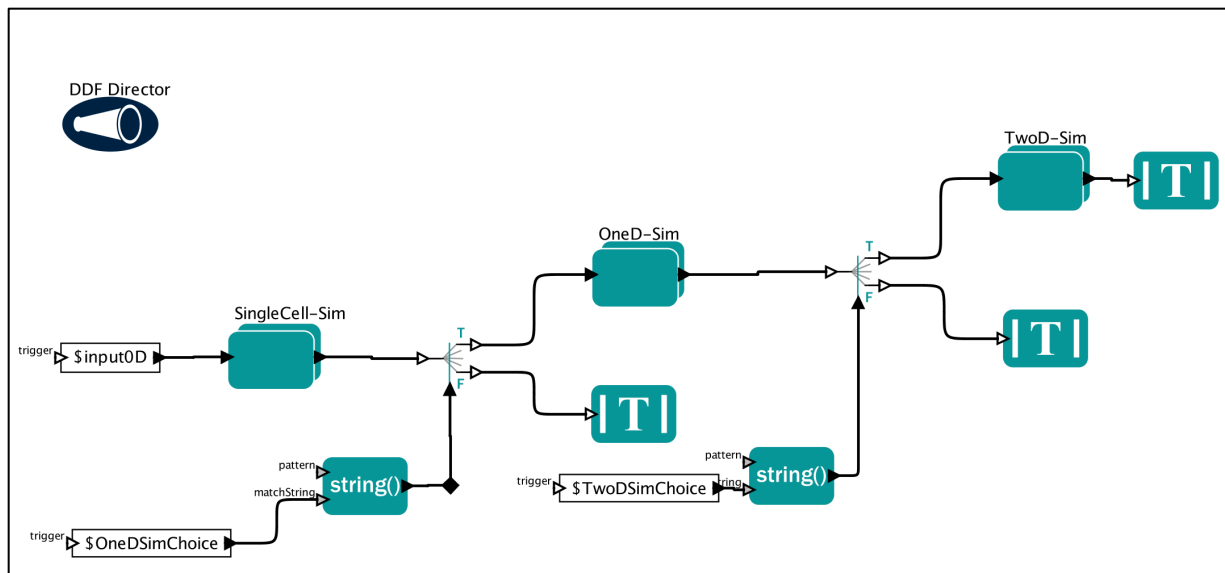


# User Manual

## Multiscale Cardiac cell Workflow



## Table of Contents

<b>CHAPTER 1: INTRODUCTION AND TOOL DESCRIPTION</b>	<b>3</b>
<b>CHAPTER 2: INSTALLATION AND PROGRAM REQUIREMENT</b>	<b>4</b>
INSTALLING KEPLER:	4
STARTING KEPLER:	4
INSTALLING BIOKEPLER:	5
<b>CHAPTER 3: DOWNLOADING THE MD WORKFLOW</b>	<b>8</b>
<b>CHAPTER 4: STEP-BY-STEP TUTORIAL</b>	<b>9</b>
1. 10	
2. 11	
3. 11	
Step 1.	10
Creating id_rsa.pub files:	10
Step 2.	11
Preparing for job submission in a remote cluster:	11
Step 3.	12
Running MD Workflow on a GPU cluster	12
4. 14	
Initial check	13
<b>CHAPTER 5: FOR ADVANCED USERS</b>	<b>14</b>
1. 15	
Step 1.	14
Close the Kepler GUI, if open	14
Step 2.	14
Edit DB Host parameter in configuration. file	14
Step 3.	14
Start HSQL	14
Step 4.	14
Run multiple workflows through the command line	14
Step 5.	15
Stop HSQL when you are done	15

## *Chapter 1: Introduction and Tool Description*

The purpose of the workflow

This workflow simulates a single cardiac cell, a string of cardiac cells in one dimension and a column of cardiac cells in two dimensions. The signal propagates from a single cell and affects the entirety of the cardiac muscles.

