Getting Started Guide for MATLAB Toolbox MENOTR

Here we present a user guide to be used in conjunction with our published manuscript. This guide will detail recommended background materials for better understanding MATLAB as well as how to edit and implement MENOTR. These instructions include both this written document as well as video tutorials that can be found online.

Contents:

Section 1: Software Requirements and MATLAB Basics

Section 2: Use MENOTR to Optimize Parameters in Closed Form Expression

Section 3: Use MENOTR to Optimize Parameters within Chemical Reaction Scheme

Section 4: Use MENOTR to Optimize Parameters Using Numerical Solutions to the Laplace Transform

Section 5: Use MENOTR to Optimize Parameters Using Implicit Fitting

Section 6: Use MENOTR to Calculate Parameter Uncertainties Using Monte-Carlo Simulations

Section 7: Use MENOTR to Calculate Parameter Uncertainties Using Grid-Search Analysis

Section 8: Use MENOTR to Simulate a Data Set with User Defined Parameter Values

Section 1: Software Requirements and MATLAB Basics

This toolbox was optimized to run on MATLAB 2021a.

MATLAB comes with a set of default toolboxes that can be used to carry out the majority of the basic functions of the MENOTR toolbox; however, there are some additional Add-Ons that will be required in order to use the software in its entirety. The list below are not recommendations, they are requirements.

Necessary MATLAB Toolboxes for MENOTR

- Communications Toolbox
- Optimization Toolbox
- Parallel Computing Toolbox
- Signal Processing Toolbox
- Statistics and Machine Learning Toolbox
- Symbolic Math Toolbox

Problem Specificity Requirements

The MENOTR toolbox, as written, can be modified to accommodate different models; however, this will require some user adjustments within specific sections of the code. The two main things that the user will need to provide are (1) a fitting equation that relates the independent variable to the experimental observable and (2) a method to solve that equation to generate a simulated data set to determine chi-squared values for a given set of parameters. The method of solving the equations is going to depend on the equation itself. For example, a closed-form expression like y = m*x + b can be simply solved using the built-in arithmetic logic of MATLAB. While the use of root finders for implicit equations, ordinary differential equation solvers for systems of ODEs, or transforms for equations expressed in alternative domains (Laplace or Fourier) may be needed depending on the fitting equation at hand.

New User to MATLAB

MathWorks offers a few beginner training sessions that are accessible online at https://matlabacademy.mathworks.com/. A new user can get acquainted with both MATLAB basics and higher-level skills. We recommend MATLAB Onramp and MATLAB Fundamentals as a starting point for using MENOTR. A few key points to be aware of when using MENOTR in the MATLAB program are described below.

