

User Manual

Multiscale Cardiac cell Workflow

Table of Contents

<i>Chapter 1: Introduction and Tool Description</i>	3
<i>Chapter 2: Installation and Program Requirement</i>	4
Installing bioKepler:	4
<i>Chapter 3: Downloading the Multiscale Cardiac Cell workflow</i>	5
<i>Chapter 4: Step-by-Step Tutorial</i>	6
1. Setting up input folders:	6
2. Running the Multiscale workflow through GUI Interface:	7
a. Local Execution mode:	8
b. Remote Execution mode:	8
3. Running the Multiscale workflow through Commandline interface:	8
a. Local Execution mode:	8
b. Remote Cluster Submission mode:	9
Step 1. Creating id_rsa.pub files:	9
Step 2. Preparing for job submission in a remote cluster:	10
Step 3. Running Multiscale Workflow on a Remote cluster	11
4. Checking the Multiscale Cardiac simulation output data:	11
Initial check	11
<i>Chapter 5: For Advanced Users</i>	13
1. Configuring Kepler for multiple workflow executions through command line	13
Step 1. Close the Kepler GUI, if open	13
Step 2. Edit DB Host parameter in configuration. file	13
Step 3. Start HSQL	13
Step 4. Run multiple workflows through the command line	13
Step 5. Stop HSQL when you are done	14

Chapter 1: Introduction and Tool Description

We have developed a Kepler based workflow for multiscale cardiac electrophysiology. Figure 1 exhibits the automated computational workflow that can perform simulations to generate user defined instances and configurations of a single-cell cardiac action potential, conduction of a cardiac action potential in a 1-dimensional (1D) or 2-dimensional (2D) tissue representation and generation of a signal average of electrical activity in time and space to represent a pseudo-ECG.

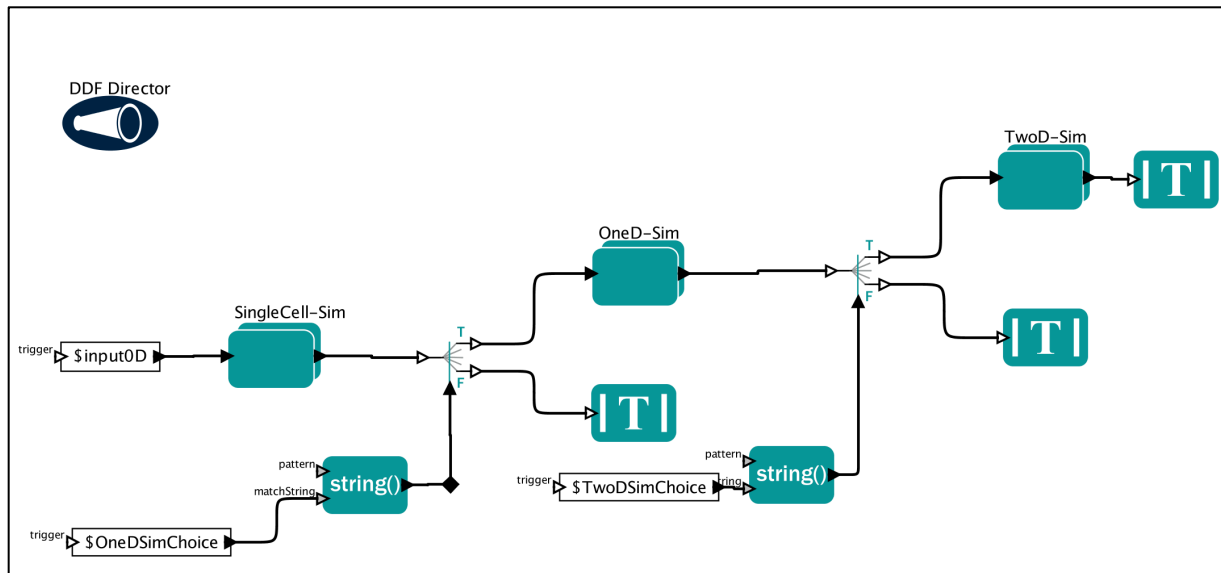


Figure 1: Multiscale Cardiac Cell Workflow

The approach enables modularity, reuse, reproducibility, portability and scalability. The workflow concept also allows a model execution platform that allows the reuse of code, reproduction of reported in silico predictions, as well as a way to run simulations in an efficient, expandable, modular and portable manner.

