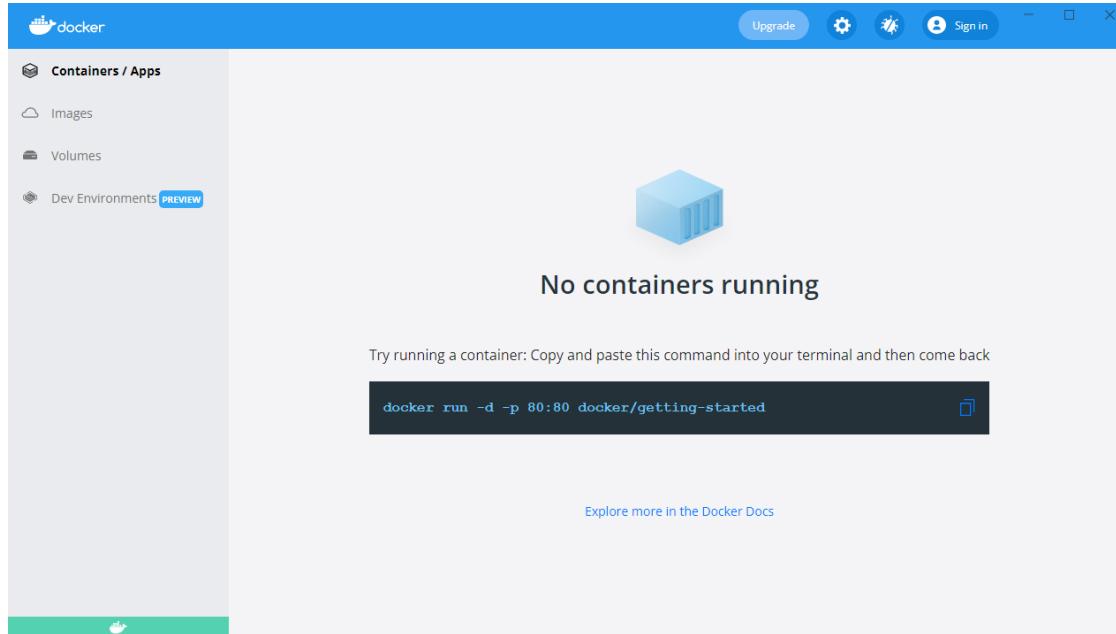
2. Run the update package downloaded in the previous step. (Double-click to run - you will be prompted for elevated permissions, select 'yes' to approve this installation.)

Once the installation is complete, move on to the next step - setting WSL 2 as your default version when installing new Linux distributions. (Skip this step if you want your new Linux installs to be set to WSL 1.)

**Note**

For more information, read the article [changes to updating the WSL2 Linux kernel](#), available on the [Windows Command Line Blog](#).

7. Restart computer again and open docker program. It should be fully installed and working now. A working program has a green color in the bottom left corner and lacks a huge error message in the center of the application.



Note: A Docker account does not need to be made in order to pull and run this program.

## Pull/Run From Docker

With docker installed, we can now use the command line to pull the program we desire, meaning we can download it to our computer from the docker cloud.

1. Open the computer's terminal (command prompt).
2. Type the below command to pull the application from the docker cloud (this takes approximately five minutes):

```
docker pull jwomack7512/biomodme
```

3. Type the following command to run the application:

```
docker run -dp 3838:3838 jwomack7512/biomodme
```

This step might need approval from windows defender or other antivirus program.

The screenshot shows a Windows Command Prompt window titled "Command Prompt - docker run -p 3030:3030 jwomack7512/data\_explorer". The window displays the following text:

```
Microsoft Windows [Version 10.0.19041.1288]
(c) Microsoft Corporation. All rights reserved.

H:\>docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
jwomack7512/data_explorer    latest   1a0012243d9a   29 hours ago  3.04GB

H:\>docker run -p 3030:3030 jwomack7512/data_explorer
R version 4.1.2 (2021-11-01) -- "Bird Hippie"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> shiny::runApp('/app', host = '0.0.0.0', port = 3030)
Loading required package: shiny

Listening on http://0.0.0.0:3030
```

4. The above command runs the program at a local host port. To access the app, open your web browser and type the following:

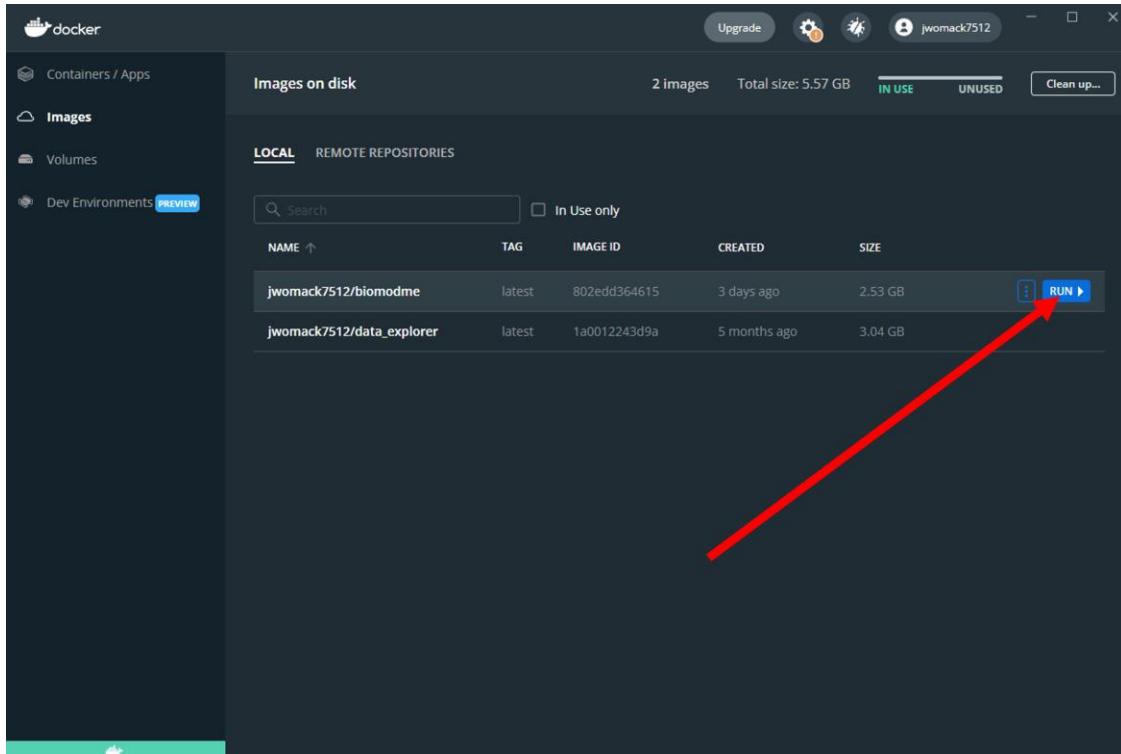
localhost:3838

5. The application should now load on the opened webpage.

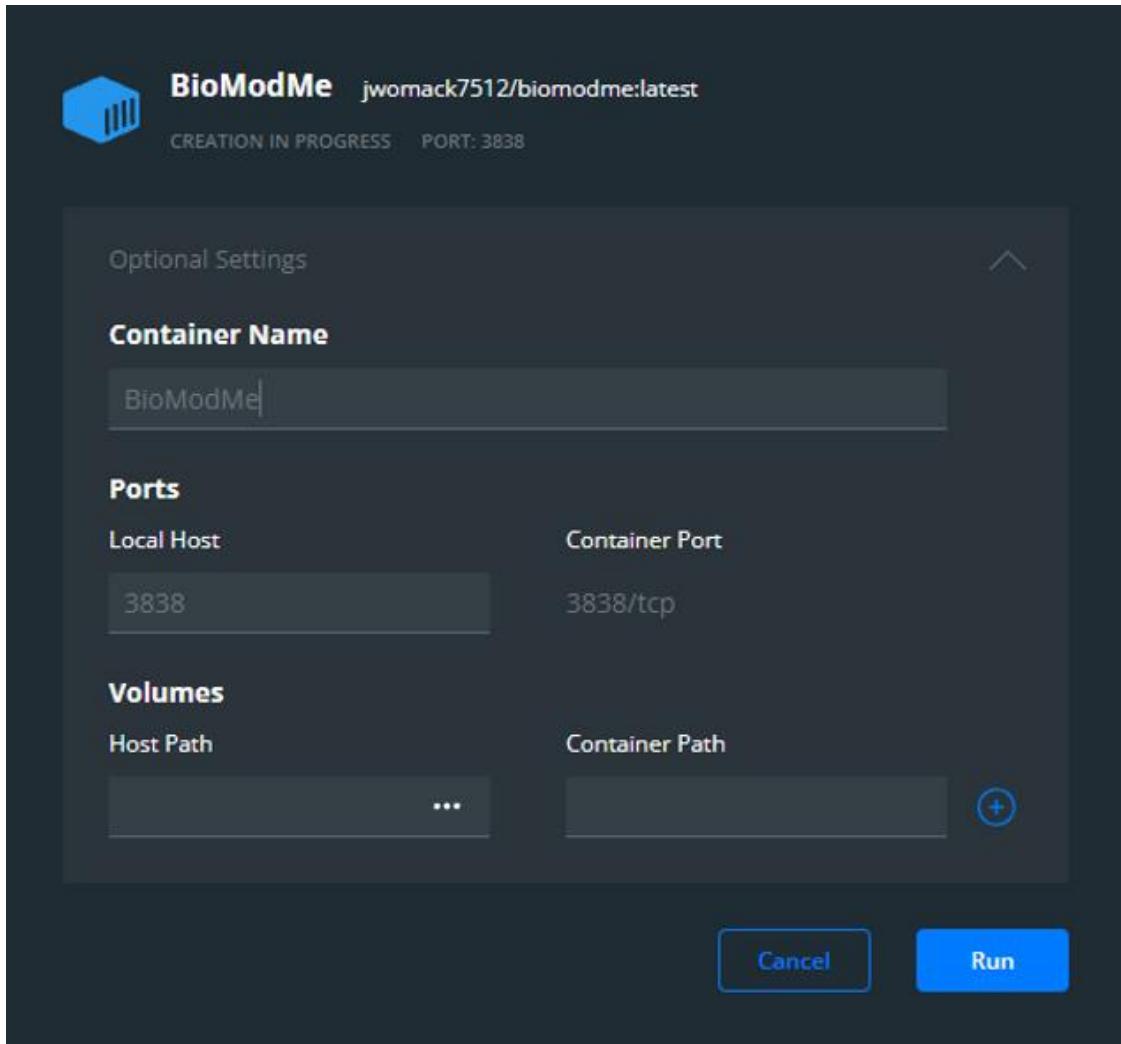
#### *Loading Application After Pull*

After the image has been pulled once, there is not a need to repull it. The image should be saved to your local docker program and can be loaded easily using the Docker Hub GUI.

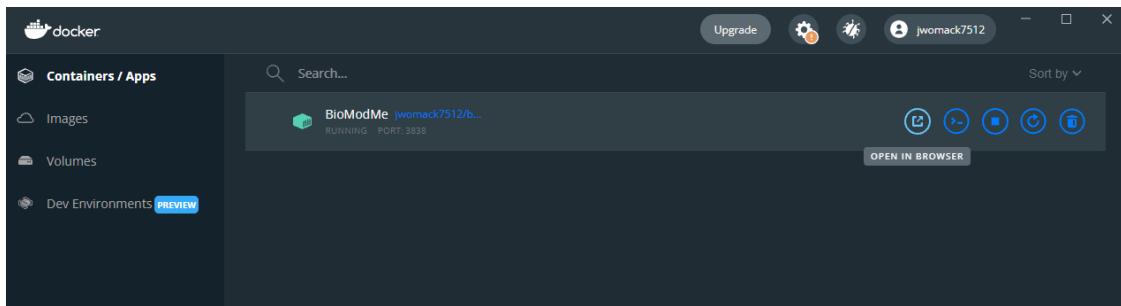
6. Open docker and navigate to the “images” tab. Here you should see a copy of the image you pulled in the previous step. This means its available without having to repull it.
7. Scroll over the image **jwomack7512/biomodme**. A “run” button will appear. Press that button.



8. A popup will appear. Click the dropdown arrow to show optional settings. Here we give the container the name “bioModMe”. If you do not give it a name, docker will randomly generate one. Type in “3838” in Local Host. This is necessary or the program will not load to the proper port the app specifies. Press the “Run” button.



9. Go to the “Containers/Apps” tab in Docker Hub. You should see the container you created with a note saying it is running at port 3838. You can access the app by simply clicking the “Open in Browser” button. Note: you may already have a container built from using the command line, meaning you could just run that one instead of rebuilding it in the previous steps.



In the above figure, if you double click the container the logs will open for the application. These can be useful for debugging and submitting errors.

## Mathematical Derivations

This section covers all the mathematical equations and derivations that run in the background of BioModME.

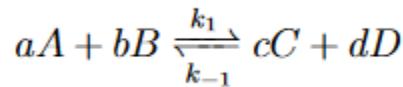
### Equations

This section contains the mathematical derivations used for the equation builders found in this program.

#### Chemical Based Equations

##### -Law of Mass Action

Chemical reactions are derived using the law of mass action. Given the following chemical reaction scheme:



where a, b, c, d are stoichiometric coefficients and A, B, C, D are the chemical species. The law is as follows:

$$-\frac{1}{a} \frac{d[A]}{dt} = -\frac{1}{b} \frac{d[B]}{dt} = \frac{1}{c} \frac{d[C]}{dt} = \frac{1}{d} \frac{d[D]}{dt} = k_1[A]^a[B]^b - k_{-1}[C]^c[D]^d$$

Where [X] denotes concentration of X. From the above law, each individual term can be derived below:

$$\frac{d[A]}{dt} = -a * k_1[A]^a[B]^b + a * k_{-1}[C]^c[D]^d$$

$$\frac{d[B]}{dt} = -b * k_1[A]^a[B]^b + b * k_{-1}[C]^c[D]^d$$

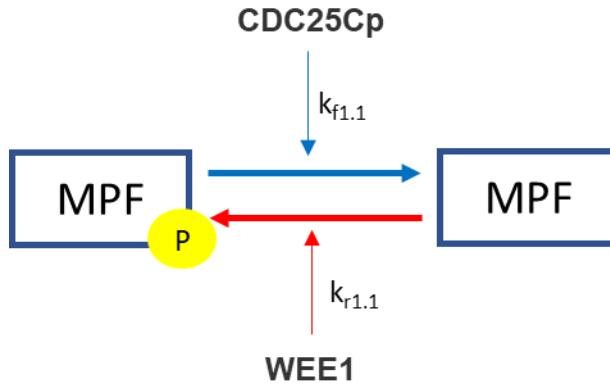
$$\frac{d[C]}{dt} = c * k_1[A]^a[B]^b - c * k_{-1}[C]^c[D]^d$$

$$\frac{d[D]}{dt} = d * k_1[A]^a[B]^b - d * k_{-1}[C]^c[D]^d$$

Each individual equation will be added to their overall flux differential equation in the differential equation solver section.

### -Law of Mass Action with Regulation

Mass action with regulation is for chemical reactions who have rate constants that are dependent on other species. An example is the regulation of inactive MPF to active MPF in the cell cycle by WEE1 and CDC25Cp. In this example, inactive MPF is phosphorylated by CDC25Cp and dephosphorylated by WEE1. These regulators are not directly affected in the differential equations in this situation while inactive and active MPF are. We have the same derivation as mass action except the rate constants are changed to reflect the regulation.



The regulators will take the respective place of the rate constants in the mass action derivation. In the mass action derivation above for [A] and [B] we have:

$$\frac{d[A]}{dt} = -a * k_1[A]^a[B]^b + a * k_{-1}[C]^c[D]^d$$

$$\frac{d[B]}{dt} = -b * k_1[A]^a[B]^b + b * k_{-1}[C]^c[D]^d$$

Adding regulator replaces the respective rate constant with the sum of its regulators and rate constants:

$$\frac{d[A]}{dt} = -a * \left( \sum k_{f_i} * [reg_i] \right) [A]^a[B]^b + a * \left( \sum k_{r_i} * [reg_i] \right) [C]^c[D]^d$$

$$\frac{d[B]}{dt} = -b * \left( \sum k_{f_i} * [reg_i] \right) [A]^a[B]^b + b * \left( \sum k_{r_i} * [reg_i] \right) [C]^c[D]^d$$

Using the above example (which is one reactant/product), we get the following result:

$$\frac{d[MPF_p]}{dt} = - (k_{f1.1} * [CDC25Cp]) * [MPF] + (k_{r1.1} * [WEE1]) * [MPF_p]$$

$$\frac{d[MPF]}{dt} = (k_{f1.1} * [CDC25Cp]) * [MPF] - (k_{r1.1} * [WEE1]) * [MPF_p]$$

### -Michaelis Menten

Michaelis Menten kinetics is a commonly used approximation for describing the rate at which enzymatic reactions occur. A single enzymatic reaction can be described by:



In the above, S is the substrate, E is the enzyme, and P is the product of the reactions. The model most commonly takes the following for describing the velocity of a reaction:

$$v = \frac{d[P]}{dt} = V_{max} \frac{[S]}{K_M + [S]} = (k_{cat} * [E]) \frac{[S]}{K_M + [S]}$$

where,

**V<sub>max</sub>** - Maximum Velocity

**E** - Enzyme Concentration

**S** - Substrate Concentration

**P** - Product Concentration

**K<sub>M</sub>** - Michaelis Menten Constant

**k<sub>cat</sub>** - Catalytic Rate of Enzyme Reaction

### [Synthesis](#)

Synthesis by rate has a species generated at a specific rate. This is useful for when the user does not particularly know what is causing the synthesis but knows how quickly it is being produced. It simply creates a constant rate of production.

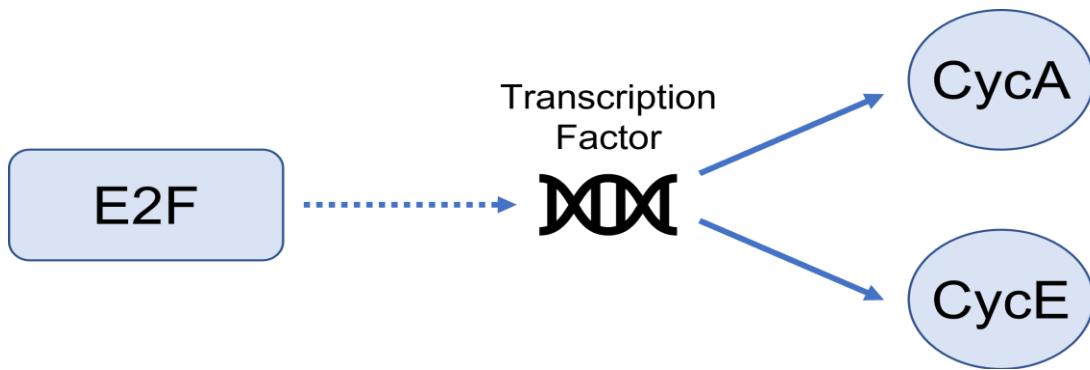
$$\frac{d[\text{Species}]}{dt} = k_{syn}$$

where,

**k<sub>syn</sub>** - synthesis rate of species

### [Synthesis by Regulation](#)

The “By Factor” law is used when a regulator directly influences the synthesis of a species but is not directly converted. An example of this is in the Cell Cycle when the molecule E2F activates transcription factors that synthesize Cyclin E and Cyclin A. If the cell contains no E2F, then there is no synthesis of Cyclin E or Cyclin A.