General Layout of the Toolbox

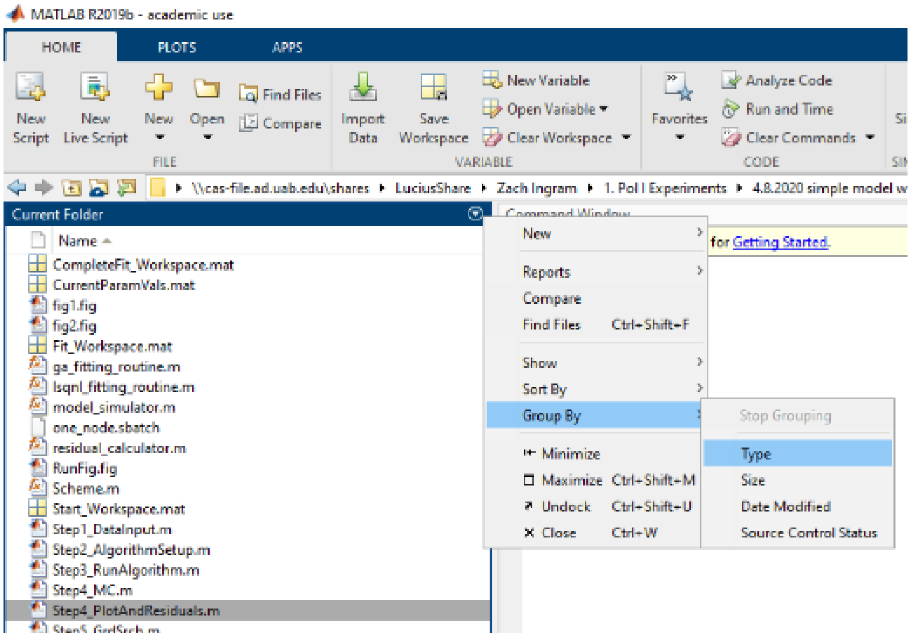

Figure 1: MATLAB GUI showing the "Current Folder" window. The figure is illustrating how to change the setting of this window to group files by type.

When opening MATLAB and navigating to a working directory the files within that directory are shown in “Current Folder” window in alphabetical order regardless of file type. We have found it helpful to have MATLAB group these files by their type. This setting can be applied by navigating to the top right corner of the “Current Folder” window. There you will find a drop down menu, denoted with a downwards points arrow in a circle (as shown in **Fig. 1**). Under “Group By” select “Type” and MATLAB should now display the files grouped into separate types.

A brief description of the different file types is discussed below:

- Files
  - This is a section where MATLAB places files that it cannot group into one of the other categories
- Figures
  - Any plots, images, or other figures that are generated by MATLAB are placed in this section.
- Scripts
  - Ordered blocks of code that are interpreted as set of commands by MATLAB.
- Functions
  - A subset of a script that generates a set of outputs when given a specific set of inputs. These are called within the general scripts to perform desired tasks. These contain the "guts" of the code and are used to perform most of the heavy lifting of the fitting routine.
- MAT-files
  - These files contain MATLAB workspaces where variables have been saved. While a MENOTR script is running and as it finishes, it will create a set of these files that contain the optimization outputs.

Section 2: Use MENOTR to Optimize Parameters in Closed Form Expression

Shown below is a general overview of the steps necessary to optimize parameters using a closed form expression such as y = m*x+b.

A link to a detailed tutorial video outlining how to use MENOTR to fit data to a line is also available. Please see the link below.

Video Tutorial - https://youtu.be/g12WHngALVo

It is also recommended for ease of use that users make a specific "Working Directory" outside of the "master" folder where the scripts can be immediately copied and subsequently edited for use. This will maintain the integrity of the files in the master folder and allow users to start with a fresh, error-free script in the event of a catastrophic issue during future use.

1. Copy "001_Closed Form Equation"

a. Copy over the folder "001_Closed Form Equation" in the MENOTR master folder to your working folder

b. Copy the directory path into or navigate to the directory in MATLAB

2. Edit model_simulator.m

a. **Modify Section 2**

   i. Define the independent variable (i.e., x)

   ii. Define each parameter (i.e., m and b) (Note: Pay close attention to the order of the parameters. Later portions of the code will only show lists of numbers and it's up to the user to remember the identity of the first parameter, the second, and so on.)

b. **Modify Section 3**

   i. Define fit equation using the independent variable and parameters you defined in Section 2 (i.e., y=mx+b).

3. <u>Run Step1_DataInput.m</u>

   a. Run the script.

   a. A matrix should appear after running Step1.

   b. Copy and paste the experimental data into the matrix as a series of columns in the form : [Independent Variable, Data set 1, St.Dev.1, Data set 2, St.Dev.2, … etc.]

4. <u>Edit Step2_AlgorithmSetup.m</u>

   a. **Modify Section 1**

      i. Define parameters used in your expression.

      ii. Define initial guess values along with upper bound values and lower bound values, denoted as "…_ub" and "…_lb" respectively.

      iii. Modify the params_vals, params_ub, params_lb, and the clear command portions in the code to match the naming convention of the parameters used.

   b. **Modify Section 2**

      i. Define parameters used in your fitting equation. The "_c" notation indicates that you're working within the "cycles" portion of the code. In this portion of the code identify which parameters are "floating" and which are "not floating". (1 is floating while 0 is don't float) Floating here means different values for the given parameter are probed. While not floating means the parameter is fixed at a particular value.

      ii. Identify if NLLS should be run within that cycle. (The first cycle does not involve NLLS and the second includes NLLS by default.)

      iii. Edit the "CyMat" matrix to accommodate the parameters present within your model.

      iv. Update parameter names in the "clear" portion of the code at the bottom of this section.

