**Slides, videos, links and more:**
https://github.com/physicell-training/ws2023

# Setting up PhysiCell on Linux

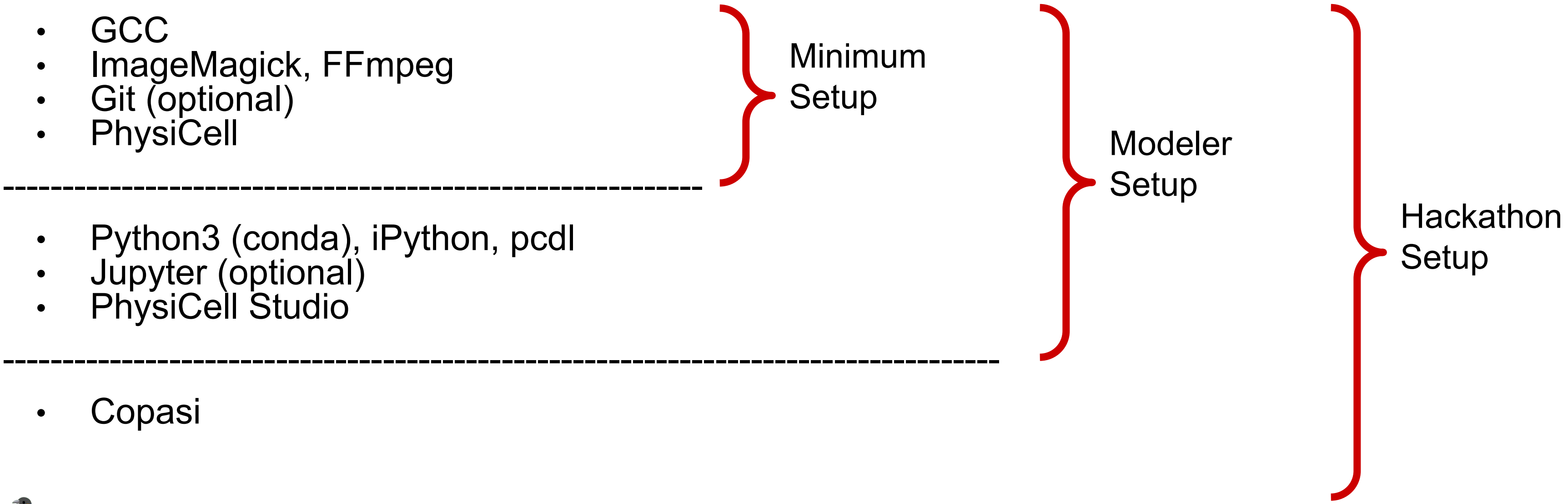
Elmar Bucher
@MathCancer

## PhysiCell Project

May 2024

# Overview

This document describes the PhysiCell installation on a Debian Linux distribution. If you run on another flavor, please adjust accordingly.



- GCC
- ImageMagick, FFmpeg
- Git (optional)
- PhysiCell

------------------------------------------------------------

- Python3 (conda), iPython, pcdl
- Jupyter (optional)
- PhysiCell Studio

------------------------------------------------------------------------------------

- Copasi

Minimum Setup

Modeler Setup

Hackathon Setup

# GCC (Minimum Setup)

- Check if GCC (the gnu compiler collection) is already installed.

```
g++ --version
```

- If not:

```
sudo apt install build-essential
```

- OpenMP (open multi processing) is a feature, used by PhysiCell, that is implemented in GCC.

# ImageMagick (Minimum Setup)

ImageMagick is used for making jpeg and gif images from PhysiCell svg image output.

- Check if ImageMagick is already installed.
  `magick --version`
  - If ok: **you have >= version 7 installed**. You are all set! You can move to the next page.
  - If you receive: `Command 'magick' not found`, try:
    `convert --version`
    if ok: **you have most probably a version 6 installed** (like the most of us)!

- If both of these commands not worked:
  `sudo apt install imagemagick`

**The PhysiCell `Makefile` is written for ImageMagick >= version 7**, which requires a `magick` command in front of each ImageMagick command (e.g. `magick convert` instead of `convert`).

**Iff you have <= version 6 installed**, you can follow the instruction below to generate a `magick` command that simply passes everything to the next command. This will make the PhysiCell Makefiles work for you too.

`cd` to the folder where you have your manual installed binaries. e.g. `~/.local/bin/`

```
echo '$*' > magick
chmod 775
which magick
```

# FFmpeg (Minimum Setup)

FFmpeg is used for making mp4 movies out of jpeg images.