There is only one resultant mathematical flux from this type of synthesis and that is the species being synthesized. The factor causing the synthesis does not have its flux altered. The general resulting equations are:

$$\frac{d[Species]}{dt} = k_{syn} * [Factor]$$

$$\frac{[Factor]}{dt} = 0$$

where,

**k<sub>syn</sub>** - synthesis rate of species

**Species** - species that is being synthesized

**Factor** - factor that is causing the synthesis

In relation to the above E2F equation, the differential equations are:

$$\frac{d[CycA]}{dt} = k_{syn} * [E2F]$$

$$\frac{d[CycE]}{dt} = k_{syn} * [E2F]$$

$$\frac{d[E2F]}{dt} = 0$$

#### -Degradation

These reactions are dependent on a rate constant. Here we have two specific options where the degradation can be concentration dependent or not. Often degradations are concentration dependent on themselves, meaning there is more degradation at higher concentrations and vice versa. The following is the derivation for a concentration dependent degradation:

$$\frac{d[Species]}{dt} = -k_{deg} * [Species]$$

where,

**k<sub>deg</sub>** - degradation rate constant

**Species** - species being degraded

When the degradation is not concentration dependent, the resulting flux is:

$$\frac{d[Species]}{dt} = -k_{deg}$$

When products are made, they are the opposite sign of the above reactions. For example, for concentration dependent reactions with one degradation to two products we would have the resulting differential equations:

$$\frac{d[Species]}{dt} = -k_{deg} * [Species]$$

$$\frac{d[Product1]}{dt} = +k_{deg} * [Species]$$

$$\frac{d[Product2]}{dt} = +k_{deg} * [Species]$$

#### -Degradation by Enzyme

This degradation option is a basic enzyme reaction using Michaelis-Menten kinetics where the substrate is the species to be degraded and there often is no product. The resulting mathematical flux for the degraded substrate would be:

$$\frac{d[S]}{dt} = -V_{max} \frac{[S]}{K_m + [S]} = -(k_{cat} * [E]) \frac{[S]}{K_m + [S]}$$

where,

**V<sub>max</sub>** - Maximum Velocity

**E** - Enzyme Concentration

**S** - Substrate Concentration

**P** - Product Concentration

**K<sub>M</sub>** - Michaelis-Menten Constant

**k<sub>cat</sub>** - Catalytic Rate of Enzyme Reaction

There is an option to add a product (P), which results in the mathematical flux equations to break down to pure Michaelis-Menten kinetics for the product and substrate which would have the same equation above except with a positive sign:

$$\frac{d[S]}{dt} = -V_{max} \frac{[S]}{K_m + [S]} = -(k_{cat} * [E]) \frac{[S]}{K_m + [S]}$$

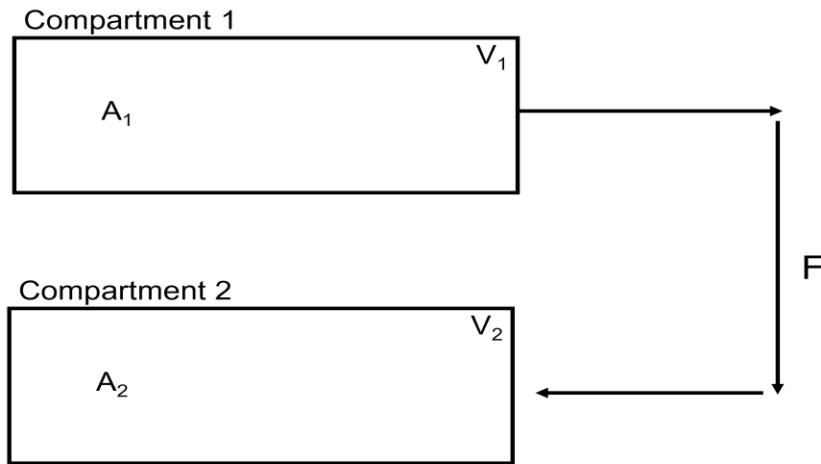
$$\frac{d[P]}{dt} = -V_{max} \frac{[S]}{K_m + [S]} = -(k_{cat} * [E]) \frac{[S]}{K_m + [S]}$$

## Input/Output

This section contains the mathematical derivations used for the input/output builders found in this program.

### -Flow Between

Flow between two or more compartments follow the mechanics described below. In the below figure, we have flow between two compartments. We have species  $A_1$  flowing out of compartment 1 into compartment 2 as  $A_2$ .



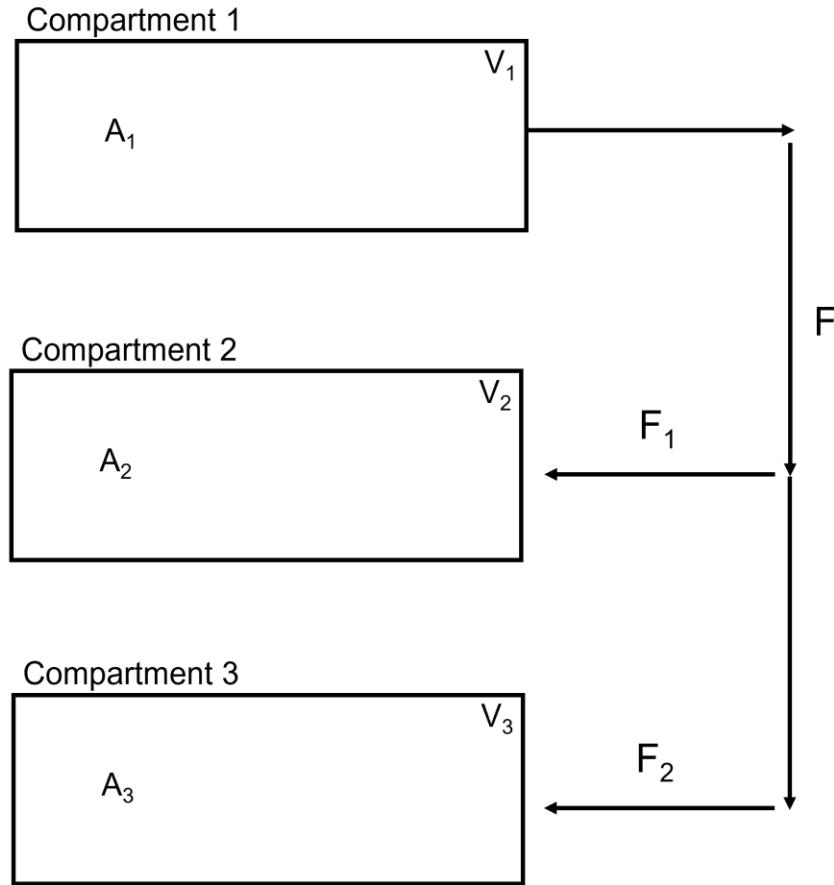
The flows from a compartment are derived as the flow rate ( $F$ ) multiplied by the concentration of the species leaving the compartment as a minus term. Flow to a compartment is the same derivation without the minus term. The flows in the above diagram would be derived as:

$$V_1 * \frac{d[A_1]}{dt} = -F * [A_1]$$

$$V_2 * \frac{d[A_2]}{dt} = F * [A_1]$$

### -Split Flow

Often, we need to split the flow from one compartment to multiple compartments. This option splits the flow from one compartment to go to multiple compartments. In the figure below, we see the flow from compartment 1 is split to go to compartment 2 and compartment 3.



The summation of the output flows will be equal to the input flow:

$$F_{out} = \sum_1^n F_{in}$$

Given the above figure this equation derives out to:

$$F = F_1 + F_2$$

The resulting flow differential equations derive as:

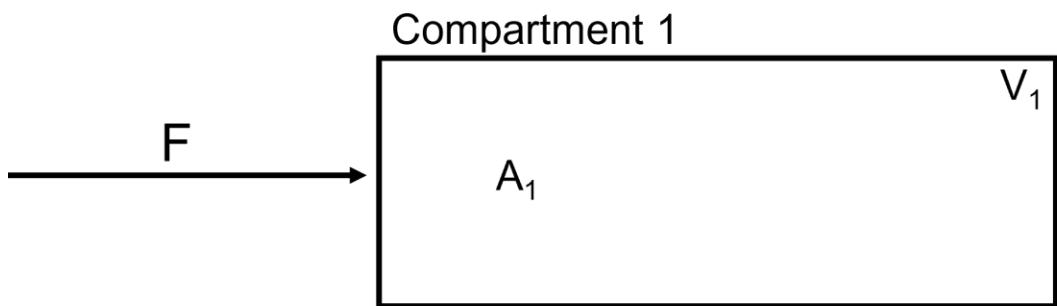
$$V_1 \frac{d[A_1]}{dt} = -F * [A_1]$$

$$V_2 \frac{d[A_2]}{dt} = -F_1 * [A_1]$$

$$V_3 \frac{d[A_3]}{dt} = -F_2 * [A_1]$$

### -Flow In

Constant flow into a compartment from an unspecified source. This could be something like an input flow into a model of species or a known constant flow from a source outside the scope of your created model. Flow In and Flow Out options can also be pieced together to create a customized Flow Between. In the Figure below, we see a constant flow,  $F$ , carrying species,  $A_1$ , into Compartment 1.

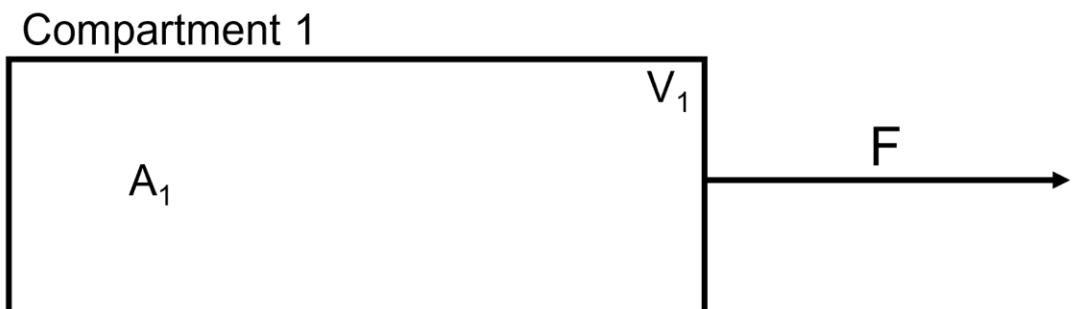


The mathematical derivation for this model input would be:

$$V_1 \frac{d[A_1]}{dt} = F * [A_1]$$

### -Flow Out

Constant flow out of a compartment that does not go to another compartment. Flow In and Flow Out options can be pieced together to create a customized Flow Between. In the Figure below, we see a constant flow,  $F$ , carrying species,  $A_1$ , out of Compartment 1.

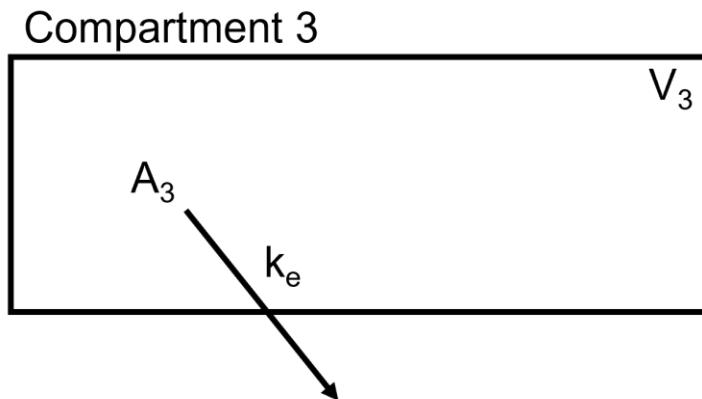


The mathematical derivation for this model output would be:

$$V_1 \frac{d[A_1]}{dt} = -F * [A_1]$$

### *-Clearance*

Clearance from a compartment is used when a species leaves the compartment and is removed from the model. In the example below, we have species  $A_3$  being cleared from compartment 3.



It takes the following derivation:

$$V_3 \frac{d[A_3]}{dt} = -k_e * [A_3] * V_3$$

where,

$k_e$  - Rate of clearance

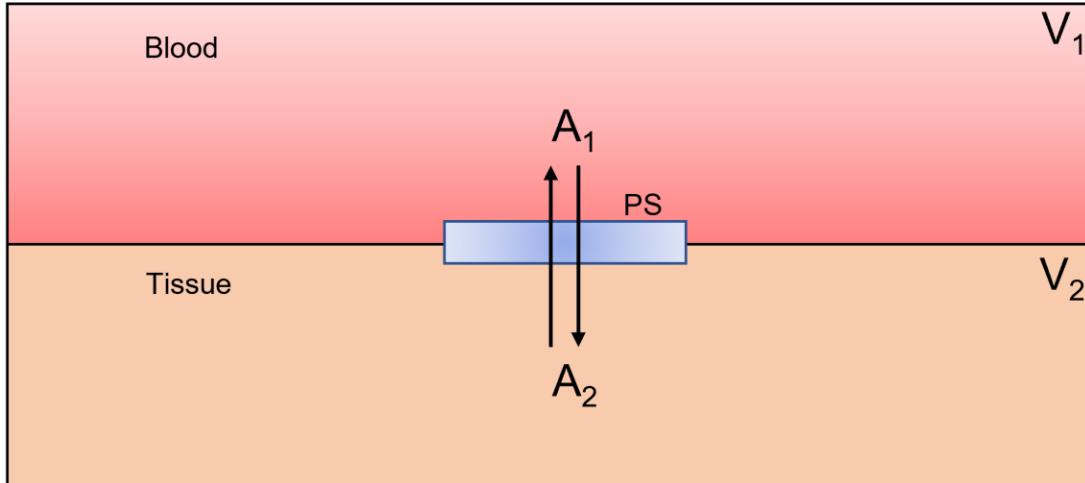
$A_3$  - Concentration of Species being cleared

$V_3$  - Volume of compartment that species is being removed from

### *-Simple Diffusion*

In simple diffusion, the substance is driven by concentration difference across a membrane. In the example below, we have a liver compartment that is separated into blood and liver compartments with a passive diffusion membrane for compound **A**.

## Liver



For the simple, one-dimensional case shown above, the mass flux through the membrane follows Ficks Law:

$$J = -D \frac{dC}{dx}$$

where,

**C** - Concentration at membrane location x

**D** - Diffusion coefficient of the molecule

We assume the concentration gradient across the membrane of thickness, x, falls linearly with x leading to:

$$\begin{aligned} \frac{dC}{dx} &= \frac{\Delta C}{\Delta X} = \frac{C_2 - C_1}{\Delta x} \\ J &= -D \frac{\Delta C}{\Delta x} = -D \frac{C_2 - C_1}{\Delta x} \end{aligned}$$

The rate of transfer (**f**) of a neutral compound across a biological membrane of surface area (**S**) is:

$$f = JS = -SD \frac{\Delta C}{\Delta x} = -PS(C_2 - C_1)$$

where,

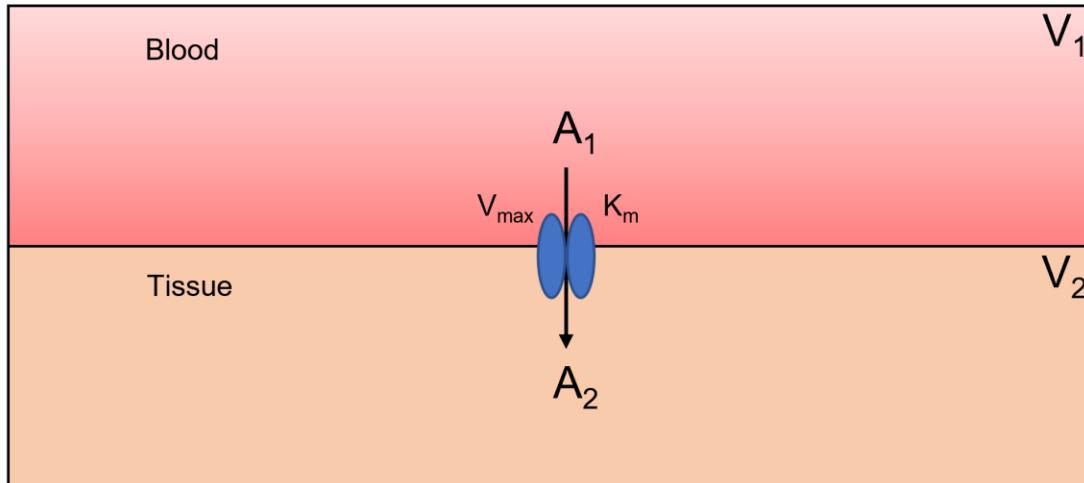
**PS** – Permeability surface area product parameter

Note: The **negative sign** on the right side of the equation indicates that the net transfer due to diffusion is in a direction away from the region with higher concentration.

### -Facilitated Diffusion

Facilitated diffusion is a type of diffusion in which the molecules move from the regions of higher concentration to the region of lower concentration assisted by a carrier molecule. It is a selective process, only allowing specific molecules to pass the gradient. In the below figure, we have a liver compartment that is separated into blood and liver compartments with a facilitated transporter for compound A. Note that this is a one-way transport.

Liver



For the derivation of the flux equation for facilitated diffusion, we take the following equation:

$$F = \frac{V_{max} * [A_1]}{K_m + [A_1]}$$

where,

$V_{max}$  - Maximum Velocity

$A_1$  - Substrate Concentration

$K_m$  - Affinity of how much substrate concentration needed to reach  $V_{max}$

The flux derivations for the above example would be:

$$V_1 \frac{d[A_1]}{dt} = -\frac{V_{max} * [A_1]}{k_m + [A_1]}$$

$$V_2 \frac{d[A_2]}{dt} = \frac{V_{max} * [A_1]}{k_m + [A_1]}$$

# Units

BioModME supports basic units for species and parameters in a model. The unit definition system is created from the R package “measurements” (<https://cran.r-project.org/web/packages/measurements/measurements.pdf>). This package supports several commonly found units. The user can define the units how they want and the application will convert them to a base unit in the backend for processing.

## Unit Conversion

To convert units, go to the units section of the data table for the variable. For example, we will look at a parameter table example. The third column in the table is units. By clicking the units square, we can change the unit type.

	Name	Value	Unit	Description
1	V_cell	1.00	l	Volume of compartment_1
2	k_f1	2.00	1/min	Forward rate constant for the reaction of A, B to C_1
3	k_r1	1.00	mol/(l*min)	Reverse rate constant for the reaction of A, B to C_1
4	Km_2	0.10	mol	Michelis Menten constant for enzymatic conversion of C_1 to C_2
5	kcat_2	3.00	1/min	Michelis Menten constant for enzymatic conversion of C_2 to C_2
6	k_f3	2.00	mol/(l*min)	Forward rate constant for the reaction of C_2 to I, P
7	k_f4	2.00	1/min	Forward rate constant for the reaction of I, Prot to I.Prot
8	k_syn5	0.50	1/min	Synthesis rate constant of A by factor Prot
9	k_d6	0.10	1/min	Degradation rate constant for I.Prot

The program will only allow you to convert to accepted units for that type. Meaning, for example, V\_cell has the units of liter, so it can only be changed to another unit of volume (such as mL). For a list of all units and their values, see the unit definition section. If you attempt to convert to a unit that is not allowed, a warning will appear.

	Name	Value	Unit	Description
1	V_cell	1.00	mol	Volume of compartment_1
2	k_f1	2.00	1/min	Forward rate constant for the reaction of A, B to C_1
3	k_r1	1.00	mol/(l*min)	Reverse rate constant for the reaction of A, B to C_1
4	Km_2	0.10	mol	Michelis Menten constant for enzymatic conversion of C_1 to C_2
5	kcat_2	3.00	1/min	Michelis Menten constant for enzymatic conversion of C_2 to C_1
6	k_f3	2.00	mol/(l*min)	Forward rate constant for the reaction of C_2 to I, P
7	k_f4	2.00	1/min	Forward rate constant for the reaction of I, Prot to I.Prot
8	k_syn5	0.50	1/min	Synthesis rate constant of A by factor Prot
9	k_d6	0.10	1/min	Degradation rate constant for I.Prot



Error...