5. <u>Run Step2_AlgorithmSetup.m</u>

6. <u>Run Step 3_RunAlgorithm.m</u>

When the optimization procedure is complete, the workspace will contain the resultant optimized parameters. The optimized parameter values can be found in the workspace element titled “Fit_ParameterValues”. The order of the values corresponds to how they were setup in the model_simulator with the first value being param(1), the second param(2), and so on.

Section 3: Use MENOTR to Optimize Parameters within Chemical Reaction Scheme

Shown below is a general overview of the steps necessary to optimize parameters within a chemical reaction scheme. The ordinary differential equation (ODE) for each species within your scheme should be constructed prior to using MENOTR. The system of ODEs is one necessary component when optimizing parameters in this manner.

A link to a detailed tutorial video outlining how to use MENOTR in fits of data using a chemical reaction scheme is also available. Please see the link below.

Video Tutorial - https://youtu.be/dD3-V13Ln84

It is also recommended for ease of use that users make a specific “Working Directory” outside of the “master” folder where the scripts can be immediately copied and subsequently edited for use. This will maintain the integrity of the files in the master folder and allow users to start with a fresh, error-free script in the event of a catastrophic issue during future use.

1. Copy “002_Chemical Reaction Scheme”
    a. Copy over the folder “002_Chemical Reaction Scheme” in the MENOTR master folder to your working folder
    b. Copy the directory path into or navigate to the directory in MATLAB.

2. Edit scheme.m
    a. **Modify Section 1**
        i. Define species within reaction scheme using individual alphabetical characters.

    b. **Modify Section 2**
        i. Input the ordinary differential equation for each species present in the reaction scheme into the matrix. (Make sure they are input in the same order as in Section 1.)

3. Edit model_simulator.m
    a. **Modify Section 2**
        i. Define kinetic parameters
        ii. Input the initial concentrations of each species within your reaction scheme.

iii. The simulated time courses monitor the concentration of a particular species as a function of time. If you would like to normalize the time course to the starting concentration for that species or some other value, input the desired value under “normalization = value”. If you would like your time course in concentration terms without being normalized then change the “normalization” value to 1.

b. **Modify Section 4**

i. Identify which of the simulated time courses from your reaction scheme correspond to the experimentally monitored product. (By default, the code is designed to assume that the last species is the monitored product.)

4. Edit Step2_AlgorithmSetup.m

a. **Modify Section 1**

i. Define parameters used in your kinetic model.

ii. Define initial guess values along with upper bound values and lower bound values.

b. **Modify Section 2**

i. Define parameters used in your kinetic scheme. The “_c” notation indicates that you’re working within the “cycles” portion of the code. In this portion of the code identify which parameters are floating and which are static. (1 is floating while 0 is static.)

ii. Identify if NLLS should be run within that cycle. (By default the first cycle does not involve NLLS and the second includes NLLS.)

iii. Change the “CyMat” matrix to accommodate the parameters present within your model.

5. Run Step1_DataInput.m

a. Run the script.

b. A matrix should appear after running Step1.

c. Copy and paste the experimental data into the matrix as a series of columns in the form : [Independent Variable, Data set 1, St.Dev.1, Data set 2, St.Dev.2, … etc.]

6. Run Step2_AlgorithmSetup.m

7. Run Step 3_RunAlgorithm.m

When the optimization procedure is complete, the workspace will contain the resultant optimized parameters. The optimized parameter values can be found in the workspace element titled “Fit_ParameterValues”. The order of the values corresponds to how they were setup in the model_simulator with the first value being param(1), the second param(2), and so on.

Section 4: Use MENOTR to Optimize Parameters Using Numerical Solutions to the Laplace Transform

Shown below is a general overview of the steps necessary to optimize parameters using numerical solutions to the Laplace transform. Please see the link below for a detailed video tutorial.