## Chapter 2: Installation and Program Requirement

The MD workflow requires users to have Kepler, bioKepler 1.2, a compiler and all necessary input scripts installed on their local machines (2-4). Users can find more information about Kepler (<https://kepler-project.org/>) and bioKepler 1.2 at (<https://www.biokepler.org/install-biokepler-1.2>). Intel C++ compiler is used when developing this workflow and the conversion of source codes. For more information, please check out Intel developer website (<https://software.intel.com/en-us/intel-compilers>).

### Installing MatLab:

Users can download the latest MatLab version from the following link:  
<https://www.mathworks.com/products/matlab.html>

To install on **Windows**, double click on the executable and follow the installer directions.  
To install on **Mac OS X**, mount the dmg and then running it. Then follow the instruction in the installer

To install on **Linux**, unzip the files anywhere you want and then open the terminal and type the command below: `sudo sh install`.

### Installing Kepler:

Users can download the latest Kepler software from the following link:  
<https://kepler-project.org/users/downloads>

To install on **Windows**, double click on the executable Kepler-2.5-win.exe.

To install on **Mac OS X**, mount the Kepler-2.5.dmg by double-clicking on it and dragging-and-dropping the Kepler-2.5 folder into the /Applications folder.

To install on **Linux**, simply untar Kepler-2.5-linux.tar.gz at a location of your choice, which will generate a folder named Kepler-2.5. This is the command line command to untar a tar.gz file:

```
tar -C /[myfolder] -zxvf Kepler-2.5-linux.tar.gz
```

For the following tutorial, we will work on a Linux platform as an example. Windows and Mac OS X installations follow similar steps.

### Starting Kepler:

Move into the Kepler installation directory.

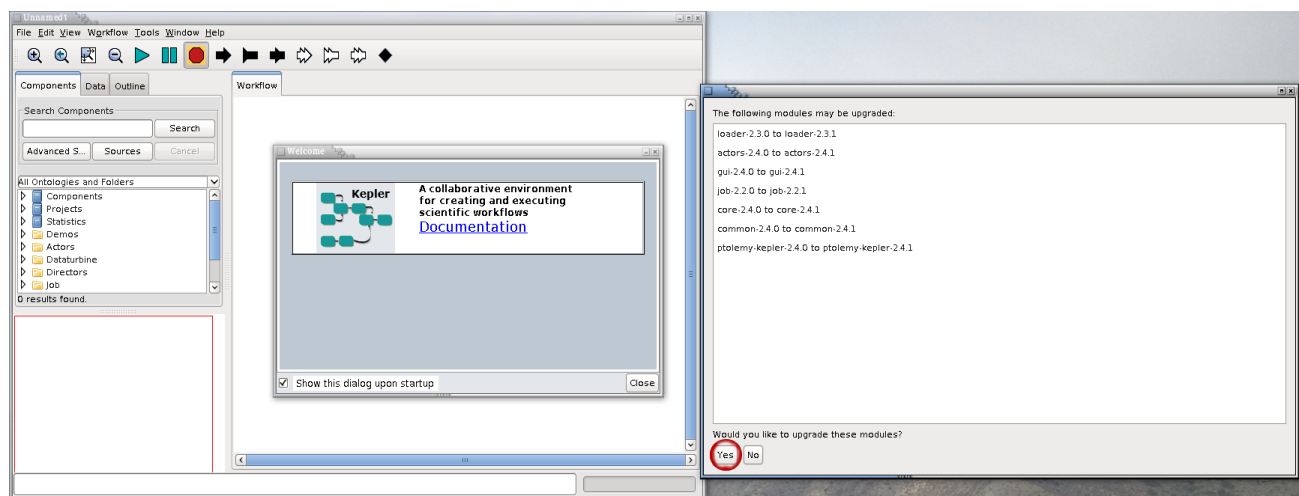
```
cd /[myfolder]/Kepler-2.5
```

To start the Kepler GUI, execute the kepler.sh script.

```
kepler.sh
```

A pop up window may prompt for first time users to upgrade several modules, click Yes (red circle in Fig. 1), and Kepler will automatically upgrade and restart.