Unit: 'mol' not a possible volume unit. Possible units are: ul, uL, ml, mL, dl, dL, l, L, cm3, dm3, m3, km3, us\_tsp, us\_tbsp, us\_oz, us\_cup, us\_pint, us\_quart, us\_gal, inch3, in3, ft3, mi3, imp\_tsp, imp\_tbsp, imp\_oz, imp\_cup, imp\_pint, imp\_quart, imp\_gal

Ok

## Converting Compound Units

Often parameter values will have multiple unit definitions. From the parameter table example above, the parameter "k\_r1" has a unit definition of mol/(l\*min). This has units of count, volume, and time. To convert this, the user needs to ensure they repeat the whole expression changing the appropriate unit in the proper place. For example,

$$mol/(l * min)$$

is an appropriate expression. However,

$$umol/(min * l)$$

will return an error (POSITION MATTERS). As well as not putting the mathematical notion back in its correct place (the division, multiplication, and parenthesis) or having any of the units not be the correct conversion type (i.e. using volume units when you should be using temp).

## Unit Definitions

Below are the unit definitions used in this application. The left-hand side is the term that can be put into the application (parenthesis terms are alternat acceptable names) while the right side is the definition of the unit abbreviation.

### Count

**fmol:** femto-mol  
**pmol:** pico-mol

<b>nmol:</b>	nano-mol
<b>umol:</b>	micro-mol
<b>mmol:</b>	milli-mol
<b>mol:</b>	mol
<b>Flow</b>	
<b>ml_per_sec:</b>	milliliter per second
<b>ml_per_min:</b>	milliliter per min
<b>ml_per_hr:</b>	milliliter per hour
<b>l_per_sec (LPS):</b>	liter per second
<b>l_per_min (LPM):</b>	liter per min
<b>l_per_hr (LPH):</b>	liter per hour

**Temperature:**

<b>C:</b>	Celsius
<b>F:</b>	Fahrenheit
<b>K:</b>	Kelvin
<b>R:</b>	Rankine

**Time:**

<b>nsec:</b>	nano-second
<b>usec:</b>	micro-second
<b>msec:</b>	milli-second
<b>sec:</b>	second
<b>min:</b>	minute
<b>hr:</b>	hour
<b>day:</b>	day
<b>wk:</b>	week
<b>mon:</b>	month
<b>yr:</b>	year
<b>dec:</b>	decade
<b>cen:</b>	century

**Volume:**

<b>ul (uL):</b>	micro-Liter
<b>ml (mL):</b>	milli-Liter
<b>dL (dL):</b>	deci-Liter
<b>l (L):</b>	Liter
<b>us_tsp:</b>	US teaspoon
<b>us_tbsp:</b>	US tablespoon
<b>us_oz:</b>	US ounce
<b>us_cup:</b>	US cup

<b>us_pint:</b>	US pint
<b>us_quart:</b>	US quart
<b>us_gal:</b>	US gallon
<b>cm3:</b>	centimeter cubed
<b>dm3:</b>	decimeter cubed
<b>m3:</b>	meter cubed
<b>km3:</b>	kilometer cubed
<b>in3:</b>	inch cubed
<b>ft3:</b>	feet cubed

Pre-generated units for rate constants:

Law	Rate Constant	Unit Definition	Example
Mass Action	Forward rate constant, kf	Order Dependent; See Below	1/min, L/(mol*min)
Mass Action	Reverse rate constant, kr	Order Dependent; See Below	1/min, L/(mol*min)
Synthesis, Rate	Ks	Count/(Volume*Time)	mol/(L*min)
Synthesis, uses Modifier	Ks	1/Time	1/min
Degradation, Rate	Kd	1/time	1/min
Degradation, by enzyme	Km	count/volume	mol/L
Degradation, by enzyme	kcat	1/time	1/min
Degradation, by enzyme	Vmax	count/(volume*time)	mol/(L*min)
Michaelis-Menten	Km	count/volume	mol/L
Michaelis-Menten	kcat	1/time	1/min
Michaelis-Menten	Vmax	count/(volume*time)	mol/(L*min)

For Order dependent constants (law of mass action) the rate constant has units of

$$\frac{\text{count}^{1-n}}{\text{volume}^{1-n} * \text{time}}$$

where n is the order of the reaction. This gives the following:

For order zero ( $n = 0$ ),

$$\frac{\text{count}}{\text{volume} * \text{time}} [=] \frac{\text{mol}}{\text{L} * \text{min}}$$

For order one ( $n = 1$ ),

$$\frac{1}{\text{time}} [=] \frac{1}{\text{min}}$$

For order two ( $n = 2$ ),

$$\frac{\text{volume}}{\text{count} * \text{time}} [=] \frac{\text{L}}{\text{mol} * \text{min}}$$

For order three ( $n = 3$ ),

$$\frac{\text{volume}^2}{\text{count}^2 * \text{time}} [=] \frac{\text{L}^2}{\text{mol}^2 * \text{min}}$$

And so on for each increasing order of n. The order of the reaction is determined by looking at the rate law and is the sum of the exponents of the species multiplied by the constant. For example, the reaction that has the forward rate constant kf:



Has the rate of,

$$v = kf[A]^1[B]^2$$

this rate constant, kf, is thus of order 3 (1 + 2).

## Creating A Custom Law

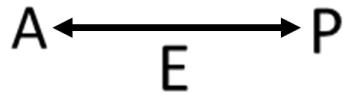
This will cover all the necessary components of creating new laws. BioModME is still a relatively new program and, as such, consists of a small number of premade laws. In this section, we will go over how to create and use a custom law in BioModME. To start, navigate to “Modeler’s Toolbox > Build Custom Law”. You should end up on a page that has the below box:

The figure shows a screenshot of the 'Build Custom Law' interface. At the top, there are fields for 'Law Name' (labeled 1) and 'Description' (labeled 2). Below that is a 'Rate Law' section containing an input field with the equation  $x1*p1*x2^2/(mod*y1)$  (labeled 3). Underneath are sections for 'Reactants' (x1, x2), 'Products' (y1, labeled 4), and 'Modifiers' (mod1). To the right, a 'Reaction Parameters' table lists variables from the rate law: p1, x1, x2, mod, and y1. A note says 'Parameters will be extracted from above rate law here'. At the bottom right is a 'Store' button (labeled 6).

In the figure above, we have the following:

1. Law Name – Enter the name to identify law by
2. Description – Description of law
3. Rate Law – Box to enter string version of the equation (red when not valid, green when valid)
4. Boxes to tell the program which variables in the model are reactants, products, or modifiers. Any variables listed in **Rate Law** are auto assigned as a parameter. Typing that value into one of the respective three boxes will remove it from the parameter tag.
  - a. Reactant – species on left hand side of equation and adopt a negative version of rate law.
  - b. Products – species on the right-hand side of equation that adopt a positive version of the rate law.
  - c. Modifiers – species that are involved in the reaction but do not have a rate law from the reaction.
5. Table of parameters extracted from **Rate Law**. Here there will be customization options.
6. Button to press to store equation.

### Example : Reversible Michaelis-Menten Reaction



Given a reversible enzyme reaction given by:

Where A is the reactant, P is the product, and E is the enzyme. We derive the following rate law:

$$\frac{V_f * \frac{[A]}{Km_A} - V_r * \frac{[P]}{Km_P}}{1 + \frac{[A]}{Km_A} + \frac{[P]}{Km_P}}$$

Where:

- $V_f$  – forward velocity (parameter)
- $V_r$  - reverse velocity (parameter)
- $Km_A$  – Michaelis-Menten Constant of A (parameter)
- $Km_P$  – Michaelis-Menten Constant of P (parameter)
- A – concentration of species A (reactant)
- P – concentration of species P (product)
- E – concentration of species E (modifier)

Note that even though the enzyme is not in the rate law, it should still be included in the equation description.

Below is how the form should look for entering this law:

Law Name	Description
Reversible Michaelis-Menten	Reversible enzyme kinetics law.

Rate Law	
$(V_f * A/Km_A - V_r * P/Km_p) / (1 + A/Km_A + P/Km_p)$	

Reactants	A	Variables	Type
Products	P	1 V_f 2 Km_A 3 V_r 4 Km_p	Parameter Parameter Parameter Parameter
Modifiers	E		

Steps:

1. Name the law: Reversible Michaelis-Menten
2. Add a description!
3. Type a string version of the rate law. We have built a parser that determines if the law is valid. The **Rate Law** box will be red if the equation is not valid and green if valid.
4. Put A in reactants, P in Products, and E in Modifiers.
5. Press the “Store” Button. A success message should appear.

Important things to note here, when creating your own law, all parameter values will be dimensionless. The parameter table provides a dropdown on “Type” if the parameters are a volume. If time, we ask you to use “t” as the variable.

To use the newly created law, travel to “Create Model >Reactions Box > +” to add a new reaction. In the **Reaction Type** input, select “Custom Reaction”. In **Reaction Law**, there should be the newly created law “Reversible Michaelis-Menten”.

Active Compartment  
compartment\_1

**Reaction Type**  
Custom Reaction

**Reaction Law** 1

Reversible Michaelis-Menter

**Build**      Description

Reactant: A	Product: P	Modifier: E
A	P	E

---

Parameter: V_f	Value
V_f	0

---

Parameter: Km_A	Value
Km_A	0

---

Parameter: V_r	Value
V_r	0

---

Parameter: Km_p	Value
Km_p	0

→

In the above:

1. Area to select custom reaction.
2. Options to add reactants, products, and modifiers based on what was entered in the law build.
3. The parameters in the equation law to be added.

Simply fill out the form as you would like any reaction, press the **Add** button and the reaction will be saved using your new law.

## Creating a Custom Equation

This option is for cases when a parameter is not a constant value. Often, parameters can be dependent on concentration levels, other parameters, or time. For these, a custom equation must be built for the interaction. This section will go over how to add these equations and how to navigate through the UI for this section of the application. To begin, navigate to “Modeler’s Toolbox -> Custom Equations”. The following box should be displayed:

The screenshot shows a dialog box for creating a custom equation. At the top, there are two input fields: 'Variable' (1) and 'Expression' (2). Below these are two tables: 'Existing Variables' (3) and 'New Variables' (4). In the center, there is a reaction display: ' = ' followed by a circled number (5). At the bottom right is a button labeled 'Add Equation' (6).

1. Box to enter variable that is dependent on the expression.
2. Mathematical expression for variable
3. Table displaying variables that are already in model.
4. Table displaying variables that are not in model.
5. Reaction display of custom equation.
6. Button to store custom reaction.

## SBML Import

Our application can import SBML files of level 2 and lower. The importer is written to look for certain tags and pull those, and not to particularly read on a level-by-level basis. For this reason, it does have the ability to read level 3 files, but some information could be lost in the process. As it stands, the importer will ignore units, but the converter for the SBML import will be included in future revisions of this application.

## Export Model

The model created in the application can be exported in a variety of ways including to different coding languages (R, MATLAB) and storage file types (rds, SBML). Model information can be exported as txt, csv, pdf, or LateX in a variety of tables. Lastly, the differential equations can be exported in a variety of formats including text, LateX, MathJax, and MathML.

## Coding Languages

Once the model is created, it can be exported to different coding languages for the users' different purposes. This export converts the code into a full script that is ready to be run right off download. However, due to the complexities and syntaxes of different languages, the files might need some editing from the user (ex. User may need to use a different ODE solver or change the variable syntax of specific species).

### MATLAB

To export to MATLAB, navigate to the “Export” tab and move to the “code” box. Name the file and press “MatLab Code”.

The code will be generated in the following code blocks:

#### **Driver Block:**

This block will define the time, load initial conditions and parameters, run the model, and plot all species. It is set to run ode15s as the ODE solver (equivalent of lsoda).

#### **Model Function:**

This is a function that contains all model information. It begins by assigning names to the species and parameters in the model, then defines the differential equations used in the model. Any custom equation will be added and shown before the differential equations of the model. Lastly, it packages the results to a vector to return to the solver.

#### **Parameter Function:**

The function “function p = pars()” will assign parameter values to the model. Each line is commented with the parameter name. The parameter values are packaged to a vector and returned.

#### **Initial Conditions Function:**

The function “function y0 = Initial\_Conditions()” will assign initial conditions to the model species. Species names are commented at the end of each line. Returned as vector.

#### **Notes:**

Variable names in BioModME can have “.” separating terms in a name. This is not a valid MATLAB syntax. By default, the export will convert all “.” to “\_”. For example, “my.new.variable” will be converted to “my\_new\_variable”.

On export, variable numerical values will default to using “base units”. These are the standard units that BioModME uses in the backend to process the model. (mol, L, min). This can cause some values to differ from those displayed in the program if different units have been used.

#### **R**

To export to R, navigate to the “Export” tab and move to the “code” box. Name the file and press “R Code”.

The code will be generated in the following code blocks:

#### **Model Function:**

This is a function that contains all model information. It takes in named vectors of initial conditions (state) and parameters (parameters) and passed them onto the functions within. The body of this function contains the differential equations of the model and returns a list of values for each species in the model.

#### **Driver Block:**

Below the model function, the driver of the code takes place. It begins by creating named vectors for parameters and initial conditions, generates a time vector, then runs a “lsoda” solver for the odes.

#### **Notes**

On export, variable numerical values will default to using “base units”. These are the standard units that BioModME uses in the backend to process the model. (mol, L, min). This can cause some values to differ from those displayed in the program if different units have been used.

## **Storage Languages**

BioModME can export the model information in a few different storage forms. These are not scriptable text but rather containers that store the model information to be used again by this application or others.

## rds

rds is an r-data structure that can be used to store data in R. This creates a list type structure user R notation. This is the main data storage type to use with this application as it will store all model information and relevant model features that are used in the application. This data type can also be read directly into R but, currently, R does not have a method of processing the structure outside of using BioModME.

## SBML

SBML (systems biology markup language) is a common data structure used in the modeling community that uses an xml style markup. Our application generates level 2 SBML type files that contain the following tags:

1. <listOfCompartments>
2. <listOfFunctionDefinitions>
3. <listOfSpecies>
4. <listOfParameters>
5. <listOfReactions>
  - a. <listOfReactants>
  - b. <listOfProducts>
  - c. <listOfModifiers>
  - d. <kineticLaw>
    - i. <math>
    - ii. <listOfParameters>
6. <listOfRules>

## Model Information Tables

Navigate to the “Export” tab. In the bottom box, there are tabs corresponding to tables for species, compartments, parameters, and equations in the model. Clicking each tab of the tabbox will change the table. Above each table are buttons for Copy, CSV, Excel, PDF, and Print that will export the respective table in that method.

- Copy – Copies the table to a clipboard to copy and paste.
- CSV – Downloads a csv file of the table.
- Excel – Downloads an xlsx file of the table.
- PDF – Downloads a pdf file of the table.
- Print – Sends the table to print.

We have created a new system that we are adding to our tables in the application. The following buttons have been added above tables:

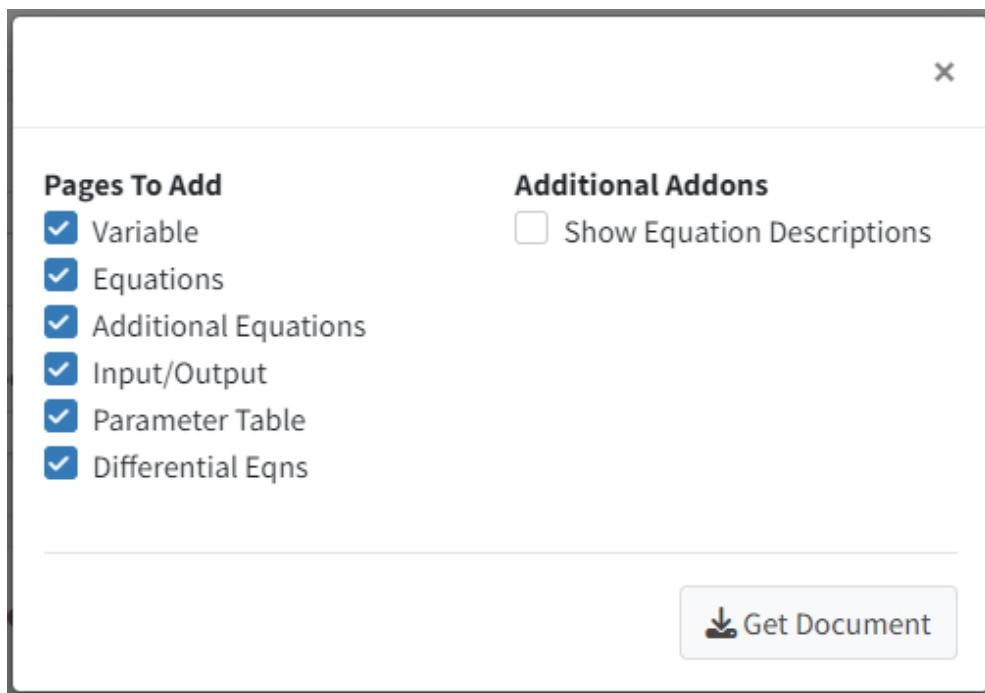


From left to right:

1. Copy – copies the entire table to computer clipboard (can be pasted to program of your choice)
2. Download csv – downloads table to comma separated value file.
3. Download xlsx – downloads table to excel file.
4. Open in New Window – Opens table to new window, separated from main app.

## LaTeX Document

The ability to download the model information in LaTeX format is a part of this application. This option will generate all the TEX needed to display the model in a formatted way. This option will also subset all variable names (i.e k\_f1 -> k<sub>f1</sub>). To use this option, navigate to the “Export” Tab and select “LaTeX Download” button after providing a name. A popup will appear asking the user to select what pages they want displayed in the document. After selecting, simply click “Get Document” and the LaTeX document will download and can be opened in your preferred TEX editor.



## Differential Equations

The differential equations in the model can be downloaded in a few ways. One option is to download them in the above LaTeXdocument. As current, long differential equations will be run off the LaTeXpage though, so the user will need to add the line breaks themselves if they wish to see them in that format.

The different equations can be downloaded in their LaTeXform and MathML form by navigating to the “Create Model” tab and the scrolling down to the “Differential Equations” box. Here your different equations are displayed one-by-one. Simply right-click on the equation you wish to get the underlying code for and navigate to “Show Math As -> MathML Code” to download the MathML version or “Show Math As -> TeX Commands” to get the TeX version of the equation.

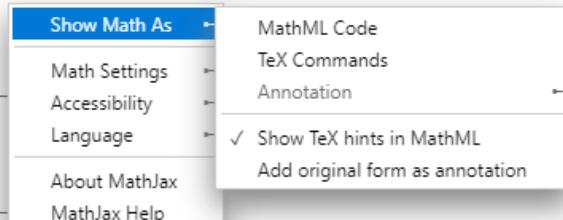
## Differential Equations

$$(1) \ V_{cell} \frac{d[Prot]}{dt} = - (V_{cell} * (k_{f4} * I * Prot))$$

$$(2) \ V_{cell} \frac{d[A]}{dt} = - (V_{cell} * (k_{f1} * A * B -$$

$$(3) \ V_{cell} \frac{d[B]}{dt} = - (V_{cell} * (k_{f1} * A * B -$$

$$(4) \ V_{cell} \frac{d[C_1]}{dt} = + (V_{cell} * (k_{f1} * A * B - k_{r1} * C_1)) - \left( \frac{k_{cat2} * Enz * C_1}{Km_2 + C_1} \right)$$



Much the same, you can display all the equations in their native plain TeX commands by right-clicking on the equation and going to “Math Settings -> Math Renderer -> Plain Source”.