Chapter 2: Installation and Program Requirement

The Multiscale Cardiac Cell workflow requires users to have Java 8 or newer, Kepler-2.5, bioKepler-1.2, Python, and necessary input scripts installed on their local machines (2-4).

bioKepler – bioKepler is a suite of add-on modules that includes Kepler. The bioKepler suite adds Provenance module, Reporting module, the Data Distributed Pattern (DDP) module and bioActors to the Kepler suite. Users can find more information about Kepler (<https://kepler-project.org/>) and bioKepler (<https://www.biokepler.org>).

C++ compiler - Coding Language for the cardiac models is C++. For more information, please check out Intel developer website (<https://software.intel.com/en-us/intel-compilers>). The Multiscale Cardiac workflow compiles the models in real-time on local or a remote cluster depending on user execution choice. Users can compile using the source code using either ICC or GCC compiler.

Installing bioKepler:

Users can download the latest bioKepler software from the following link:

<https://www.biokepler.org/bioKepler-Dec-2018-Build>

- **Linux** - Simply untar bioKepler-Dec2018-build.tar.gz at a location of your choice, which will generate a folder named bioKepler-Dec2018-build. This is the command line command to untar a tar.gz file:

```
tar -zxvf bioKepler-Dec2018-build.tar.gz
```

- **Mac OS X** - mount the bioKepler-Dec2018-build.dmg by double-clicking on it and dragging-and-dropping the bioKepler-Dec2018-build folder into the /Applications folder.
- **Windows**
 - Select the Windows installer.
 - Save the install file to your computer.
 - Double-click the install file to open the install wizard.
 - Follow the steps presented to complete the Kepler installation process. Once the installation process is complete, a Kepler shortcut icon will appear on your desktop and/or in the Start Menu.

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

Chapter 3: Downloading the Multiscale Cardiac Cell workflow

The Multiscale Cardiac 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 Multiscale Cardiac Cell workflow, this manual, and sample inputs can be downloaded from github.

Multiscale workflow Github Repository - https://github.com/ClancyLabUCD/Workflow_Kepler

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: “Single_Cell” and “1D2D” that contain source code, dependent code files and input file. Users can also find the sample input folder here ([LINK HERE - Rabbit_OD1D2DWF](#)).

1D2D:	Single_Cell:
ECG_1D.py	stim_param.txt
README.docx	README.md
README.md	README.txt
snapshot2D.py	SS_rabbit_varNames.txt
initial_WTstates.txt	initial_WTstates.txt
integrate_rk2.h	integrate_rk2.h
masterCompute.cpp	masterCompute.cpp
masterCompute_1D.h	plotall.py
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	MSWF-KeyValuePairParser.py
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/Workflow_Kepler-master.zip Workflow_Sample
```

Unzip the Workflow_Kepler-master.zip file and then remove it from the directory.