**Fig. 1**

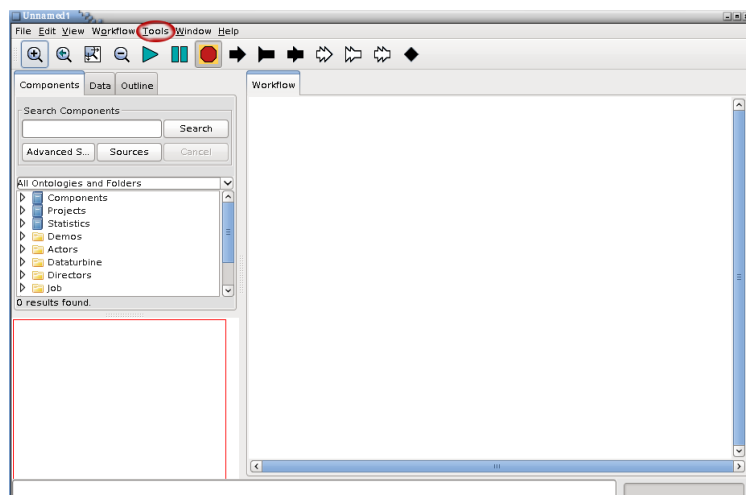


### Installing bioKepler:

Our workflow allows users to choose between running the MD simulations with either a GPU cluster or a local GPU card. To allow this flexibility in a single workflow, we utilize a module from bioKepler. Hence, besides installing the Kepler software, we need to also install the bioKepler module.

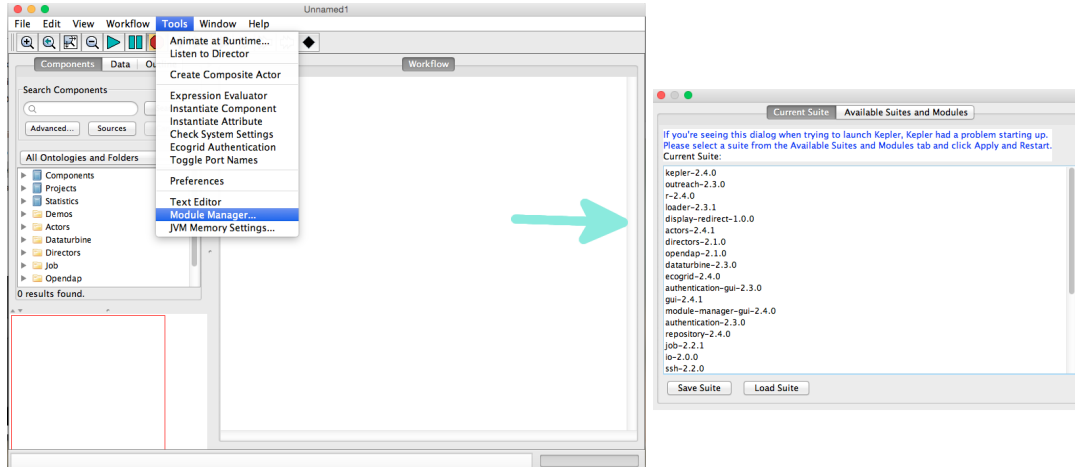
The Kepler GUI will reopen after the modules are successfully updated (Fig. 2). Go to the menu bar on the top left of the screen and select Tools (Fig. 2, red circle).

**Fig. 2**



From the drop down menu under Tools, select Module Manager and a new window will pop-up (Fig. 3).