### Differential Equations

```
\begin{equation}(1) \ V_{cell} \frac{d[Prot]}{dt} = - (V_{cell} * (k_{f4} * I * Prot))
```

```
\begin{equation}(2) \ V_{cell} \frac{d[A]}{dt} = - (V_{cell} * (k_{f1} * A * B - k_{r1} * C_1)) + (k_{syn5} * Prot)
```

```
\begin{equation}(3) \ V_{cell} \frac{d[B]}{dt} = - (V_{cell} * (k_{f1} * A * B - k_{r1} * C_1))
```

```
\begin{equation}(4) \ V_{cell} \frac{d[C_1]}{dt} = + (V_{cell} * (k_{f1} * A * B - k_{r1} * C_1)) - \left( \frac{k_{cat2} * Enz * C_1}{Km_2 + C_1} \right)
```

```
\begin{equation}(5) \ V_{cell} \frac{d[C_2]}{dt} = + \left( \frac{k_{cat2} * Enz * C_1}{Km_2 + C_1} \right) - (V_{cell} * (k_{f3} * C_2))
```

```
\begin{equation}(6) \ V_{cell} \frac{d[Enz]}{dt} = 0
```

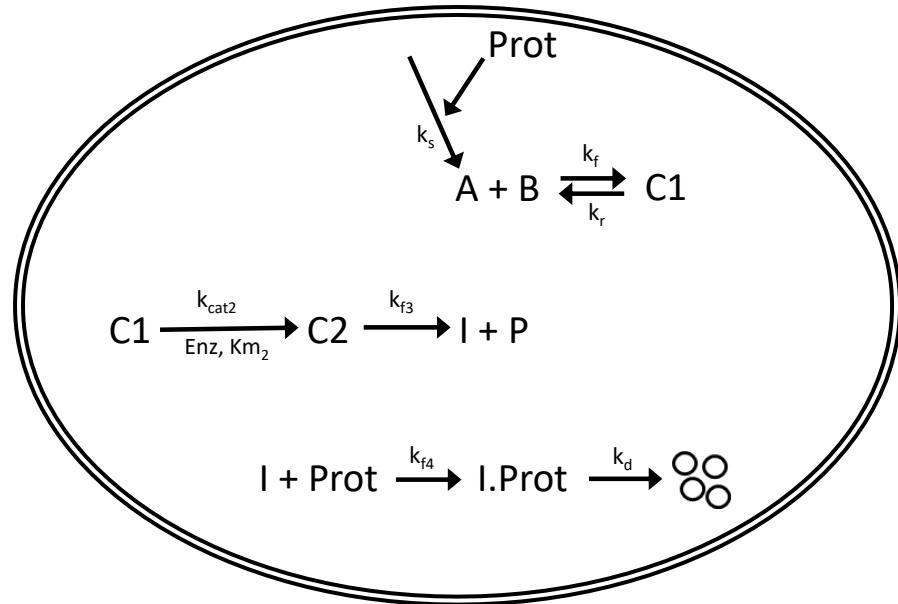
```
\begin{equation}(7) \ V_{cell} \frac{d[P]}{dt} = + (V_{cell} * (k_{f3} * C_2))
```

```
\begin{equation}(8) \ V_{cell} \frac{d[I]}{dt} = + (V_{cell} * (k_{f4} * I * Prot)) - (V_{cell} * (k_{f6} * I * Prot))
```

```
\begin{equation}(9) \ V_{cell} \frac{d[Prot]}{dt} = + (V_{cell} * (k_{f4} * I * Prot)) - (V_{cell} * (k_{f6} * I * Prot))
```

## Basic Tutorial - Introduction

This tutorial covers basics features of BioModME by building a simplified, single region model. In this example, we focus on a set of reactions occurring in a cell. Compound "A" is synthesized by "Prot" at a constant rate and reacts with a limited supply of compound "B" found in the cell to make an intermediate "C1". C1 undergoes an enzymatic conversion to "C2" and this new intermediate undergoes self-cleavage to the desired end product, "P", and an inhibitor, "I", of Prot. This inhibitor feedback binding to Prot stopping the synthesis of A, and thereby the entire reaction scheme.



In the following sections of this tutorial, we will design the above system and look at how this program can be used to visualize and modify the data. The contents include the following:

1. Creating model variables
2. Building a system of equations
3. Entering the inputs and outputs of the system
4. Inputting numerical parameter values
5. Solving and plotting the differential model
6. Exploring the different types of plotting features
7. Exporting the model and its features in meaningful ways

To use the online version of this application visit <https://biomodme.ctsi.mcw.edu/>. Your current model can be downloaded locally using the "Export" tab. The file format is an .RDS file. This file

can be loaded in by opening the right sidebar of the application. A local version of the application can be directly downloaded at <https://github.com/MCWComputationalBiologyLab/BioModME> and opened using the R programming language using stand Shiny app protocol. Refer to documentation for pulling a docker version of the application.

## Constructing Model

This section will cover how to build and construct a model.

### Define Variables

The application loads on the “Create Model” tab. Here, we will build our model, setting up variables, equations, and parameters. Each of these components are their own box on the page. On load, the application will create a starting compartment named “comp\_1”, which can be seen in the “Compartments” box. Click on the “Name” cell of the table to change the compartments name (cell). We will also change the volume name here (V\_cell). In a single region model, the physical volume is not of much importance so we will leave that as a value of 1 to avoid any model scaling effects.

Compartments				
Name	Volume	Value	Unit	Description
cell	V_cell	1.00	l	
				

Next, the model species will be added. To add a species, move to the next box (“Species”). Here we press the button with a plus icon to add species whose information can be edited in the table.

Species	
Instructions	
Press addition button below to add species to compartment.	

We will add nine species to the model and then rename them appropriately: Prot, A, B, C\_1, C\_2, Enz, P, I, I.Prot.

	Name	Value	Unit	Compartment	Description
1	Prot	0.00	mol/L	cell	
2	A	0.00	mol/L	cell	
3	B	0.00	mol/L	cell	
4	C_1	0.00	mol/L	cell	
5	C_2	0.00	mol/L	cell	
6	Enz	0.00	mol/L	cell	
7	P	0.00	mol/L	cell	
8	I	0.00	mol/L	cell	
9	I.Prot	0.00	mol/L	cell	

+
-

The second column of the species is “Value”, which is the initial amount of that species in the model. For this model we have the following initial concentrations:

- Prot = 3
- A = 0
- B = 5
- C\_1 = 0
- C\_2 = 0
- Enz = 0.5
- P = 0
- I = 0
- I.Prot = 0

	Name	Value	Unit	Compartment	Description
1	Prot	3.00	mol/L	cell	
2	A	0.00	mol/L	cell	
3	B	5.00	mol/L	cell	
4	C_1	0.00	mol/L	cell	
5	C_2	0.00	mol/L	cell	
6	Enz	0.50	mol/L	cell	
7	P	0.00	mol/L	cell	
8	I	0.00	mol/L	cell	
9	I.Prot	0.00	mol/L	cell	

+
-

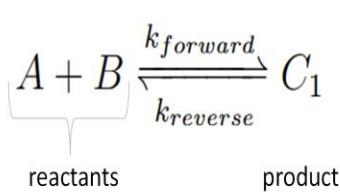
Lastly, if desired, we can add descriptions to our species to remind ourselves (or let other users) know what the species in the model are. Below, we have entered brief descriptions of the beginning species. Editing can be done by clicking the appropriate cell in the species table.

Species					
	Name	Value	Unit	Compartment	Description
1	Prot	3.00	mol/L	cell	Protein synthesizing A
2	A	0.00	mol/L	cell	Compound which binds to B
3	B	5.00	mol/L	cell	Compound which binds to A
4	C_1	0.00	mol/L	cell	Intermediate Substrate
5	C_2	0.00	mol/L	cell	
6	Enz	0.50	mol/L	cell	
7	P	0.00	mol/L	cell	
8	I	0.00	mol/L	cell	
9	I.Prot	0.00	mol/L	cell	

## Create Reactions

Scroll down to the “Reactions” box in the “Create Model” tab. In this section we will build up the sets of reactions that make up our reaction network. Press the “+” button to begin adding equations.

## Equation 1 - Law of Mass Action, Reversible



Chemical reactions like this follow the **law of mass action**. This first equation is a reversible reaction. The equation consists of two reactants, one product, and moves in both directions where forward rate constant is “ $k_f1$ ” and our reverse is “ $k_r1$ ”.

This reaction input can be seen in the below figure. The steps for the following equation are:

1. Select “Chemical Reaction” for equation type.
  2. Select “Mass Action” as the law.
  3. Select “Reversible” as the reaction direction.
  4. Set the equation to have two reactants and one product.
  5. Fill out equation information. Rate constant names are autogenerated for all equation types using the number of the current equation as their number at the end.
  6. This is the equation builder that shows what your built equation looks like.
  7. Press the “Add Equation” button.

**Active Compartment**

cell

---

**Reaction Type**

All (1)

**Reaction Law**

Mass Action (2)

**Reversability?**

Reversible (3)

---

**Build**      Description

Number of Reactants	Number of Products
2	1

(4)

Reactant 1	Product 1	Forward Rate Constant	Reverse Rate Constant
1 A	1 C_1	k_f1	k_r1

(5)

Value
0

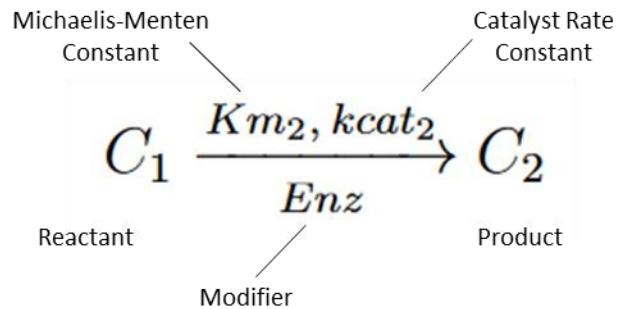
(6)

$A + B \xrightleftharpoons{k_{f1}, k_{r1}} C_1$

---

Close on Add (7) Add Equation

## Equation 2 - Michaelis-Menten Kinetics



Next, we enter the second equation, conversion of C1 to C2 by the enzyme “Enz”. These reactions follow Michaelis-Menten kinetics. Figure 7 shows how this equation looks entered the program.

The steps are as follows:

1. Change the Reaction Type dropdown to “Enzyme Based Reaction”.
2. Select “Michaelis Menten” as the Law.
3. Enter the equation information into the builder.
4. Press the “Add Equation” Button.

**Active Compartment**

**Reaction Type**  

1

**Reaction Law**  

2

Use Vmax

**Substrate**

**Product**

**Enzyme**

**Km**  
Value

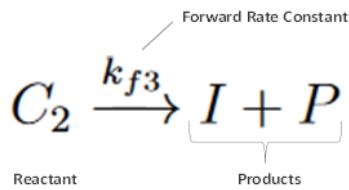
**kcat**  
Value

$$C_1 \xrightarrow[\text{Enz}]{Km_2, kcat_2} C_2$$

Close on Add

4

### Equation 3 - Law of Mass Action, Unidirectional



The next reaction is a conversion of C2 to product P and inhibitor I. This is a one directional reaction with one reactant and two products following the law of mass action.

1. Change the Equation Type dropdown to “Chemical Reaction”.
2. Select “Mass Action” as the Law.
3. Change the reversibility option to “Irreversible”.
4. Set the Number of Reactants to “1” and the Number of Products to “2”.
5. Enter the equation information into the builder.
6. Press “Add Equation” button.

**Active Compartment**

**Reaction Type**

1

**Reaction Law**

2

**Reversability?**

3

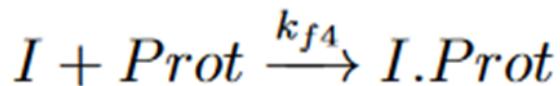
<b>Number of Reactants</b>	<b>Number of Products</b>
<input style="border: 1px solid #ccc; border-radius: 10px; padding: 2px; width: 30px; height: 20px; margin-bottom: 5px;" type="text" value="1"/> <span style="border: 1px solid black; border-radius: 50%; padding: 2px 10px; font-size: 14px; margin-right: 10px;">4</span>	<input style="border: 1px solid #ccc; border-radius: 10px; padding: 2px; width: 30px; height: 20px; margin-bottom: 5px;" type="text" value="2"/> <span style="border: 1px solid black; border-radius: 50%; padding: 2px 10px; font-size: 14px; margin-right: 10px;">4</span>
<b>Reactant 1</b>	<b>Product 1</b>
<input style="border: 1px solid #ccc; border-radius: 10px; padding: 2px; width: 30px; height: 20px; margin-bottom: 5px;" type="text" value="1"/> <span style="border: 1px solid black; border-radius: 50%; padding: 2px 10px; font-size: 14px; margin-right: 10px;">5</span>	<input style="border: 1px solid #ccc; border-radius: 10px; padding: 2px; width: 30px; height: 20px; margin-bottom: 5px;" type="text" value="I"/> <span style="border: 1px solid black; border-radius: 50%; padding: 2px 10px; font-size: 14px; margin-right: 10px;">5</span>
<b>Product 2</b>	<b>Forward Rate Constant</b>
<input style="border: 1px solid #ccc; border-radius: 10px; padding: 2px; width: 30px; height: 20px; margin-bottom: 5px;" type="text" value="1"/> <span style="border: 1px solid black; border-radius: 50%; padding: 2px 10px; font-size: 14px; margin-right: 10px;">5</span>	<b>Value</b>
<input style="border: 1px solid #ccc; border-radius: 10px; padding: 2px; width: 100px; height: 20px; margin-bottom: 5px;" type="text" value="k_f3"/> <span style="border: 1px solid black; border-radius: 50%; padding: 2px 10px; font-size: 14px; margin-right: 10px;">5</span>	<input style="border: 1px solid #ccc; border-radius: 10px; padding: 2px; width: 30px; height: 20px; margin-bottom: 5px;" type="text" value="0"/> <span style="border: 1px solid black; border-radius: 50%; padding: 2px 10px; font-size: 14px; margin-right: 10px;">5</span>

$$C_2 \xrightarrow{k_{f3}} I + P$$

Close on Add

6
Add Equation

#### Equation 4 - Law of Mass Action, Unidirectional

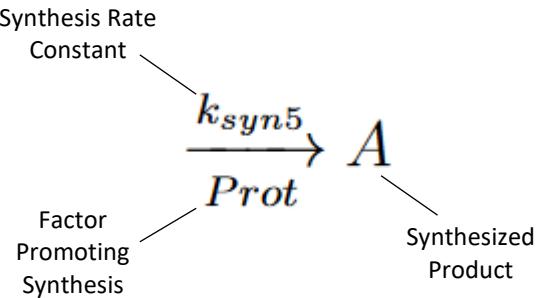


Our next reaction is the inhibitor of Prot. This is another chemical reaction following the law of mass as in Equations 1 and 3 with two Reactants I and Prot converted to I.Prot. The user interface input is below in Figure 9. The steps should be familiar to your by now, but they are as follows:

1. Change the Reaction Type dropdown to “Chemical Reaction”.
2. Select “Mass Action” as the Law.
3. Change the reversibility option to “Irreversible”.
4. Set the Number of Reactants to “2” and the Number of Products to “1”.
5. Enter the equation information into the builder.
6. Press “Add Equation” button.

The screenshot shows a software interface for building chemical reactions. On the left, under "Active Compartment", a dropdown menu is set to "cell". In the center, there are several input fields and dropdown menus. A vertical column on the left contains numbered circles (1 through 6) corresponding to the steps listed above. Step 1 is "Reaction Type: Chemical Reaction". Step 2 is "Reaction Law: Mass Action". Step 3 is "Reversability?: Irreversible". Step 4 is "Number of Reactants: 2" and "Number of Products: 1". Step 5 is the reaction builder area where "Reactant 1" is "I" and "Product 1" is "I.Prot", with a "Forward Rate Constant" of "k\_f4" and a value of "0". Step 6 is the "Add Equation" button at the bottom right. At the bottom left is a checkbox for "Close on Add". The final equation shown is  $I + Prot \xrightarrow{k_{f4}} I.Prot$ .

#### Equation 5 - Synthesis by Factor



Our next reaction is the synthesis of A by Prot, where Prot is a factor promoting synthesis (as opposed to being directly converted).

1. Set the Reaction Type to “Chemical Reaction”.
2. Change the Reaction Law dropdown to “Synthesis”.
3. Click the “Factor Driving Synthesis?” checkbox.
4. Fill out equation builder with the species to synthesize and its corresponding factor.
5. Press the “Add Equation” button.

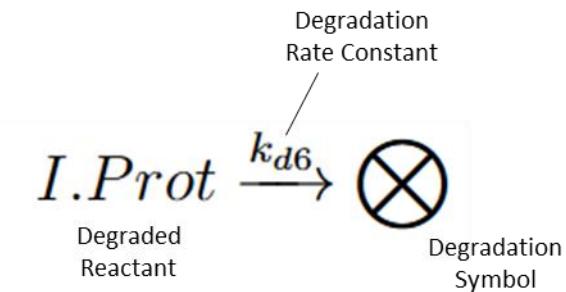
**Active Compartment**  
cell

<b>Reaction Type</b> <input type="radio"/> All <b>1</b> <b>Reaction Law</b> <input type="radio"/> Synthesis <b>2</b> <input checked="" type="checkbox"/> Factor Driving Synthesis? <b>3</b>	<b>Build</b>	<b>Description</b> <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top;"> <b>Species to synthesize</b>  <input type="radio"/> A         </td> <td style="width: 50%; vertical-align: top;"> <b>Factor causing synthesis</b>  <input type="radio"/> Prot         </td> </tr> <tr> <td style="vertical-align: top;"> <b>Rate Constant</b>  <input type="text" value="k_synth5"/> </td> <td style="vertical-align: top;"> <b>Value</b>  <input type="text" value="1"/> </td> </tr> </table> <b>4</b>	<b>Species to synthesize</b> <input type="radio"/> A	<b>Factor causing synthesis</b> <input type="radio"/> Prot	<b>Rate Constant</b> <input type="text" value="k_synth5"/>	<b>Value</b> <input type="text" value="1"/>
<b>Species to synthesize</b> <input type="radio"/> A	<b>Factor causing synthesis</b> <input type="radio"/> Prot					
<b>Rate Constant</b> <input type="text" value="k_synth5"/>	<b>Value</b> <input type="text" value="1"/>					

$$\xrightarrow[k_{syn5}]{Prot} A$$

Close on Add
**5**
Add Equation

## Equation 6 - Degradation by Rate



Our final reaction is the degradation of I.Prot by a rate. This is useful for when we know the rate at which a protein is degraded in the cell but do not really know what is causing the degradation. These are often concentration dependent.

1. Set the Reaction Type to “Chemical Reaction”.
2. Change the Reaction Law dropdown to “Degradation (Rate)”.
3. Fill out equation builder with the species to degrade and its rate constant.
4. Make sure the “Concentration Dependent” box is checked.
5. Press the “Add Equation” button.

Active Compartment

cell

Reaction Type

All 1

Reaction Law

Degradation (Rate) 2

Degrade Into Products?

**Build** Description

Species to degrade

I.Prot 3

Rate Constant

k\_d6 4

Concentration Dependent

$I.Prot \xrightarrow{k_{d6}} \text{⊗}$

Close on Add 5 Add Equation

## Compartment Input/Output

The next box is the Input/Output box. This box is used to add flow/transfers between regions. Our model does not have any flows to account for so we will not use this box for anything.



## Parameters

Navigate to the next box: **Parameters**.

Below is the generated parameter table that has been created so far by the above entered information. All descriptions are pre-generated based on where the parameter originated. Like other tables, all values are editable.