```
cd Workflow_Sample  
unzip Workflow_Kepler-master.zip  
rm Workflow_Kepler-master.zip
```

One workflow XML file “DBP-1D2D-WF.xml” and two Input folders named, 1D2D and Single-Cell, will now appear in the empty directory.

Now, you are ready to run the workflow. Users can execute workflow through GUI or commandline interface. There are two modes of execution and we will go over each of them in the next section: **local execution**, and execution on a **Remote cluster**.


2. Running the Multiscale workflow through GUI Interface:

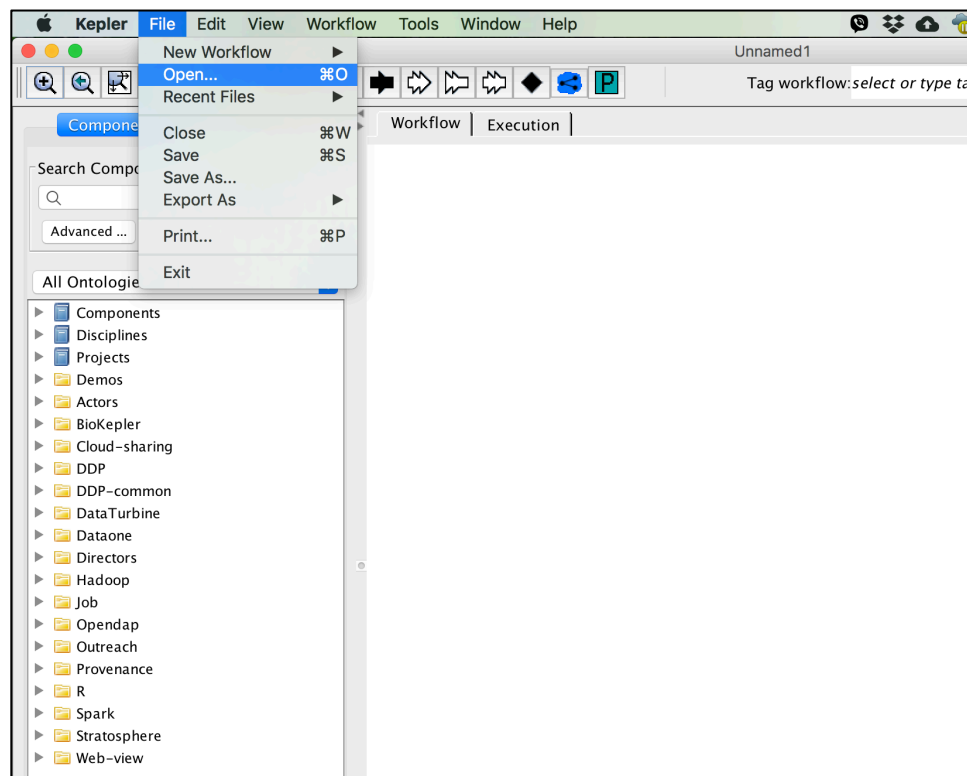
In your local computer, go to the directory where Kepler is installed and where kepler.sh is located.

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

The general command to start Kepler through command prompt is

```
./kepler.sh
```

- In Menu bar, go to “File” tab and select “Open” option and open the desired workflow file (DBP-1D2D-WF.xml).
- Customize workflow parameters, for example, ExecutionChoice, input folder paths, compilerProgram and other Shared Parameters and Execution Choice specific parameters.
- From the Toolbar, select the Run button. 
- The workflow will execute and produce the specified output files.



a. Local Execution mode through GUI:

Parameters: Please find brief description of the parameters specific to local execution below. Please customize below parameters for your execution:

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/Single-Cell

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

b. Remote Execution mode through GUI:

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/Single-Cell

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

remoteDir: Work Directory on the remote cluster. For example:
/home/username/Workflow_Sample

3. Running the Multiscale workflow through Commandline interface:

a. 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 Multiscale 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/Single_Cell -input1D2D /PATH/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/Single-Cell

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

b. Remote 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 Multiscale Workflow on a Remote cluster

1. Running the workflow using NBCR Kivid 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/Single-Cell -input1D2D /Path/to/1D2D -remoteDir  
/data/username/Workflow_Sample /path/to/DBP-1D2D-WF.xml
```

4. Checking the Multiscale Cardiac simulation output data:

Initial check

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

Single-Cell/outputFolder	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	1D2D/output2DFolder
IKr_IK1.png	

IKs_ICFTR.png	ap*.dat files
INa.png	apt*.png
Ito.png	
Cai.png	
CaSRT_Caj.png	
ICa.png	
vm.png	

Chapter 5: For Advanced Users

1. Configuring Kepler for multiple workflow executions 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
```