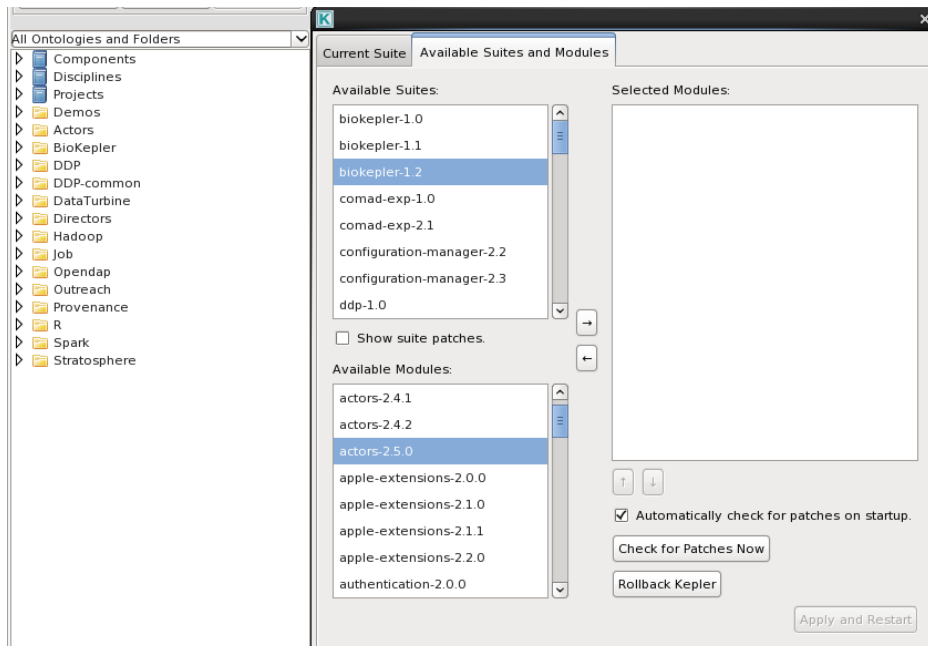
**Fig. 3**



There are two tabs at the top of the new window. Click on Available Suites and Modules, which will display a list of available suites and modules. Among the Available Suites, select bioKepler 1.2, and then click on the right arrow to transfer bioKepler 1.2 to the Selected Modules window (Fig. 4).

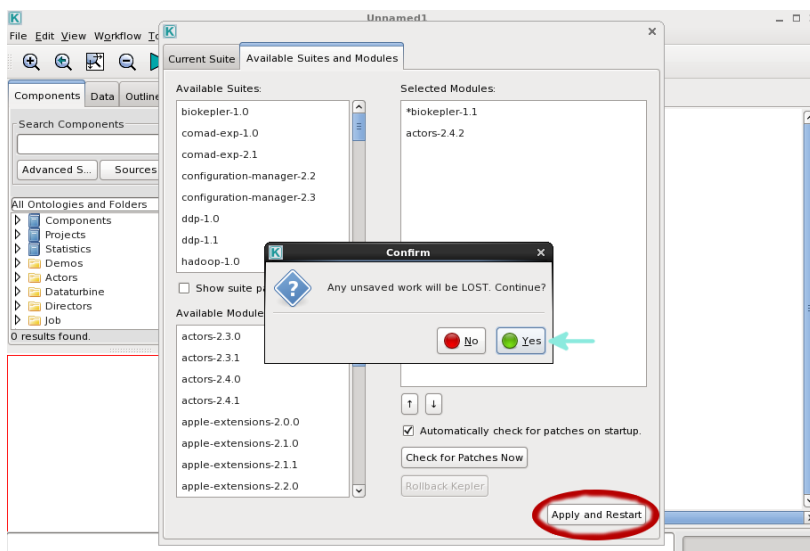
Do the same thing for the actors-2.4.2 modules. Among the Available Modules, select actors-2.5.0, and then click on the right arrow to transfer actors-2.5.0 to the Selected Modules window (Fig. 4).

**Fig. 4**



Finally, apply the selected modules, and restart Kepler by clicking on the Apply and Restart tab at the bottom right corner of the window (Fig. 5). Kepler will warn users that any unsaved work will be disregarded. Please save any unsaved work, and click Yes (Fig. 5).

**Fig. 5**



The Kepler GUI will reopen after successfully installing bioKepler. You can close it because we will run the MD workflow using command prompt.