### Parameters

#### View By:

All

	Name	Value	Unit	Description
1	V_1	1	L	Volume of compartment_1
2	k_f1	0	L/(mol*min)	Forward rate constant for the reaction of A, B to C_1
3	k_r1	0	1/min	Reverse rate constant for the reaction of A, B to C_1
4	Km_2	0	mol/L	Michelis Menten constant for enzymatic conversion of C_1 to C_2
5	kcat_2	0	1/min	Michelis Menten constant for enzymatic conversion of C_2 to C_2
6	k_f3	0	1/min	Forward rate constant for the reaction of C_2 to I, P
7	k_f4	0	L/(mol*min)	Forward rate constant for the reaction of I, Prot to I.Prot
8	k_syn5	0	1/min	Synthesis rate constant of A by factor Prot
9	k_d6	0	1/min	Degradation rate constant for I.Prot

The parameter values for the model are as follows:

- $k_f1 = 2$
- $k_r1 = 1$
- $Km_2 = 0.1$
- $kcat_2 = 3$
- $k_f3 = 2$
- $k_f4 = 2$
- $k_s5 = 0.5$
- $k_d6 = 0.1$

The below figure shows the parameter table with the values entered.

## Parameters

View By:

All

	Name	Value	Unit	Description
1	V_1	1.00	L	Volume of compartment_1
2	k_f1	2.00	L/(mol*min)	Forward rate constant for the reaction of A, B to C_1
3	k_r1	1.00	1/min	Reverse rate constant for the reaction of A, B to C_1
4	Km_2	0.10	mol/L	Michelias Menten constant for enzymatic conversion of C_1 to C_2
5	kcat_2	3.00	1/min	Michelias Menten constant for enzymatic conversion of C_2 to C_2
6	k_f3	2.00	1/min	Forward rate constant for the reaction of C_2 to I, P
7	k_f4	2.00	L/(mol*min)	Forward rate constant for the reaction of I, Prot to I.Prot
8	k_syn5	0.50	1/min	Synthesis rate constant of A by factor Prot
9	k_d6	0.10	1/min	Degradation rate constant for I.Prot

## Differential Equations

Navigate to the final box “Differential Equations”.

This tab is used to generate and see the mathematical equations that govern our model. This tab will begin with all species being equal to “NA” until the generate button is pressed or model is executed.

Differential equations are solved whenever an equation is added. The “Generate” button is used to refresh the viewing of the differential equations if necessary.

Differential Equations

---

(1)  $V_1 \frac{d[Prot]}{dt} = -(V_1 * (k_{f4} * I * Prot))$

(2)  $V_1 \frac{d[A]}{dt} = -(V_1 * (k_{f1} * A * B - k_{r1} * C_1)) + (V_1 * (k_{syn5} * Prot))$

(3)  $V_1 \frac{d[B]}{dt} = -(V_1 * (k_{f1} * A * B - k_{r1} * C_1))$

(4)  $V_1 \frac{d[C_1]}{dt} = +(V_1 * (k_{f1} * A * B - k_{r1} * C_1)) - \left( V_1 * \left( \frac{kcat_2 * Enz * C_1}{Km_2 + C_1} \right) \right)$

(5)  $V_1 \frac{d[C_2]}{dt} = + \left( V_1 * \left( \frac{kcat_2 * Enz * C_1}{Km_2 + C_1} \right) \right) - (V_1 * (k_{f3} * C_2))$

(6)  $V_1 \frac{d[Enz]}{dt} = 0$

(7)  $V_1 \frac{d[P]}{dt} = +(V_1 * (k_{f3} * C_2))$

(8)  $V_1 \frac{d[I]}{dt} = +(V_1 * (k_{f3} * C_2)) - (V_1 * (k_{f4} * I * Prot))$

(9)  $V_1 \frac{d[I.Prot]}{dt} = +(V_1 * (k_{f4} * I * Prot)) - (V_1 * (k_{d6} * I * Prot))$

---

Generate

## Solve Model

Navigate to the “Execute Model” tab on the lefthand tabbar. This tab solves our model and provides us with the options to customize the solver. For this model, we will just solve using the standard options:

### Note

The right sidebar of the “System of Differential Equations” box has an option for viewing the differential equations including turning off the math rendering and showing each derivation line as a newline for each differential equation.

1. Enter the following times: Starting = 0, End = 15, Step = 0.1, Unit = min. This generates the model to be solved and integrated at time points 0, 0.1, 0.2.....14.8, 14.9, 15.
2. Press the “Run Solver” button.
3. An output table should be generated with the concentration of each species along each time step of the solved model.

You can download this model using the buttons above the table. In order the buttons are: copy to clipboard, download to csv, download to xlsx, and open in new window.

Options
Post Processing

Starting Time	End Time	Time Step	1	Unit
0	15	0.1	min	<input type="button" value="▼"/>

Advanced Options
  Viewing Options

Run Solver

Show 100 entries

time (min)	Prot	A	B	C_1	C_2	Enz	P	I	I.Prot
0	3	0	5	0	0	0.5	0	0	0
0.1	3	0.096	4.946	0.036	0.017	0.5	0.001	0.001	0
0.2	2.998	0.136	4.836	0.089	0.066	0.5	0.009	0.007	0.002
0.3	2.991	0.157	4.707	0.139	0.126	0.5	0.028	0.019	0.009
0.4	2.975	0.17	4.571	0.182	0.187	0.5	0.059	0.034	0.025
0.5	2.949	0.18	4.433	0.22	0.244	0.5	0.103	0.052	0.05

3

2

Run Solver

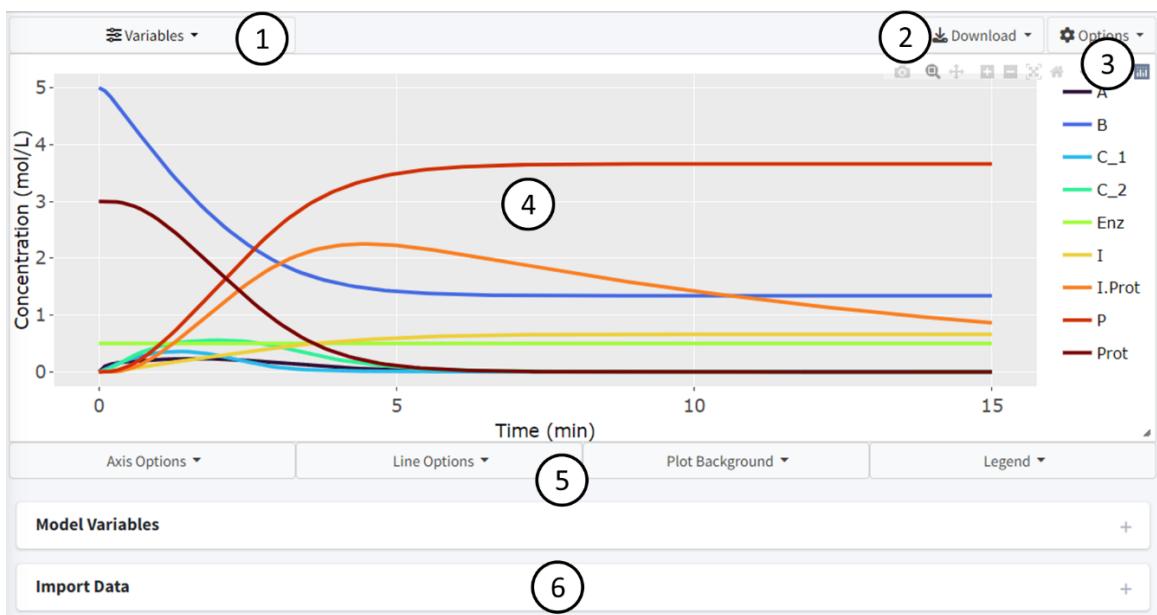
## Visualize Model

This section will cover how to visualize our created model.

Navigate to the “Visualization” tab on the lefthand tabbar.

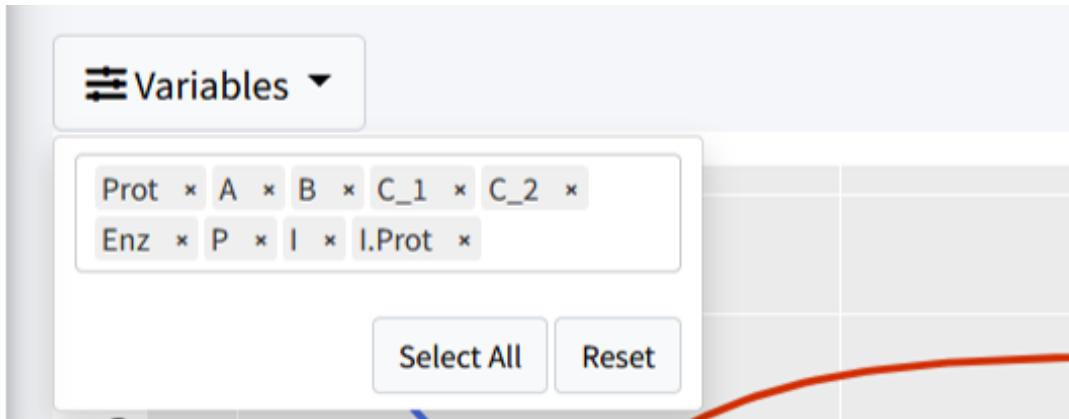
The plotting tab has four modes, we will cover two of them. In this section we will cover “Normal Plot” which is the default when you enter the tab. The tab auto loads the plot when the model is solved, plotting all species of the model. These are the main features of the plot:

1. Variables - Change what species are plotting in the model. All species in the model are plotted by default.
2. Download - Dropdown containing options to download plot.
3. Options - Dropdown containing the following:
  - Plot mode - Changes the plotting types and features. Here we are plotting “Normal Plot”.
  - Plot Renderer – type of plot used. “plotly” is an interactive plot while “ggplot2” is the standard R plotting. The interactive is the default as it allows you to interact with the data using features such as zoom, pan, and hovering over data to view its value.
4. Plot - Plot area with adjustable sizing on bottom right corner.
5. Plot Options – Four dropdowns that allow the addition of labels, change of axis sizing, and control of line color and size.
6. Boxes containing menus to change model variables and import datasets to plot.



By default, the page should look like above when this tutorial model is solved. All variables are displayed using an evenly spaced rainbow pattern coloring.

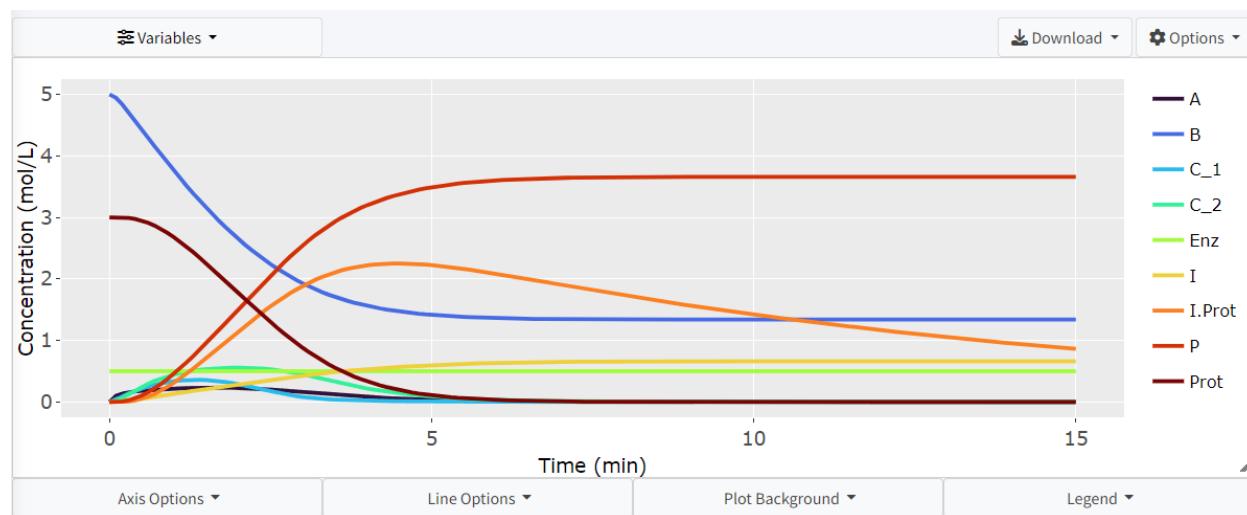
Press the “Variables” dropdown to change the species plotted in the graph. Press the “x” on the species to remove them. The “Select All” button reselects all species while the “Reset” button removes all species.



Let's remove the following species to clean the plot up:

- C\_1
- C\_2
- Enz

We now have a less cluttered plot:



The interactive plot uses Plotly (<https://plotly.com/r/plotly-fundamentals/>) as a render and includes many of its base features including:

1. By clicking on a variable in the legend, they can be removed from the plot.

2. Double clicking a legend variable will remove all species but that one from the plot.
3. The top right bar includes options to pan, hover, and zoom the plot.
4. Press, hold, and drag the mouse on the plot to select specific features.
5. By hovering over lines on the plot, a popup will show with the values at that point.

There are many features included to let the user make the plot their own. These are all located in the dropdowns below the plot. Here, almost any feature of the plot is customizable. Feel free to play around and make the plot your own.

## Export

BioModME allows the built model to be exported into 4 main categories:

1. Prebuild, ready to run code, for either MATLAB or R.
2. An “.rds” file (type of R data file) that can be reloaded into the program. This is the way to save and load a model into the program currently.
3. A prebuild LaTeXdocument all main model information.
4. Downloading tables for parameters, equations, and initial conditions in csv, txt, or pdf form.

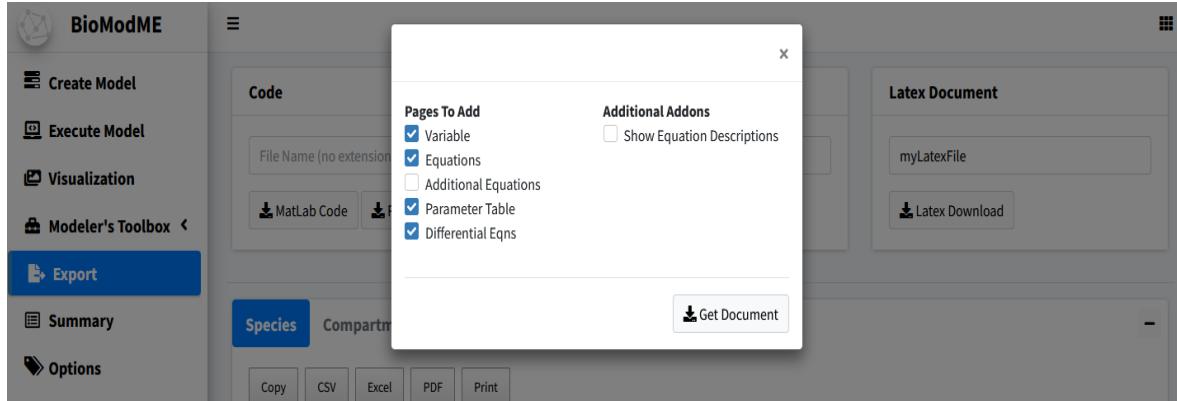
The screenshot shows the BioModME software interface with the 'Export' menu selected. The interface is organized into four main sections, each with a circled number:

- Code (1):** Contains fields for 'File Name (no extension)' and download buttons for 'MatLab Code' and 'R Code'.
- Model File (2):** Contains a field for 'File Name (no extension)' and a 'Save Model' button.
- Latex Document (3):** Contains a field for 'File Name (no extension)' and a 'Latex Download' button.
- Species Data (4):** A table showing species parameters. The columns are Name, Value, Unit, Compartment, and Description. The rows are:

Name	Value	Unit	Compartment	Description
Prot	3	mol	cell	
A	0	mol	cell	
B	5	mol	cell	
C_1	0	mol	cell	

Feel free to try the code export if you have an interest in solving your model in those programming languages. Make sure to give any downloaded file a name.

LaTeXWhen downloading the LaTeXdocument a popup will appear, showing further customization options for that document.



In this example, we will remove the “Additional equations” page as we did not create any non-standard equations. The generated document is created as a text file that can be copied to our LaTeX editor of choice and run. I recommend using overleaf ([www.overleaf.com](http://www.overleaf.com)) but any TeX editor will work. Below is a figure of the overleaf output:

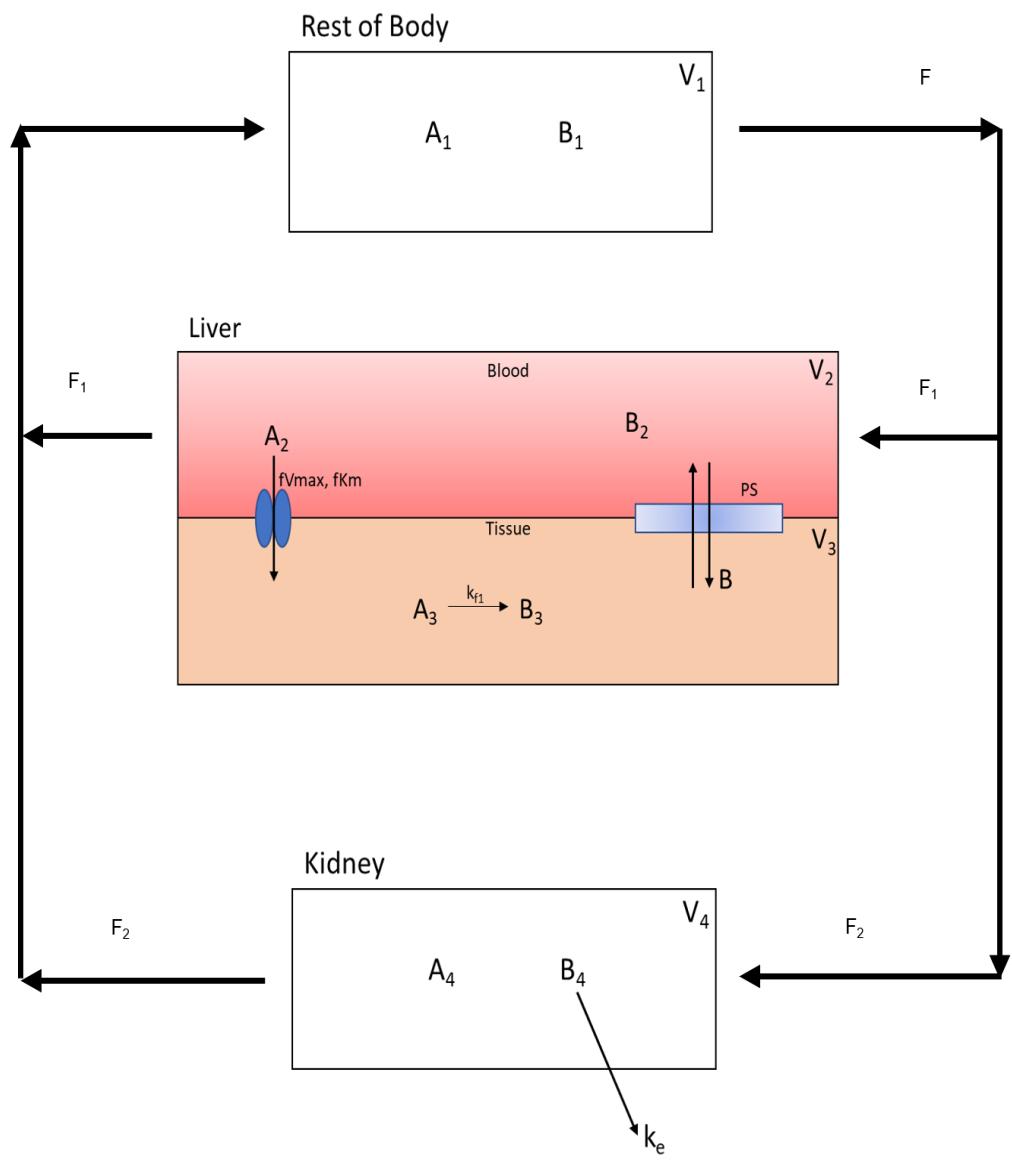
```

\begin{aligned}
V_{cell} \frac{d(\text{Prot})}{dt} &= -(V_{cell} * (k_{f4} * I * \text{Prot})) \\
V_{cell} \frac{d(A)}{dt} &= -(V_{cell} * (k_{f1} * A * B - k_{r1} * C_1)) + (V_{cell} * (k_{syn5} * \text{Prot})) \\
V_{cell} \frac{d(B)}{dt} &= -(V_{cell} * (k_{f1} * A * B - k_{r1} * C_1)) \\
V_{cell} \frac{d(C_1)}{dt} &= +(V_{cell} * (k_{f1} * A * B - k_{r1} * C_1)) - (V_{cell} * (\frac{kcal_2 * \text{Enz} * C_1}{Km_2 + C_1})) \\
V_{cell} \frac{d(C_2)}{dt} &= +(V_{cell} * (\frac{kcal_2 * \text{Enz} * C_1}{Km_2 + C_1})) - (V_{cell} * (k_{f3} * C_2)) \\
V_{cell} \frac{d(\text{Enz})}{dt} &= 0 \\
V_{cell} \frac{d(P)}{dt} &= +(V_{cell} * (k_{f3} * C_2)) \\
V_{cell} \frac{d(I)}{dt} &= +(V_{cell} * (k_{f3} * C_2)) - (V_{cell} * (k_{f4} * I * \text{Prot})) \\
V_{cell} \frac{d(I.\text{Prot})}{dt} &= +(V_{cell} * (k_{f4} * I * \text{Prot})) - (V_{cell} * (k_{d6} * I.\text{Prot}))
\end{aligned}

```

## Multi-Compartmental Model Tutorial

A bolus of a neutral drug "A" was administrated intravenously (i.v.) at time  $t = 0$ . Within the liver, "A" can diffuse (unidirectional) from liver blood region tissue liver tissue region via facilitated diffusion with  $V_{max}$  (nmoles/min) and  $K_m$  (Molar). Within the liver cells, "A" is converted to "B" via a chemical reaction that follows the law of mass action with a first order rate constant  $k_1$  (1/min). The product "B" can also diffuse (simple diffusion) between liver tissue region and blood region with a permeability-surface area product  $PS$  (ml/min). Within the kidneys, "B" is excreted from the blood region via a linear process with a rate constant  $k_e$  (1/min). Here, the variable "V" is the compartmental volume of the respective compartment, numbered to differentiate between them. The variable,  $F$ , is our rate of flow leaving or entering a compartment.