- Check if FFmpeg is already installed.
  ```
  ffmpeg -version
  ```

- If not:
  ```
  sudo apt install ffmpeg
  ```

# Git (optional, Minimum Setup)

`git` is an alternative way to install the PhysiCell and the Studio source code. The common way is to use `wget`.

- Check if git is already installed

  ```
  git --version
  ```

- If not:

  ```
  sudo apt install git
  ```

# wget & unzip

Check if you have `wget` and `unzip` installed.

-
  ```
  wget --version
  ```
  if not:
  ```
  sudo apt install wget
  ```

-
  ```
  unzip
  ```
  if not:
  ```
  sudo apt install unzip
  ```

# PhysiCell (Minimum Setup)

Download PhysiCell and place it where you want to work with this source code.

- Git:
  ```
  git clone git@github.com:MathCancer/PhysiCell.git
  ```

- Wget:
  ```
  wget
  https://github.com/MathCancer/PhysiCell/releases/download/1.13.1/PhysiCell_V.1.13.1.zip
  unzip PhysiCell_V.n.n.n.zip
  ```

Test the installation with the biorobots sample project.

```
cd PhysiCell
make biorobots-sample
make
./biorobots
make jpeg
make movie
```

# Overview

This document describes the PhysiCell installation on a Debian Linux distribution. If you run on another flavor, please adjust accordingly.

- GCC
- ImageMagick, FFmpeg
- Git (optional)
- PhysiCell

Minimum Setup

-----------------------------------------------------------

- Python3 (miniconda), iPython, PhysiCell DataLoader
- Jupyter (optional)
- PhysiCell Studio

Modeler Setup

-----------------------------------------------------------------------------------

- Copasi

Hackathon Setup

🐧🐧

# Python3 part I (Modeler Setup)

It is up to you to use the python3 that ships with the distribution. The **mamba** package manager (faster and robuster than conda, otherwise the same and totally compatible with pip and conda) comes in handy, when you want to run somewhere python3 where you have no root rights.
https://mamba.readthedocs.io/en/latest/user_guide/mamba.html

- `cd` to the folder where you have your manual installed programs.
  e.g. `~/.local/lib/`

- Adjust the download according to your CPU architecture (here x86_64)!
  https://github.com/conda-forge/miniforge

-
  ```
  wget https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-Linux-x86_64.sh
  chmod 775 Miniforge3-Linux-x86_64.sh
  ./Miniforge3-Linux-x86_64.sh
  ```

- Adjust the installation location to where your manual installed programs are.
  e.g. `home/<user>/.local/lib/miniforge3`

- You wish the installer to initialize mamba!
  `yes`
  `source ~/.bashrc` (or log out and in again).

# Python3 part II (Modeler Setup)

- With **mamba** you can now generate virtual python environment, in addition to the (base) environment, with the command below. Beware: You have at least to list one package (e.g. ipython) to successfully generate an environment!

  ```
  mamba create -n nameofmyenv ipython
  ```

- You can list all existing virtual environments like this:

  ```
  mamba env list
  ```

- You can activate a virtual environment like this:

  ```
  mamba activate nameofmyenv
  ```

- To escape from all virtual environments, e.g. to work with the python that ships with the operating system, like this:

  ```
  mamba deactivate
  ```

# Python3 part III (Modeler Setup)

- If you have virtual environments, the environment you want to use for PhysiCell related work should now be activated.

- Check if the python and pip paths point to the installed location:

```
which python3
which pip3
```

- Install the iPython shell:

```
pip3 install ipython
```

- Install the PhysiCell DataLoader:

```
pip3 install pcdl
```

- Optional: install Jupyter:

```
pip3 install jupyterlab
```

# PhysiCell Studio (Modeler Setup)

Download the studio and place it where you want to work with this source code.

- Git:

  ```
  git clone git@github.com:PhysiCell-Tools/PhysiCell-Studio.git
  ```

- Wget:

  ```
  wget https://github.com/PhysiCell-Tools/PhysiCell-Studio/archive/refs/tags/v2.37.9.zip
  unzip vn.n.n.zip
  ```

Put the studio under the PATH:

`cd` to the folder where you have your manual installed binaries. e.g. `~/.local/bin/`

```
echo 'python /absolute/path/to/PhysiCell-Studio/bin/studio.py $*' > studio
chmod 775
which studio
```

Install the Qt library dependencies:

```
sudo apt install qtbase5-dev
pip3 install PyQt5
```

Test the installation with the biorobots sample project.

`cd` to wherever you have PhysiCell installed. Inside the PhysiCell folder:

```
studio
```

PhysiCell Studio should open, loaded with the biorobots settings file.