### *Chapter 3: Downloading the MD workflow*

The MD workflow is prepared to run on the Kepler platform. The workflow itself is a product developed by NBCR (<http://nbcrc.ucsd.edu>) and WorDS (<http://words.sdsc.edu/>).

The MD workflow, this manual, and sample inputs can be downloaded from github or from the following link:

[INSERT LINK]

## Chapter 4: Step-by-Step Tutorial

### 1. Setting up input folders:

Before running the workflow, you need to set up and organize input folders in a specific required format. Inside the input directory, there are two required sub-directories: DBP-Clancy-rabbit-1D2D and DBP-Clancy-Rabbit-Single-Cell that contain source code, dependent code files and input file. Users can also find the sample input folder here ([LINK HERE](#)).

DBP-Clancy-rabbit-1D2D:	DBP-Clancy-Rabbit-Single-Cell:
ECG_1D.m	stim_param.txt
README.docx	README.md
README.md	README.txt
ap_surface_1D.m	SS_rabbit_varNames.txt
initial_WTstates.txt	initial_WTstates.txt
integrate_rk2.h	integrate_rk2.h
masterCompute.cpp	masterCompute.cpp
masterCompute_1D.h	plotall.m
masterCompute_2D.h	soltis_biophysJ2010_BARsignalling_odefile.h
soltis_biophysJ2010_BARsignalling_odefile.h	soltis_biophysJ2010_CaMKII_signaling_ODEfile.h
soltis_biophysJ2010_CaMKII_signaling_ODEfile.h	soltis_biophysJ2010_camODEfile.h
soltis_biophysJ2010_camODEfile.h	soltis_biophysJ2010_eccODEfile.h
soltis_biophysJ2010_eccODEfile.h	soltis_biophysJ2010_eccODEfile_old.h
soltis_biophysJ2010_masterODEfile.h	soltis_biophysJ2010_masterODEfile.h
spiralwave_mov_2D.m	
stim_param.txt	

Download the files and create a new directory.

```
mkdir Workflow_Sample
```

Then, move the zip file into the new directory.

```
mv ~/Downloads/Input_Files.zip Workflow_Sample
```

Change directory into the MD\_TEST folder, unzip the Input\_Files.zip file and then remove it from the directory.

```
cd Workflow_Sample  
unzip Input_Files.zip  
rm Input_Files.zip
```

Two folders named DBP-Clancy-rabbit-1D2D and DBP-Clancy-Rabbit-Single-Cell will now appear in the empty directory.

Now, you are ready to run the workflow. There are three modes of execution and we will go over each of them in the next section: **local execution**, execution on the **XSEDE Comet GPU cluster**, and execution on a **private GPU cluster**.

## 2. Running the workflow using the Local Execution mode:

In your local computer, go to the directory where Kepler was installed and where kepler.sh is located.

```
cd /PATH/kepler-2.5
```

And type the following command line in that directory to start the MD workflow. Users will need to modify the command based on their needs. The definition for each parameter is listed below. An **ABSOLUTE** path is need when running Kepler. If an absolute path is not specifically typed out in command prompt, Kepler won't be able to find the files or folders of interest and it will throw an error message.

The general command to run Kepler through command prompt is

```
./kepler.sh -runwf -nogui -parameters values /PATH/workflow.xml
```

Here is an example of how Kepler can be run for our test case:

```
./kepler.sh -runwf -nogui -ExecutionChoice LocalExecution -input0D  
/PATH/DBP-Clancy-Rabbit-Single-Cell -input1D2D /PATH/DBP-Clancy-rabbit-  
1D2D /PATH/Workflow.xml
```

For more information about each parameter, please check the parameters (see section below).

### Parameters:

ExecutionChoice: This parameter gives option to run same workflow on a local machine or a remote Cluster. For local execution: LocalExecution

input0D: The inputFolder path for singlecell on your local machine. For example: /Path/to/DBP-Clancy-Rabbit-Single-Cell

input1D2D: The inputFolder path for 1D2D on your local machine. For example:  
/Path/to/DBP-Clancy-rabbit-1D2D

## 3. Running the workflow using the GPU Cluster Submission mode:

Step  
1.