The following are the parameter and concentration tables for the model:

Parameter	Value	Unit	Variable	Initial Value
F	5	L/min	A <sub>1</sub>	10 uM
F_1	3	L/min	B <sub>1</sub>	0
F_2	2	L/min	A <sub>2</sub>	0
fVmax	100	nmol/min	B <sub>2</sub>	0
fKm	10000	nmol/L	A <sub>3</sub>	0
PS	15	ml/min	B <sub>3</sub>	0
k_f1	6	1/min	A <sub>4</sub>	0
k_e	4	1/min	B <sub>4</sub>	0
V_1	3000	ml		
V_2	1250	ml		
V_3	1000	ml		
V_4	750	ml		

In the following sections of this tutorial, we will design the above multi compartment method.

The contents focus on:

1. Adding multiple compartments.
2. Adding variables between compartments.
3. Properly assigning flows between compartments
4. Using transporters to pass species between compartments.

To use the online version of this application visit <https://biomodme.ctsi.mcw.edu/>. Your current model can be downloaded in the “Export” tab as an .RDS file and can be loaded in by opening the right sidebar of the application. The application can also be directly downloaded at <https://github.com/MCWComputationalBiologyLab/BioModME> and opened using the R programming language using standard Shiny app protocol.

## Constructing Model

This section will cover how to build and construct a multi compartment model with a focus on input and outputs from different compartments.

### Create Compartments

We will start by setting up our four compartments in this model: Rest\_Of\_Body, Liver\_Blood, Liver\_Tissue, Kidney.

Steps:

1. Go to the **Compartments** box in the **Create Model** tab.
2. Press the “+” button three times to add three compartments.
3. Change the Names of the compartments in the table to **Rest\_Of\_Body**, **Liver\_Blood**, **Liver\_Tissue**, and **Kidney**.
4. Volume names are autogenerated with numerical numbering.
5. Change the volume value and units to match parameter table (these values can also be changed in the parameter table for the model).

The result should be the following table:

Name	Volume	Value	Unit	Description	-
Rest_Of_Body	V_1	3000	mL		
Liver_Blood	V_2	1250	mL		
Liver_Tissue	V_3	1000	mL		
Kidney	V_4	750	mL		

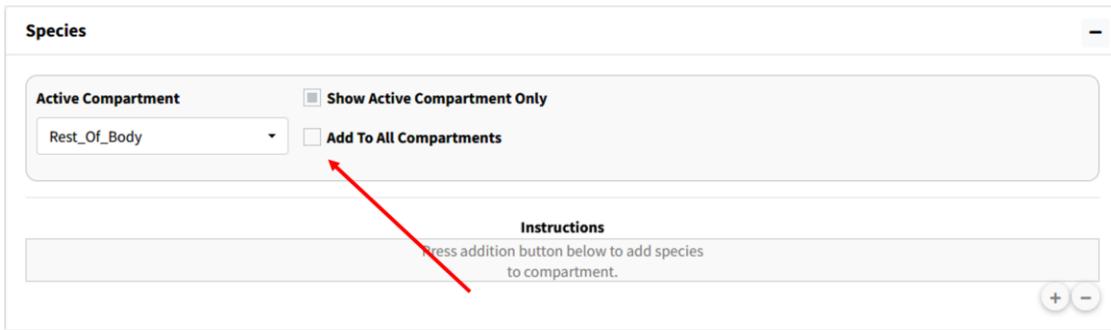
+ -

## Define Variables

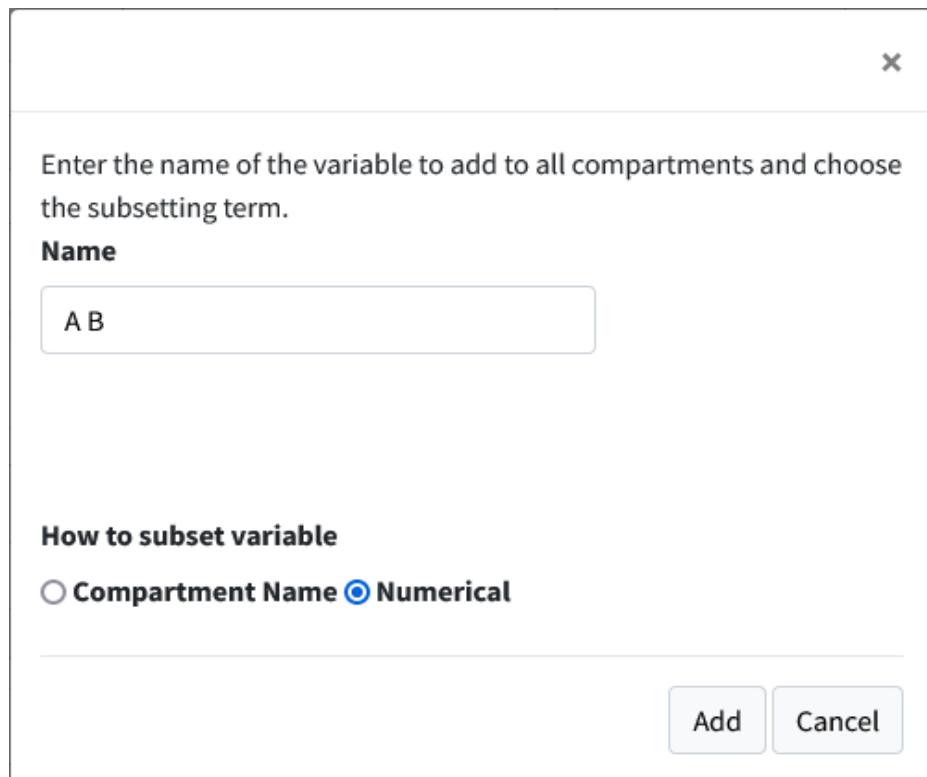
There are two variables (**A** & **B**) to add to all four compartments. We will make use of the “Add To All Compartments” option to quickly add the same species to all compartments.

Steps:

1. Select the “Add To All Compartments” checkbox.



2. Press the “+” button to generate a popup.
3. In Popup type “A B” separated by a space. Check “Numerical” instead of “Compartment Name” and press the add button. This will add the variables **A\_1**, **A\_2**, **A\_3**, **A\_4**, **B\_1**, **B\_2**, **B\_3**, and **B\_4** to the model.



4. Unselect the “Show Active Compartment Only Checkmark” to see all the species in the model.

Res

**Species**

---

**Active Compartment**  Show Active Compartment Only  
 Rest\_Of\_Body   Add To All Compartments

	Name	Value	Unit	Compartment	Description
1	A_1	0.00	mol/L	Rest_Of_Body	
2	A_2	0.00	mol/L	Liver_Blood	
3	A_3	0.00	mol/L	Liver_Tissue	
4	A_4	0.00	mol/L	Kidney	
5	B_1	0.00	mol/L	Rest_Of_Body	
6	B_2	0.00	mol/L	Liver_Blood	
7	B_3	0.00	mol/L	Liver_Tissue	
8	B_4	0.00	mol/L	Kidney	

5. Change the concentration of **A\_1** to match the initial condition table.

**Species**

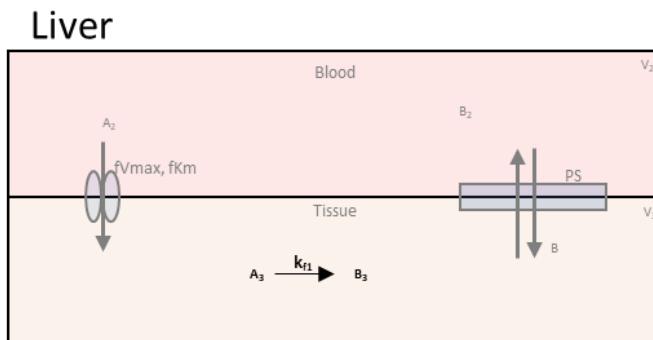
---

**Active Compartment**  Show Active Compartment Only  
 Rest\_Of\_Body   Add To All Compartments

	Name	Value	Unit	Compartment	Description
1	A_1	10.00	umol/L	Rest_Of_Body	
2	A_2	0.00	mol/L	Liver_Blood	
3	A_3	0.00	mol/L	Liver_Tissue	
4	A_4	0.00	mol/L	Kidney	
5	B_1	0.00	mol/L	Rest_Of_Body	
6	B_2	0.00	mol/L	Liver_Blood	
7	B_3	0.00	mol/L	Liver_Tissue	
8	B_4	0.00	mol/L	Kidney	

## Add Equations

This tutorial model has only one reaction: the conversion of A to B (**A<sub>3</sub>** & **B<sub>3</sub>**) in the liver tissue.



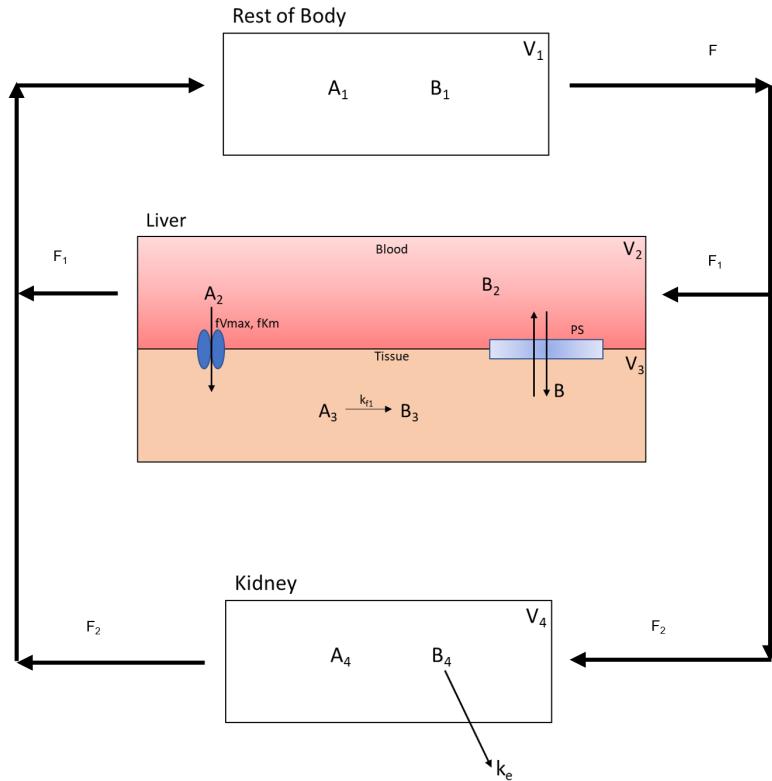
Steps:

1. Scroll down to the **Reactions** box in the **Create Model** tab. Press the addition button below the equation table.
2. Change the **Active Compartment** to “Liver\_Tissue”.
3. Make sure **Mass Action** is selected as the **Law** for this reaction. Change the **Reaction Direction** to “Forward” for this is a one way reaction.
4. This reaction has one reactant and one product. Make sure **Reactant 1** is **A\_3** and **Product 1** is **B\_3**. Make the **Forward Rate Constant** to **k\_f1**, if not already. You can set the parameter value here or later on the parameter table. For this tutorial most values will be set on the parameter table.
5. Press the **Add Equation** button.

The screenshot shows the 'Add Equation' dialog box. Step 2 highlights the 'Active Compartment' dropdown set to 'Liver\_Tissue'. Step 3 highlights the 'Reversability?' dropdown set to 'Irreversible'. Step 4 highlights the 'Reactant 1' and 'Product 1' dropdowns both set to 'A\_3' and 'B\_3' respectively, and the 'Forward Rate Constant' field set to 'k\_f1'. Step 5 highlights the 'Add Equation' button at the bottom right.

## Add Compartment Inputs/Outputs

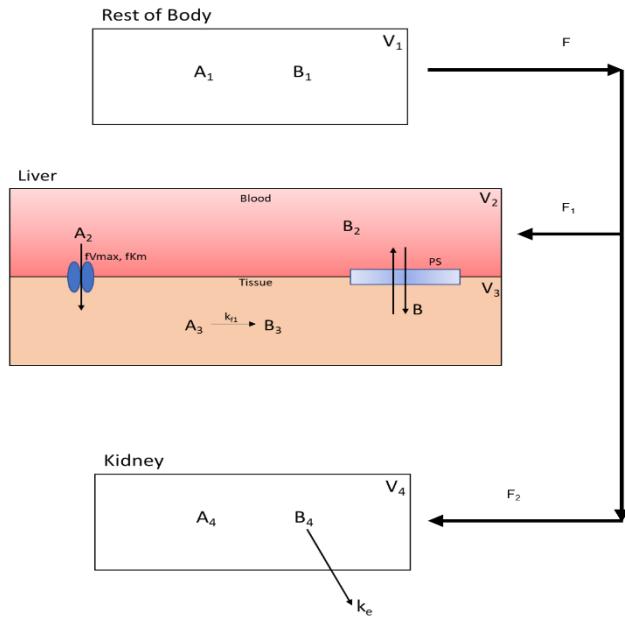
In this section of the tutorial, we will cover how to add flows, inputs, and outputs of compartments to the model. Begin by scrolling to the **Input/Output** box on the **Create Model** tab. Press the addition button to begin adding the below flows between species information. Again, here is the overall flow model:



In each subsequent section we break down the flows into individual components.

## 1. Flow Out of Body

The flow out of the **Rest\_Of\_Body** is split into two flow rates as it moves species **A** and **B** to the **Liver\_Blood** and **Kidney**. Below is the part of the flow diagram we will be entering:



1. In Options select **Flow Between Compartments**. This is used to account for flows leaving one compartment and entering one or more compartments.
2. Select the **Split Flow** checkbox. This option is selected when one outflow is being split into multiple flows. Keep **Number of Splits** to **2** as we are splitting the flow to two compartments.
3. Set the information for the compartment the flow is leaving from. It has the following specifications:
  - **Flow\_Out\_of:** Rest\_Of\_Body
  - **Species Out:**  $A_1$
  - **Flow Variable:**  $F$
  - **Flow Value:** 5

Current specifications need to have species in flow entered one at a time. Meaning a new flow will be added for species **B**. With all parameters, values can be changed later in the parameter section.

4. Set the information from the first split flow into **Liver\_Blood**. Enter the following:
  - **Flow\_Out\_of:** Liver\_Blood
  - **Species Out:**  $A_2$

- **Flow Variable:** F\_1
- **Flow Value:** 3

5. Set the information from the second split flow into **Kidney**. Enter the following:

- **Flow\_Out\_of:** Kidney
- **Species Out:** A\_4
- **Flow Variable:** F\_2
- **Flow Value:** 2

6. Press the **Add** button to add this flow. You can select the **Keep Active** checkbox on the bottom of the page to prevent this page from closing on pressing the add button.

Options		Flow Out Of		Species Out		Flow Variable	Flow Value (l/min)
Flow Between Compartments		Rest_Of_Body		A_1		(3) F	5
<input checked="" type="checkbox"/> Split Flow		Liver_Blood		A_2		(4) F_1	3
Number of Splits		Kidney		A_4		(5) F_2	2
<input type="checkbox"/> Keep Active on Add							
<span>(6) Add</span>							

7. Repeat the previous steps for species **B**. This should involve only changing the **Species** dropdowns.

<b>Options</b> Flow Between Compartments	<b>Flow Out Of</b> Rest_Of_Body	<b>Species Out</b> B_1	<b>Flow Variable</b> F	<b>Flow Value (l/min)</b> 5
<b>Split Flow</b> <b>Number of Splits</b> 2				
	<b>Flow Into</b> Liver_Blood	<b>Species In</b> B_2	<b>Flow Variable</b> F_1	<b>Flow Value (l/min)</b> 3
	<b>Flow Into</b> Kidney	<b>Species In 2</b> B_4	<b>Flow Variable 2</b> F_2	<b>Flow Value (l/min)</b> 2

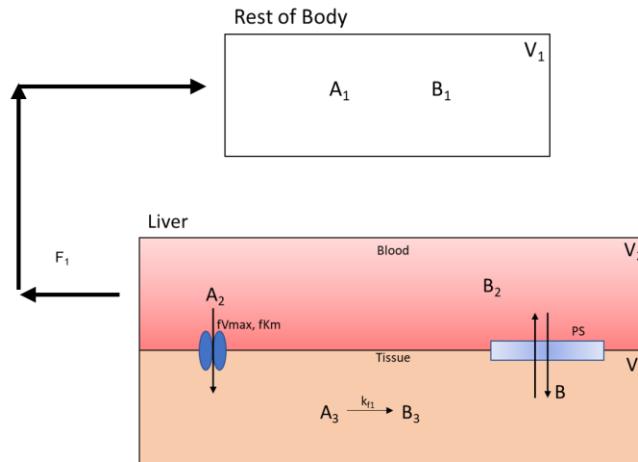
**Keep Active on Add**
Add

### Note

When adding **B**, we are reusing the flow variables **F**, **F\_1**, and **F\_2**. BioModME allows multiple saving of parameters but will check that the parameters have the same base unit. If not, the parameter will be rejected. The value for the parameter will overwrite on each new overwrite of the parameter.

## 2. Flow Back: Liver to Body

The flow back from the liver to the rest of the body looks like:



This is like the previous step but with the compartments switched around.

Steps:

1. Make sure **Options** is still on **Flow Between Compartments**.
2. Turn off **Split Flow**.
3. Set the information for the compartment the flow is leaving from. It has the following specifications:
  - **Flow Out Of:** Liver\_Blood
  - **Species Out:** A\_2
  - **Flow Variable:** F\_1
  - **Flow Value:** 3
4. Set the information for the compartment the flow is entering. It has the following specifications:
  - **Flow Into:** Rest\_Of\_Body
  - **Species In:** A\_1
  -
5. Press the **Add** button to add this flow.