Video Tutorial - https://youtu.be/BhVziS38whM

It is also recommended for ease of use that users make a specific "Working Directory" outside of the "master" folder where the scripts can be immediately copied and subsequently edited for use. This will maintain the integrity of the files in the master folder and allow users to start with a fresh, error-free script in the event of a catastrophic issue during future use.

1. Copy "003_Numerical Sol of Laplace Transform"

a. Copy over the folder "003_Numerical Sol of Laplace Transform" in the MENOTR master folder to your working folder

b. Copy the directory path into or navigate to the directory in MATLAB.

2. Edit model_simulator.m

a. **Modify Section 2**

    i. Define independent variable

    ii. Define parameters

b. **Modify Section 3**

    i. Define fit equation using the independent variable and parameters you defined in Section 2.

3. Edit Step2_AlgorithmSetup.m

a. **Modify Section 1**

    i. Define parameters used in your expression.

    ii. Define initial guess values along with upper bound values (_ub) and lower bound values (_lb).

    iii. Modify naming convention in params_vals, params_ub, params_lb, and the clear command to match the naming convention of the parameters

b. **Modify Section 2**

    i. Define parameters used in your kinetic scheme. The "_c" notation indicates that you're working within the "cycles" portion of the code. In this portion of the code identify which parameters are floating and which are static. (1 is floating while 0 is static.)

    ii. Identify if NLLS should be run within that cycle. (By default the first cycle does not involve NLLS and the second includes NLLS.)

    iii. Change the "CyMat" matrix to accommodate the parameters present within your model.

    iv. Update parameters in “clear” portion of the code at the bottom of this section.

4. Run_Step1_DataInput.m
    a. Run the script.
    b. A matrix should appear after running Step1.
    c. Copy and paste the experimental data into the matrix as a series of columns in the form : [Independent Variable, Data set 1, St.Dev.1, Data set 2, St.Dev.2, … etc.]

5. Run_Step2_AlgorithmSetup.m

6. Run_Step 3_RunAlgorithm.m

Note: The computer used in the tutorial, is a work horse computer used for difficult calculations in a research environment. Slower run times are expected when using personal laptop computers.

When the optimization procedure is complete, the workspace will contain the resultant optimized parameters. The optimized parameter values can be found in the workspace element titled “Fit_ParameterValues”. The order of the values corresponds to how they were setup in the model_simulator with the first value being param(1), the second param(2), and so on.

Section 5: Use MENOTR to Optimize Parameters Using Implicit Fitting

Shown below is a general overview of the steps necessary to optimize parameters when the use of implicit fitting is necessary.

A link to a detailed tutorial video outlining how to use MENOTR using implicit fitting is also available. Please see the link below.

Video Tutorial - https://youtu.be/jBQhjJXHjlk

It is also recommended for ease of use that users make a specific “Working Directory” outside of the “master” folder where the scripts can be immediately copied and subsequently edited for use. This will maintain the integrity of the files in the master folder and allow users to start with a fresh, error-free script in the event of a catastrophic issue during future use.

1. Copy “004_Implicit Fitting”

   a. Copy over the folder “004_Implicit Fitting” in the MENOTR master folder to your working folder.
   b. Copy the directory path into or navigate to the directory in MATLAB.

2. Edit model_simulator.m
   a. **Modify Section 2**
      i. Define the Xbar equation (be sure to type equation with apostrophes like so: ‘equation’ .
      ii. Define the total macromolecule concentrations (“Mt”).
      iii. Edit code to accommodate the fit parameters .