Creating id\_rsa.pub files:

(Skip to step 3 if the user has already configured the SSH key.)

Most remote clusters require login authentications and passwords. In order to connect to the remote cluster through the command-line interface without having to type in password for each submission step, the easiest way is to generate an SSH key. To run Kepler from the

command prompt smoothly, we will create SSH keys for logging into a remote cluster without password.

First, create an .ssh directory in your home directory in your local computer. If users already have an .ssh directory in their home directory, they can skip this step.

```
mkdir ~/.ssh
```

Then, in the .ssh directory, generate the public/private RSA key pairs with the below command line:

```
cd ~/.ssh  
ssh-keygen -t rsa
```

The system will ask you the following questions:

```
Generating public/private rsa key pair.  
Enter file in which to save the key (/Users/you/.ssh/id_rsa): [Press enter]  
Enter passphrase (empty for no passphrase): [Press enter]  
Enter same passphrase again: [Press enter]  
Your identification has been saved in /Users/you/.ssh/id_rsa.  
Your public key has been saved in /Users/you/.ssh/id_rsa.pub.  
The key fingerprint is: # 01:0f:f4:3b:ca:85:d6:17:a1:7d:f0:68:9d:f0:a2:db
```

Step  
2.

Preparing for job submission in a remote cluster:

Log in to the remote cluster of your choice, for example, the NBCR Kivid Cluster.

```
ssh username@kivid.ucsd.edu
```

Type in the password. Go to the home directory and create an .ssh folder in the remote cluster. (The .ssh directory may already be there, which is fine; ignore this step).

```
cd  
mkdir -p ~/.ssh  
logout
```

Then, in your local computer, append the public key to your remote cluster.

```
cat ~/.ssh/id_rsa.pub | ssh username@kivid.ucsd.edu 'cat >> ~/.ssh/authorized_keys'
```

Now you should be able to log into the remote cluster from your local computer without needing to type your password.

Log in to the remote cluster and create a folder. This folder is the working directory in the remote cluster where users will run their simulation jobs

```
ssh username@kivid.ucsd.edu  
  
cd  
mkdir Workflow_Sample
```

You are ready to run the MD workflow.

Step  
3.

Running MD Workflow on a GPU cluster

1. *Running the workflow using NBCR Cluster:*

In your local computer, go to the directory where Kepler was installed and where kepler.sh is located.

```
cd /PATH/WHERE/KEPLER/IS/INSTALLED/kepler-2.5
```

And type the following command line in that directory to start the MD workflow. Users will need to modify the command based on their needs. The definition for each parameter is listed below. An **ABSOLUTE** path for each file is need when running Kepler. If an absolute path is not typed out, Kepler won't be able to find the files or folders of interest and throw an error message.

The general command to run Kepler through command prompt is

```
./kepler.sh -runwf -nogui -parameters values /PATH/workflow.
```

Here is an example of how Kepler can be run for our test case.

```
./kepler.sh -runwf -nogui -ExecutionChoice kividCluster -TargetHost  
username@kivid.ucsd.edu -IdentityFile /Users/username/.ssh/id_rsa -  
input0D /Path/to/DBP-Clancy-Rabbit-Single-Cell -input1D2D /Path/to/DBP-  
Clancy-rabbit-1D2D -remoteDir /data/username/Workflow_Sample  
/path/to/DBP-1D2D-WF.xml
```

Please find brief description of the parameters below. Please customize below parameters for your execution:

## Parameters:

ExecutionChoice – This parameter gives option to run same workflow on a local machine or a remote Cluster. For remote execution: kividCluster