1 Options Flow Between Compartments

2 Split Flow Number of Splits 2

Flow Out Of	Species Out	Flow Variable	Flow Value (l/min)
Liver_Blood	A_2	F_1	3

Flow Into	Species In
Rest_Of_Body	A_1

3 4

5 Keep Active on Add Add

6. Repeat the previous steps for species **B**. This should involve only changing the **Species** dropdowns.

Options Flow Between Compartments

Split Flow Number of Splits 2

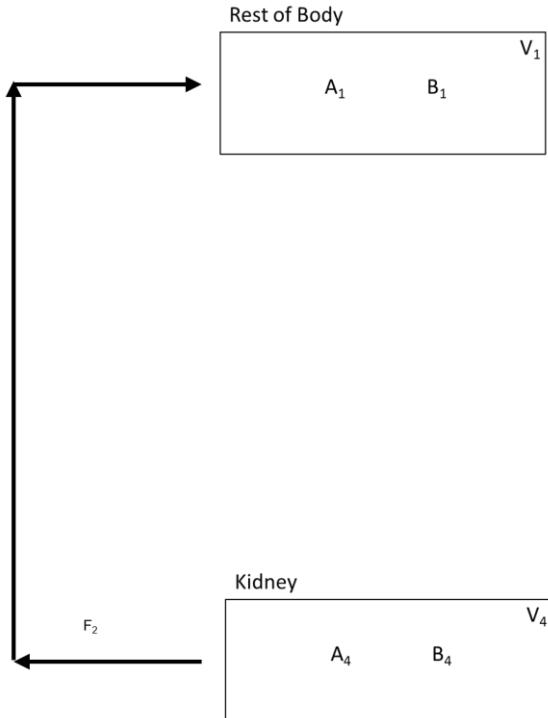
Flow Out Of	Species Out	Flow Variable	Flow Value (l/min)
Liver_Blood	B_2	F_1	3

Flow Into	Species In
Rest_Of_Body	B_1

Keep Active on Add Add

### 3. Flow Back: Kidney to Body

The flow back from the kidney to the rest of the body is:



Steps:

1. Make sure **Options** is still on **Flow Between Compartments**.
2. Check that **Split Flow** is off.
3. Set the information for the compartment the flow is leaving from. It has the following specifications:
  - **Flow Out Of:** Kidney
  - **Species Out:** A<sub>4</sub>
  - **Flow Variable:** F<sub>2</sub>
  - **Flow Value:** 2
4. Set the information for the compartment the flow is entering. It has the following specifications:
  - **Flow Into:** Rest\_of\_Body
  - **Species In:** A<sub>1</sub>
5. Press the **Add** button to add this flow.

1 Options  
Flow Between Compartments

2  Split Flow  
Number of Splits  
2

3 Flow Out Of Species Out  
Kidney A\_4

4 Flow Into Species In  
Rest\_Of\_Body A\_1

5  Keep Active on Add Add

6. Repeat the previous steps for species **B**. This should involve only changing the **Species** dropdowns.

1 Options  
Flow Between Compartments

2  Split Flow  
Number of Splits  
2

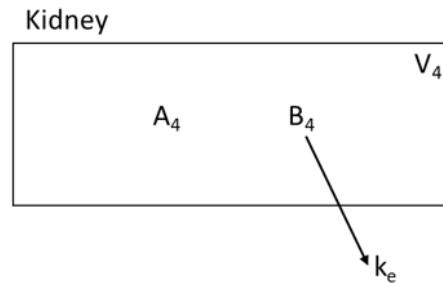
3 Flow Out Of Species Out  
Kidney B\_4

4 Flow Into Species In  
Rest\_Of\_Body B\_1

5  Keep Active on Add Add

#### 4. Clearance of A From Kidney

Drug A is excreted from the kidney at a constant rate. The isolated process is shown below:



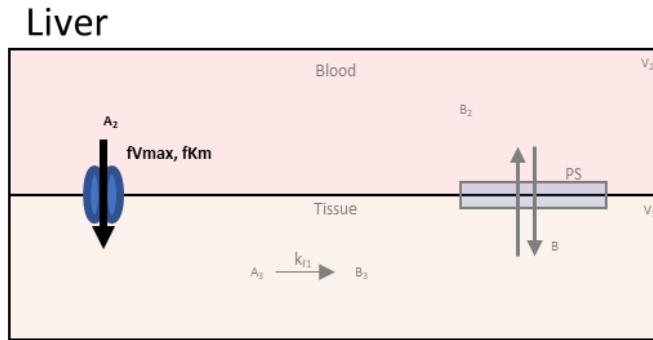
Steps:

1. Select **Clearance** in the **Options** dropdown.
2. Enter the following information in the main box:
  - **Compartment:** Kidney
  - **Species:** B\_4
  - **Rate:** k\_e
3. Press the **Add** button to add the clearance of B from the kidney.

Options Clearance	(1) ▾	Compartment Kidney	Species B_4	Flow Rate Variable k_e	Value (1/min) 0
<input checked="" type="checkbox"/> Keep Active on Add			(2) ▾	(3)	Add

## 5. Facilitated Diffusion of A

The next two seconds will look at diffusion processes from the liver blood to the liver tissue and back. Below is the facilitated diffusion of molecule **A** from the liver blood to the liver tissue.



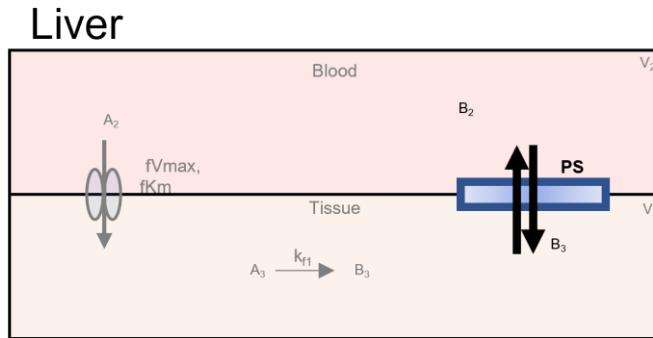
Steps:

1. Select **Facilitated Diffusion** in the **Options** dropdown.
2. Enter the following information in the first row of the main box:
  - **From Compartment:** Liver\_Blood
  - **From Species:**  $A_2$
  - **Vmax:**  $fVmax\_1$
  - **Km:**  $fKm\_1$
3. Enter the following information in the second row of the main box:
  - **To Compartment:** Liver\_Tissue
  - **From Species:**  $A_3$
4. Press the **Add** button to add this facilitated diffusion flow to the model.

<b>Options</b>	<b>From Compartment</b>	<b>From Species</b>	<b>Vmax</b>	<b>Value</b>
Facilitated Diffusion	Liver_Blood	$A_2$	$fVmax\_1$	0
<b>To Compartment</b>	<b>To Species</b>	<b>Km</b>	<b>Value</b>	
Liver_Tissue	$A_3$	$fKm\_1$	0	

Keep Active on Add      4      Add

## 6. Simple Diffusion of B



Steps:

1. Select **Simple Diffusion** in the **Options** dropdown.
2. Enter the following information in the first row of the main box:
  - **Compartment:** Liver\_Blood
  - **Species:** B\_2
  - **Permeability surface area product:** PS
3. Enter the following information in the second row of the main box:
  - **Compartment:** Liver\_Tissue
  - **Species:** B\_3
4. Press the **Add** button to add this simple diffusion flow to the model.

<b>Options</b>	<b>Compartment</b>	<b>Species</b>	<b>Diffusivity Coefficient</b>	<b>Value</b>
Simple Diffusion 1	Liver_Blood	B_2 2	PS	0
	Compartment	Species		
	Liver_Tissue	B_3 3		
<input type="checkbox"/> Keep Active on Add	4 Add			

### Note

The order of entered compartments and species does not matter for simple diffusion.

This should be the last term entered in the Input/Output box. There should be nine terms in the results table.

## Parameters

Navigate to the next box: **Parameters**.

Below is the generated parameter table that has been created so far by the above entered information. All descriptions are pregenerated based on where the parameter originated. Like other tables, all values are editable.

View By:				
	Name	Value	Unit	Description
1	V_1	3000.00	mL	Volume of compartment_1
2	V_2	1250.00	mL	Volume of compartment_2
3	V_3	1000.00	mL	Volume of compartment_3
4	V_4	750.00	mL	Volume of compartment_4
5	k_f1	0.00	1/min	Forward rate constant for the reaction of A_3 to B_3
6	F	5.00	L/min	Flow rate out from Rest_Of_Body, Flow rate out from Rest_Of_Body
7	F_1	3.00	L/min	Flow rate from Rest_Of_Body to Liver_Blood, Flow rate from Rest_Of_Body to Liver_Blood, Flow rate out from Liver_Blood, Flow rate out from Liver_Blood
8	F_2	2.00	L/min	Flow rate from Rest_Of_Body to Kidney, Flow rate from Rest_Of_Body to Kidney, Flow rate out from Kidney, Flow rate out from Kidney
9	k_e	0.00	1/min	Clearance rate constant for B_4 of compartment Kidney
10	fVmax_1	0.00	nmol/min	Maximum velocity for the facilitated Diffusion of A_2 to A_3
11	fKm_1	0.00	nmol	Michaelis Menten constant for the facilitated Diffusion of A_2 to A_3
12	PS	0.00	L/min	Solubility constant for the simple diffusion of B_2 to B_3

The parameter values for the model are as follows:

Parameter	Value	Unit
F	5	L/min
F_1	3	L/min
F_2	2	L/min
fVmax	100	nmol/min
fKm	10000	nmol/L
PS	15	ml/min
k_f1	6	1/min
k_e	4	1/min
V_1	3000	ml
V_2	1250	ml
V_3	1000	ml
V_4	750	ml

## Note

In the Figure above, we did not keep the units of our parameters consistent. In some cases, ml was used and in other liters was used. This is okay. The application converts all units to a “base” unit in the background and performs all calculations at that base level.

The below figure shows the parameter table with the values entered.

View By:

All

	Name	Value	Unit	Description
1	V_1	3000.00	mL	Volume of compartment_1
2	V_2	1250.00	mL	Volume of compartment_2
3	V_3	1000.00	mL	Volume of compartment_3
4	V_4	750.00	mL	Volume of compartment_4
5	k_f1	6.00	1/min	Forward rate constant for the reaction of A_3 to B_3
6	F	5.00	L/min	Flow rate out from Rest_Of_Body, Flow rate out from Rest_Of_Body
7	F_1	3.00	L/min	Flow rate from Rest_Of_Body to Liver_Blood, Flow rate from Rest_Of_Body to Liver_Blood, Flow rate out from Liver_Blood, Flow rate out from Liver_Blood
8	F_2	2.00	L/min	Flow rate from Rest_Of_Body to Kidney, Flow rate from Rest_Of_Body to Kidney, Flow rate out from Kidney, Flow rate out from Kidney
9	k_e	4.00	1/min	Clearance rate constant for B_4 of compartment Kidney
10	fVmax_1	100.00	nmol/min	Maximum velocity for the facilitated Diffusion of A_2 to A_3
11	fKm_1	10000.00	nmol	Michaelis Menten constant for the facilitated Diffusion of A_2 to A_3
12	PS	15.00	L/min	Solubility constant for the simple diffusion of B_2 to B_3

## Differential Equations

Navigate to the final box “**Differential Equations**”.

This tab will begin with all species being equal to “0” and gets auto-generated based on reactions added. The "Generate" button will refresh the equations, if for any reason they need to be refreshed.

### Differential Equations

- ≡

$$(1) V_1 \frac{dA_1}{dt} = -(F * A_1) + (F_1 * A_2) + (F_2 * A_4)$$

$$(2) V_2 \frac{dA_2}{dt} = +(F_1 * A_1) - (F_1 * A_2) - \left( \frac{fVmax_1 * A_2}{fKm_1 + A_2} \right)$$

$$(3) V_3 \frac{dA_3}{dt} = -(V_3 * (k_{f1} * A_3)) + \left( \frac{fVmax_1 * A_2}{fKm_1 + A_2} \right)$$

$$(4) V_4 \frac{dA_4}{dt} = +(F_2 * A_1) - (F_2 * A_4)$$

$$(5) V_1 \frac{dB_1}{dt} = -(F * B_1) + (F_1 * B_2) + (F_2 * B_4)$$

$$(6) V_2 \frac{dB_2}{dt} = +(F_1 * B_1) - (F_1 * B_2) - (PS * (B_2 - B_3))$$

$$(7) V_3 \frac{dB_3}{dt} = +(V_3 * (k_{f1} * A_3)) + (PS * (B_2 - B_3))$$

$$(8) V_4 \frac{dB_4}{dt} = +(F_2 * B_1) - (F_2 * B_4) - (k_e * B_4 * V_4)$$

Generate

### Note

The right sidebar of the “System of Differential Equations” box has an option for viewing the differential equations including turning off the math rendering and showing each derivation line as a newline for each differential equation

## Results

In this section, we will cover executing the model and its different outputs.

### Solve Model

Navigate to the “Execute Model” tab on the lefthand tabbar. This tab solves our model and provides us with the options to customize the solver. For this model, we will just solve using the standard options:

1. Enter the following times: Starting = 0, End = 5, Step = 0.001, Unit = hr. This generates the model to be solved and integrated at time points 0, 0.001, 0.002.....5.998, 5.999, 6.
2. Press the “Run Solver” button.
3. An output table should be generated with the concentration of each species along each time step of the solved model.

You can download this solved model data to a csv file in the download tab on this page. Post processing is not used in this tutorial.

The screenshot shows the 'Options' tab selected in the top navigation bar. Below it, there are four input fields: 'Starting Time' (0), 'End Time' (5), 'Time Step' (0.001), and 'Unit' (hr). To the right of these fields are two checkboxes: 'Advanced Options' and 'Viewing Options'. At the bottom right is a large 'Run Solver' button.

### Note

BioModME is set by default to show the results table rounded to 3 digits. Most terms in these results at the beginning will round to 0. To change this, check the **Viewing Options** box and turn off the **Round** option or turn on the **Scientific Notation** option.

Below is how the table for this model should look after the initial solve:

time (hr)	A_1	A_2	A_3	A_4	B_1	B_2	B_3	B_4
0	0	0	0	0	0	0	0	0
0.001	0	0	0	0	0	0	0	0
0.002	0	0	0	0	0	0	0	0
0.003	0	0	0	0	0	0	0	0
0.004	0	0	0	0	0	0	0	0
0.005	0	0	0	0	0	0	0	0
0.006	0	0	0	0	0	0	0	0
0.007	0	0	0	0	0	0	0	0
0.008	0	0	0	0	0	0	0	0
0.009	0	0	0	0	0	0	0	0
0.01	0	0	0	0	0	0	0	0

To see the values for the model in a good form, we will change the viewing options:

1. Click the **Viewing Options** Checkbox.
2. Change Results Units to **umol**.

Viewing Options

---

**Viewing Options**

Scientific Notation     Round

Digits      Round to

Results Units

umol

---

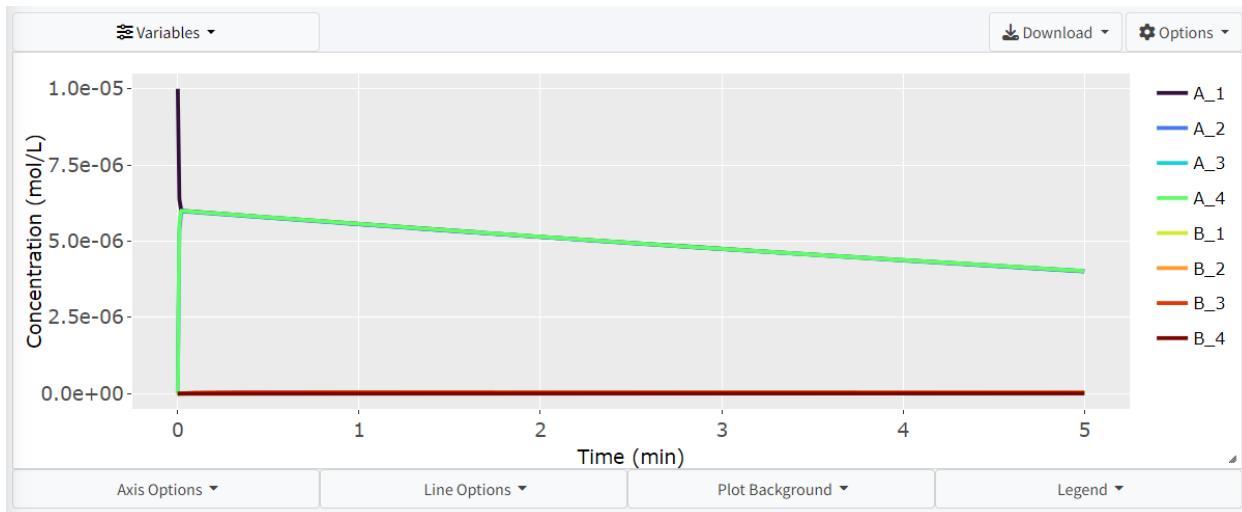
Show All ▾ entries

time (hr)	A_1	A_2	A_3	A_4	B_1	B_2	B_3	B_4
1	0	10	0	0	0	0	0	0
2	0.001	9.111	1.283	0	1.415	0	0	0
3	0.002	8.421	2.287	0.001	2.503	0	0	0
4	0.003	7.884	3.073	0.002	3.339	0	0	0
5	0.004	7.466	3.689	0.003	3.98	0	0.001	0.001
6	0.005	7.141	4.173	0.003	4.472	0	0.001	0.001
7	0.006	6.889	4.553	0.004	4.848	0	0.001	0.002
8	0.007	6.692	4.851	0.004	5.135	0	0.001	0.003
9	0.008	6.538	5.087	0.005	5.354	0	0.002	0.004
10	0.009	6.419	5.272	0.005	5.52	0	0.003	0.004

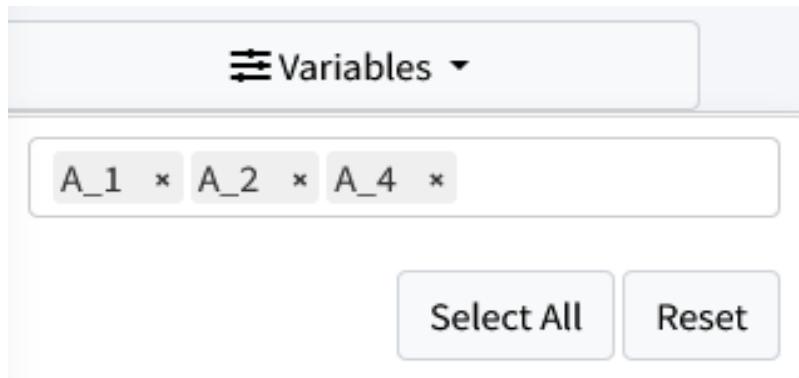
In this section, we will examine the plotting of this model.