3. Edit Step2_AlgorithmSetup.m
   a. **Modify Section 1**
      i. Define parameters used in your expression.
      ii. Define initial guess values along with upper bound values and lower bound values.
   b. **Modify Section 2**
      i. Define parameters used in your kinetic scheme. The “_c” notation indicates that you’re working within the “cycles” portion of the code. In this portion of the code identify which parameters are floating and which are static. (1 is floating while 0 is static.)
      ii. Identify if NLLS should be run within that cycle. (By default the first cycle does not involve NLLS and the second includes NLLS.)
      iii. Change the “CyMat” matrix to accommodate the parameters present within your model.
      iv. Update parameters in “clear” portion of the code at the bottom of this section.
4. Run Step1_DataInput.m
   a. Run the script.
   b. A matrix should appear after running Step1.
   c. Copy and paste the experimental data into the matrix as a series of columns in the form : [Independent Variable, Data set 1, St.Dev.1, Data set 2, St.Dev.2, … etc.]

5. Run Step2_AlgorithmSetup.m

6. Run Step 3_RunAlgorithm.m

   When the optimization procedure is complete, the workspace will contain the resultant optimized parameters. The optimized parameter values can be found in the workspace element titled “Fit_ParameterValues”. The order of the values corresponds to how they were setup in the model_simulator with the first value being param(1), the second param(2), and so on.

Section 6: Use MENOTR to Calculate Parameter Uncertainties Using Monte-Carlo Simulations

A general overview the tasks necessary to generate parameter uncertainty on the resultant fit parameters in MENOTR are shown below. A link to a detailed tutorial video outlining how to use MENOTR to determine parameter uncertainty values is also available. Please see the link below.

Video Tutorial - https://youtu.be/AxmV5fYg1ZM

1. Open "005_Parameter Uncertainty Scripts"
   a. Copy "Step4_MC.m" into your working directory

2. Edit Step4_MC.m
   a. **Modify Section 1**
      i. Define number of Monte Carlo simulations that would like to be carried out.
      ii. Define parameters which should be checked for correlation. If you're not wanting to do this, then leave the corrparams matrix empty (corrparams = [ ];) (Plots of the parameters are automatically saved in the working directory.)
3. Run_Step4_MC.m

Section 7: Use MENOTR to Calculate Parameter Uncertainties Using Grid-Search Analysis

A quick list of the tasks necessary to fit data using MENOTR are shown below.

1. Open "005_Parameter Uncertainty Scripts"
   a. Copy "Step5_GrdSrch.m" into your working directory

2. Edit Step4_MC.m
   a. **Modify Section 1**
      i. Modification 1: Identify what parameter you wish to perform grid search on (fxparams = [1]). It's recommended doing one parameter at a time.
      ii. Modification 2: Define the number of parameter values you would like to probed in your grid search. (grdpnts = 10);
      iii. Modification 3: Define the range of parameter values that should be considered in the error contour. This is achieved by assigning a percentage value. i.e. if 30 is chosen, then a contour is generated where the optimized parameter is the minimum and parameter values +/- 30% of the magnitude of the optimized parameter value are considered. The parameter that corresponds to this percentage is divval. (divval = 30)

3. Run_Step5_GrdSrch.m

Section 8: Use MENOTR to Simulate a Data Set With User Defined Parameter Values

This data simulator assumes that you have already created a model_simulator.m file and have edited the file to describe your model of interest. The code shown below will be using this model_simulator.m file in the construction of the simulated data set.

A general overview of the tasks necessary to fit data using MENOTR are shown below.

A link to a tutorial video outlining how to use MENOTR to simulate a data set is also available. Please see the link below.

Video Tutorial - https://youtu.be/qF1SXM6tlZo

1. Open "006_Simulate_Data_Set"
   a. Copy "data_simulator.m" into your working directory. (your working directory should already have a model_simulator.m file from one of the previous mentioned fitting strategies. This new script is going to be using the model_simulator.m file)

2. Edit data_simulator.m
   a. **Modify Section 2**
      i. Define the starting point for your independent variable, the end point, and the number of data points.
   b. **Modify Section 3**
      i. Identify whether you would like the simulated data points to be linearly spaced or logarithmically spaced by using % in front of the code you don't want used.
   c. **Modify Section 4**
      i. Define parameters and parameter values (The order of these parameters must match the listed parameters in your model_simulator.m exactly.)
      ii. Edit params vector to match your parameters.
3. Run data_simulator.m
      i. The simulated data set is located in the MATLAB workspaces and is called "Sim_Data_Set"