TargetHost: Username@host. For example: username@kivid.ucsd.edu

IdentityFile: The identity File to make SSH and SCP password less. For example:  
/Users/username/.ssh/id\_rsa

input0D: The inputFolder path for singlecell on your local machine. For example: /Path/to/DBP-Clancy-Rabbit-Single-Cell

input1D2D: The inputFolder path for 1D2D on your local machine. For example:  
/Path/to/DBP-Clancy-rabbit-1D2D

remoteDir: Work Directory on the remote cluster. For example:  
/home/username/Workflow\_Sample

## 4. Checking the MD simulation output data:

### Initial check

The outputs contain an output directory for each set of simulations (DBP-Clancy-Rabbit-Single-Cell/outputFolder, DBP-Clancy-rabbit-1D2D/output1DFolder, and DBP-Clancy-rabbit-1D2D/output2DFolder). Inside each output folder, the following files can be found:

DBP-Clancy-Rabbit-Single-Cell/outputFolder	DBP-Clancy-rabbit-1D2D/output1DFolder
apds_1Hz.txt	ECG.png
vm_1Hz.txt	y_1D.txt
allresult_1Hz.txt	ECGs.txt
RyR.png	
I_NCX.png	
Na.png	<b>DBP-Clancy-rabbit-1D2D/output2DFolder</b>
IKr_IK1.png	movie_FBL.avi
IKs_ICFTR.png	ap*.dat files
INa.png	
Ito.png	
Cai.png	
CaSRT_Caj.png	
ICa.png	
vm.png	

## Chapter 5: For Advanced Users

1. Configuring Kepler for multiple workflow executions simultaneously through command line

### Step 1.

*Close the Kepler GUI, if open*

Go to KeplerData directory. It is usually located in the user's home directory

```
cd /home/username/KeplerData
```

or

```
cd ~/KeplerData
```

### Step 2.

*Edit DB Host parameter in configuration. file*

```
vi modules/provenance/resources/configurations/latest/configuration.
```

In configuration file: provide DB Host value as "localhost" and save changes.

```
<!-- host name running the database -->
<pair>
  <name>DB Host</name>
  <value>localhost</value>
</pair>
```

### Step 3.

*Start HSQL*

Go to the directory where Kepler is installed. Start HSQL.

```
cd /PATH/WHERE/KEPLER/IS/INSTALLED/kepler-2.5
```

```
kepler.sh -hsql start
```

### Step 4.

*Run multiple workflows through the command line*

```
./kepler.sh -runwf -nogui -parameters values /PATH/workflow1.  
./kepler.sh -runwf -nogui -parameters values /PATH/workflow2.  
./kepler.sh -runwf -nogui -parameters values /PATH/workflow3.  
./kepler.sh -runwf -nogui -parameters values /PATH/workflow4.
```

#### Step 5.

*Stop HSQL when you are done*

```
kepler.sh -hsql stop
```

**Any more??**

When running the workflow in GUI mode, there should be multiple display outputs. If you are only getting a few, there is something wrong during the runtime.

Local execution may ask you for the password for the kivid account or other remote cluster passwords. If you are still running on local execution, you can ignore the popup and it would still run perfectly

Before running the workflow, make sure the output folders are set in the desired locations or else it would be hard to find.

Common Errors and solutions:

Rebuilding the matlab files could be used to solve some problems involving matlab not running properly on kepler. There a guide on it here: <https://kepler-project.org/developers/teams/build/matlab-lib-build-instructions-in-kepler-for-matlab-actor/?searchterm=Matlab>.

Wrong Java version was used. Kepler supports mainly Java 8, you might need to switch your Java version for it to work.

When switching Java versions, you might need to recompile and build kepler again.

The default compiler is the intel C++ compiler. You will get a “not found” error if you do not have the compiler installed. You can find the compiler here: <https://software.intel.com/en-us/c-compilers>.

Not all the default parameters need to be changed for it to work for the first time. Some of them should not be changed until further analysis is needed. It is best to test if the system works before changing advanced parameters.

The output folder is located in the folder the workflow was ran under. For example, if you ran the single cell workflow, the output folder would be in the single cell workflow folder.