## Visualization

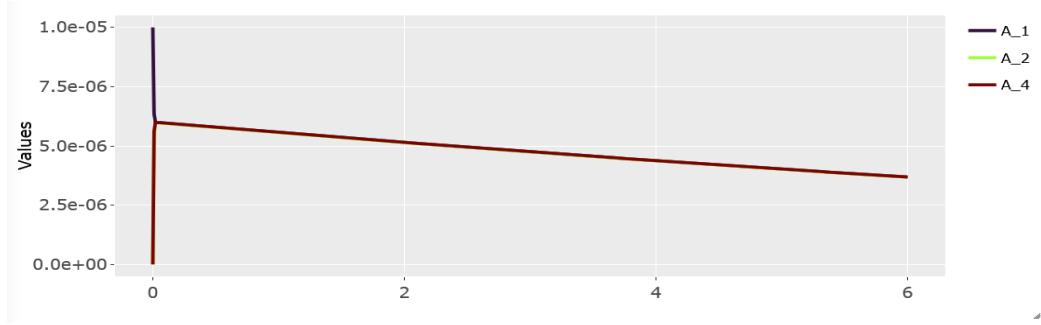
Move to the **Visualization** tab. The starting plot from the model should look like the following:



For this problem we want to examine how the concentration of A varies throughout the bloodstream (A<sub>1</sub>, A<sub>2</sub>, A<sub>4</sub>). Click the **Variable** dropdown and remove all necessary variables:



The resulting plot should look like:

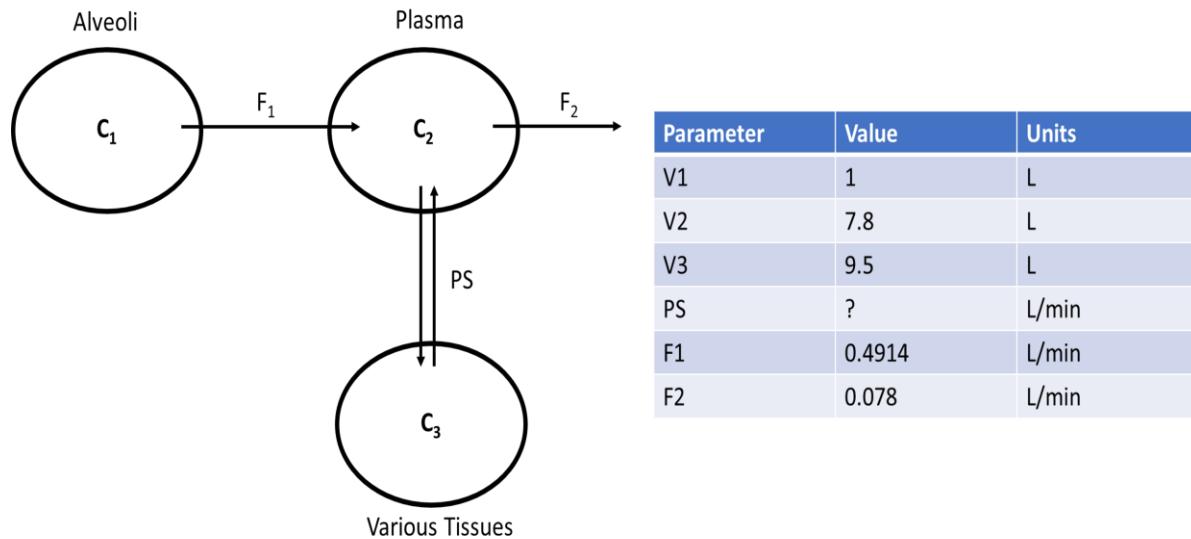


## Parameter Estimation Tutorial

This is a basic parameter estimation tutorial. To download the example data used in this tutorial go to [https://github.com/Jwomack7512-bio/BobTheBuilder/blob/main/shared\\_files/Nicotine\\_cigarette.csv](https://github.com/Jwomack7512-bio/BobTheBuilder/blob/main/shared_files/Nicotine_cigarette.csv).

We assume you have basic knowledge of creating a model and will skip showing every step of that process. Please see the application documentation for details on the solver and options available to use with the parameter estimation solver.

We will construct the following simple three compartment model:



In the above diagram, we model the concentration of a drug, Nicotine (**C**), as it moves from the alveoli of the lungs, through the plasma of the body, and a compartment summarizing various tissues of the body. We assume Nicotine flows out of the alveoli with flow rate (**F\_1**), into the plasma of the body. From there, Nicotine diffuses to various tissues of the body through simple diffusion (**PS**) and is also removed from the plasma at a flow rate of **F\_2**. In this model, we are unsure of the value of **PS** and will use parameter estimation to estimate the value from experimental data that found the concentration of nicotine in the plasma at various timepoints. The starting concentration of **C** in the alveoli is 1.2 mol/L.

## Build Model

To begin building the model, we will add our three compartments with their corresponding volumes:

- Compartment: Alveoli, Volume = 1 L
- Compartment: Plasma, Volume = 7.8 L
- Compartment: Various\_Tissues, Volume = 9.5 L

### Compartments

Name	Volume	Value	Unit	Description
Alveoli	V_1	1.00	L	
Plasma	V_2	7.80	L	
Various_Tissues	V_3	9.50	L	

+

-

Next, we will add a concentration (C) variable for nicotine to each compartment and give an initial concentration of 1.2 mol/L to C\_1.

### Species

Active Compartment  Show Active Compartment Only  
Alveoli   Add To All Compartments

	Name	Value	Unit	Compartment	Description
1	C_1	1.20	mol/L	Alveoli	
2	C_2	0.00	mol/L	Plasma	
3	C_3	0.00	mol/L	Various_Tissues	

+

-

Next, we need to add the compartment input/outputs. We have three to add:

#### 1. Flow between Alveoli and Plasma

Options Flow Between Compartments

Flow Out Of	Species Out	Flow Variable	Flow Value (L/min)
Alveoli	C_1	F_1	0.4914

Flow Into Species In

Plasma	C_2
--------	-----

Split Flow Number of Splits 2

Keep Active on Add

Add

## 2. Flow out of Plasma

Options

Compartment Species Flow Variable Flow Value (L/min)

Plasma C\_2 F\_2 0.078

Keep Active on Add

Add

## 3. Simple diffusion between Plasma and Various\_Tissues

Options

Compartment Species Diffusivity Coefficient Value (l/min)

Plasma C\_2 PS 0

Compartment Species

Various\_Tissues C\_3

Keep Active on Add

Add

With all our model components constructed, we need to assign parameter values. We do not know the value of PS but we will give the model a reasonable starting estimate (0.5 L/min).

### Parameters

**View By:**

All

	Name	Value	Unit	Description
1	V_1	1.00	L	Volume of compartment_1
2	V_2	7.80	L	Volume of compartment_2
3	V_3	9.50	L	Volume of compartment_3
4	F_1	0.49	L/min	Flow rate out from Alveoli
5	F_2	0.08	L/min	Flow rate out of Plasma
6	PS	0.50	L/min	Solubility constant for the simple diffusion of C_2 to C_3

Note:

The parameter table rounds to two decimal places. This is why F\_2 shows as 0.08 in the table. If you double-click the value, it will show as the appropriate 0.078.

For reference, here are the solved differential equations:

## Differential Equations

- ⚙

$$(1) V_1 \frac{dC_1}{dt} = - (F_1 * C_1)$$

$$(2) V_2 \frac{dC_2}{dt} = + (F_1 * C_1) - (F_2 * C_2) - (PS * (C_2 - C_3))$$

$$(3) V_3 \frac{dC_3}{dt} = + (PS * (C_2 - C_3))$$

Generate

Move to the **Execute Model** tab and enter the time for this model:

- Start Time = 0
- End Time = 70
- Time Step = 1
- Unit = min

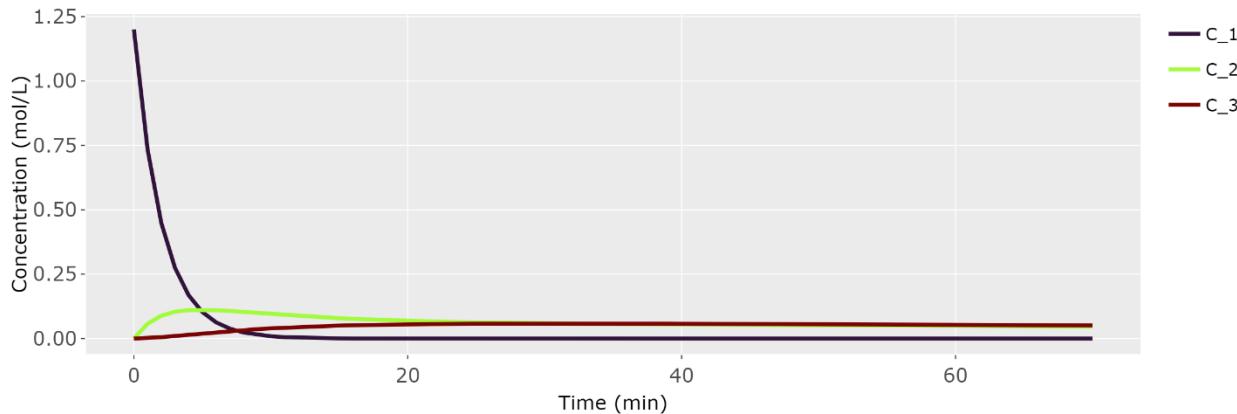
The screenshot shows the 'Execute Model' interface with the 'Options' tab selected. The 'Starting Time' is set to 0, 'End Time' is set to 70, 'Time Step' is set to 1, and 'Unit' is set to 'min'. There are two checkboxes: 'Advanced Options' (unchecked) and 'Viewing Options' (unchecked). A 'Run Solver' button is at the bottom right.

The beginning of the results table should look like this:

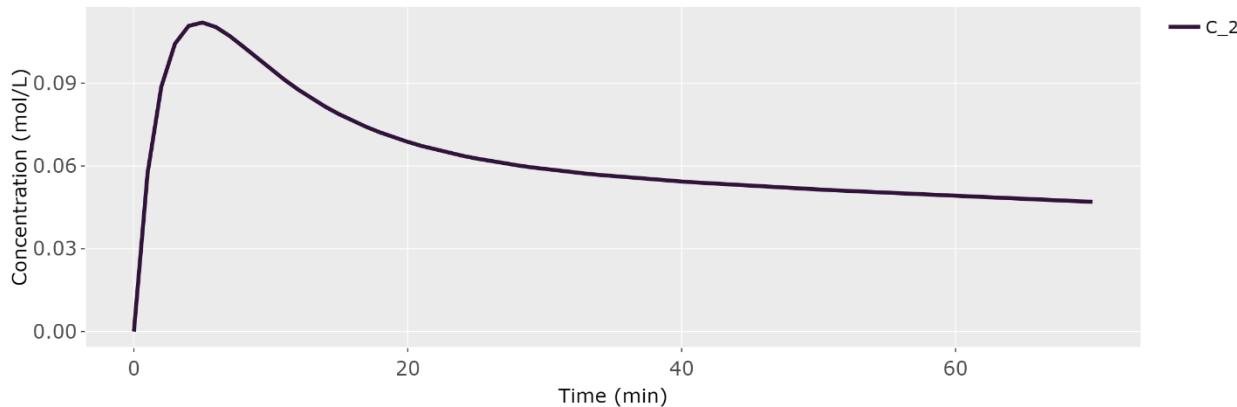
time (min)	C_1	C_2	C_3
0	1.2	0	0
1	0.734	0.057	0.002
2	0.449	0.089	0.005
3	0.275	0.104	0.01
4	0.168	0.111	0.015
5	0.103	0.112	0.02
6	0.063	0.11	0.025
7	0.038	0.107	0.029
8	0.024	0.103	0.033
9	0.014	0.099	0.036

## Visualize Model

Move to the **Visualization** tab. A starting plot should be generated that looks like:



We want to examine the concentration of Nicotine in plasma (**C\_2**). Go into the **Variables** dropdown and remove variables **C\_1** and **C\_3**.



We should import our data for Nicotine in the plasma to overlay with the plot. Scroll below the plot to the box header that is titled **Import Data**. Browse your desktop for the nicotine dataset (which can be downloaded from our GitHub). Be sure to click the **Apply Overlay** checkbox to show the imported data. For importing data, the first row of the data is the headers. The first column should be time while the remaining columns should be your concentrations at those times. The header names have to match the model names for the data to be correctly plotted.

**Import Data**

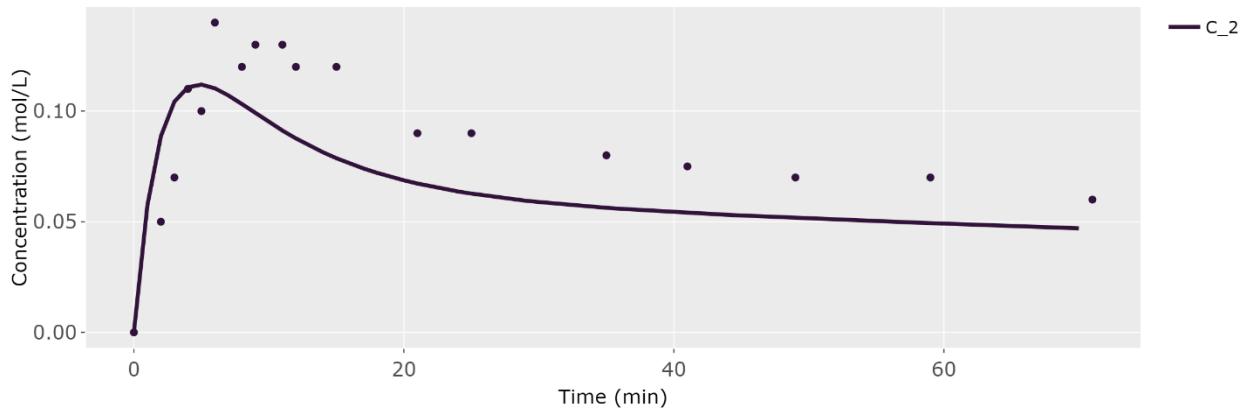
**Import Data** **Time** **y**  **Apply Overlay**

Browse... Nicotine\_cigarette.csv time C\_2

Upload complete

	time	C_2
1	0	0.00
2	2	0.05
3	3	0.07
4	4	0.11
5	5	0.10
6	6	0.14
7	8	0.12
8	9	0.13
9	11	0.13
10	12	0.12
11	15	0.12
12	21	0.09
13	25	0.09
14	35	0.08
15	41	0.08
16	49	0.07
17	59	0.07
18	71	0.06

The resulting plot should look like the figure below:



In this overlayed plot, we can see our model has the general shape of our data but doesn't quite mimic it well. We will run parameter estimation for PS to see if we can find a better fit.

## Perform Parameter Estimation

Go to the **Modeler's Toolbox Tab**, subtab **Parameter Estimation**. Here, we will start by importing our data into this module.

The screenshot shows the 'Import Observational Data' interface. A file named 'Nicotine\_cigarette.csv' is selected in the 'Browse...' field. The 'Upload complete' button is highlighted in blue at the bottom of the file selection area. There is also a 'View Table' checkbox.

Move to the next box on the page **Select Parameters**. Use the dropdown to select **PS**.

The screenshot shows the 'Select Parameters to Estimate' interface. A dropdown menu is open, showing 'PS' selected. Below it is a table with one row:

	Parameters	Initial Guess	Lower Bound	Upper Bound	Calculated
1	PS	0.50	0.00	1.00	-

A blue 'Run' button is located on the right side of the interface.

There are five columns in the generated table. They are as follows:

- Parameters - the selected values to estimate.
- Initial Guess - the starting value to begin parameter estimation at.
- Lower Bound - the lowest acceptable value this parameter can be.
- Upper Bound - the highest acceptable value this parameter can be.
- Calculated Value - the found value from parameter estimation after calculations.

Values for Lower and Upper bound can be left blank if no bounds want to be used. Here we use the following:

- Initial Guess - 0.50
- Lower Bound - 0
- Upper Bound - 1

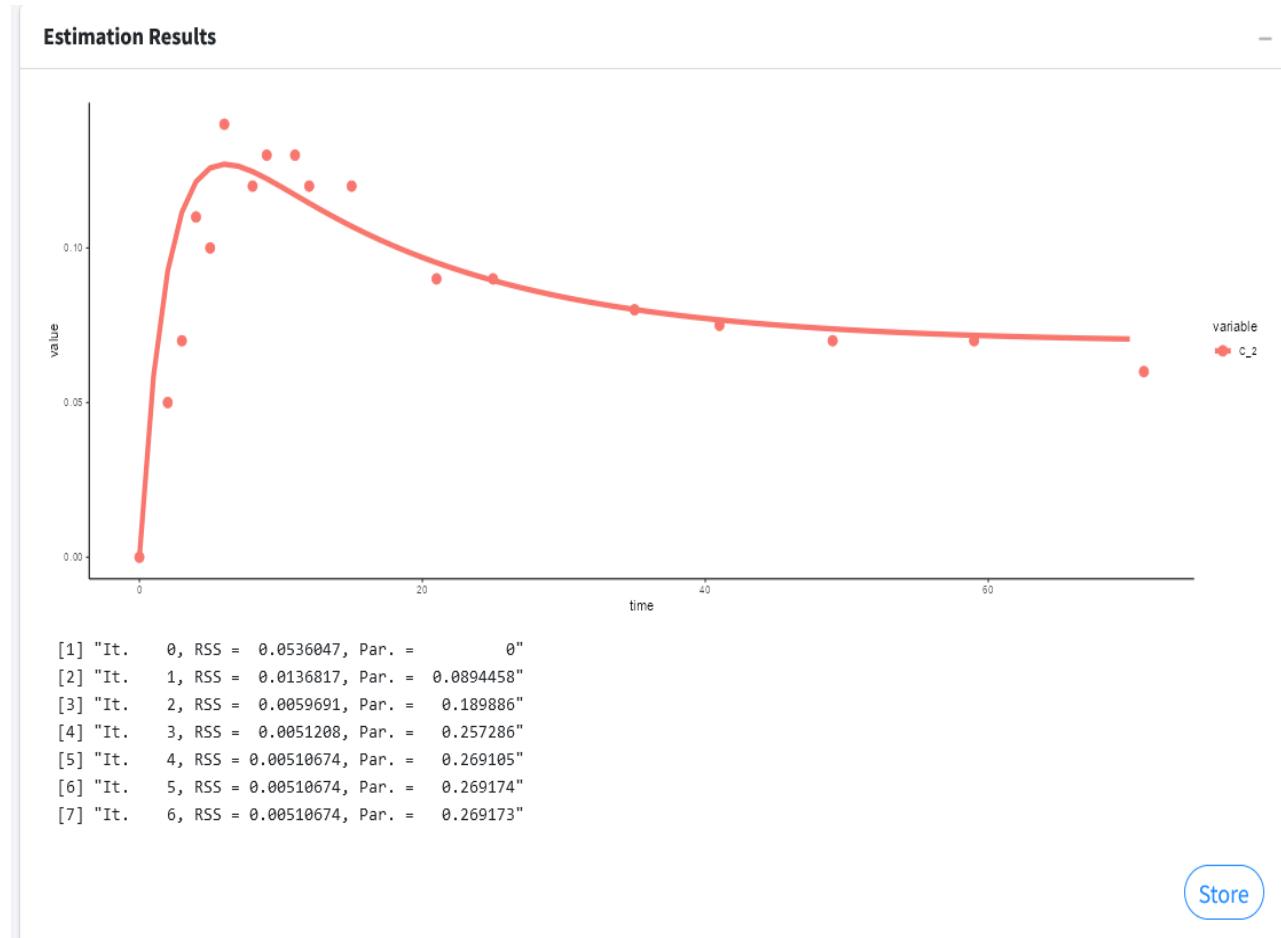
Press the **Run** button and the program should output a value of approximately 0.10.

The screenshot shows the 'Select Parameters to Estimate' interface again. The 'Run' button has been pressed, and the calculated value has been updated to 0.269172931192078. The table now looks like this:

	Parameters	Initial Guess	Lower Bound	Upper Bound	Calculated
1	PS	0.50	0.00	1.00	0.269172931192078

A blue 'Run' button is located on the right side of the interface.

The next box **Estimation Results** will contain the model fit of the variable with its corresponding data along with the results of the iterations of the parameter estimation algorithm.



Press the **Store** button to overwrite your current parameters in the model with the estimated values.

**Note:**

You can estimate as many parameters as you want with this setup. Just note that the more uncertainty you introduce into your model the longer the algorithm can take to find a solution. It can also affect the probability of finding a solution.