# Overview

This document describes the PhysiCell installation on a Debian Linux distribution. If you run on another flavor, please adjust accordingly.

- GCC
- ImageMagick, FFmpeg
- Git (optional)
- PhysiCell

Minimum Setup

------------------------------------------------------------

- Python3 (miniconda), iPython, PhysiCell DataLoader
- Jupyter (optional)
- PhysiCell Studio

Modeler Setup

------------------------------------------------------------------------------------

- Copasi

Hackathon Setup

# Copasi (Hackathon Setup)

- The detailed instructions can be found here: https://copasi.org/Support/Installation/Linux/

- `cd` to the folder where you have your manual installed programs. e.g. `~/.local/lib/`

- download copasi:

  ```
  wget
  https://github.com/copasi/COPASI/releases/download/Build-288/COPASI-4.43.288-Linux-64bit.sh
  ```

- Install copasi:

  ```
  chmod 775 COPASI-n.n.n-Linux-64bit.sh
  ./COPASI-n.n.n-Linux-64bit.sh
  ```

  Thereby change the installation directory setting to where you right now are.

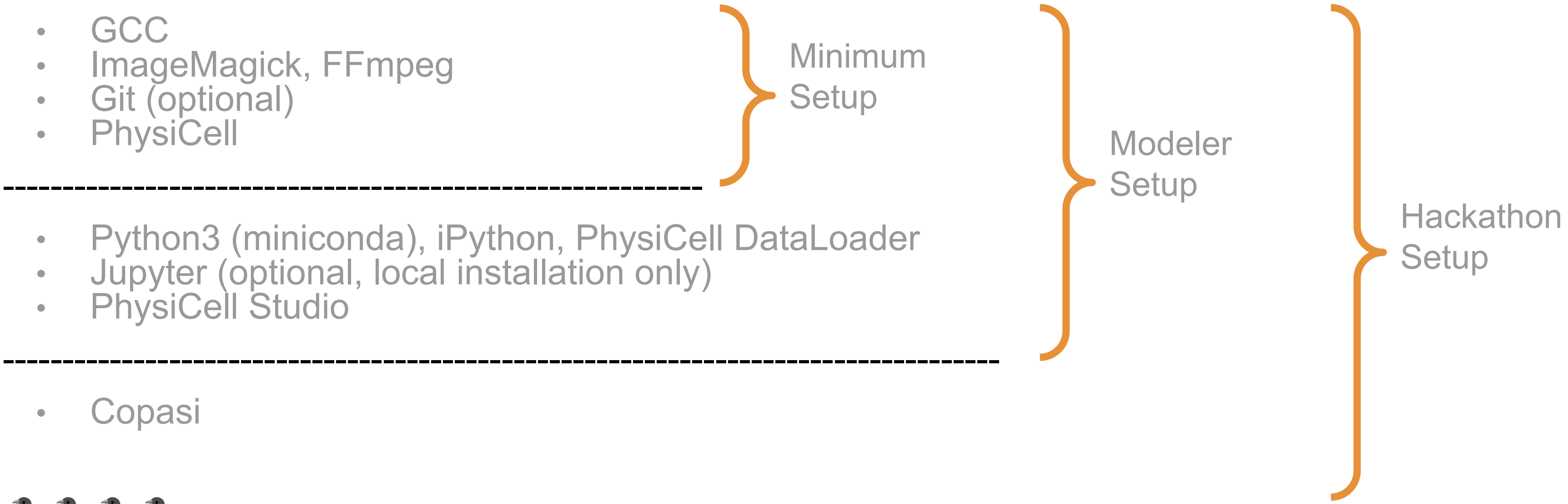
- Put copasi under the PATH

  `cd` to the folder where you have your manual installed binaries. e.g. `~/.local/bin/`

  ```
  ln -s /absolute/path/to/installed/CopasiSE
  ln -s /absolute/path/to/installed/CopasiUI
  ```

# Overview

This document describes the PhysiCell installation on a Debian Linux distribution. If you run on another flavor, please adjust accordingly.



- GCC
- ImageMagick, FFmpeg
- Git (optional)
- PhysiCell

Minimum Setup

-------------------------------------------------------------

- Python3 (miniconda), iPython, PhysiCell DataLoader
- Jupyter (optional, local installation only)
- PhysiCell Studio

Modeler Setup

---------------------------------------------------------------------------------------

- Copasi

Hackathon Setup

# Funding Acknowledgements

## PhysiCell Development:

- Breast Cancer Research Foundation
- Jayne Koskinas Ted Giovanis Foundation for Health and Policy
- National Cancer Institute (U01CA232137)
- National Science Foundation (1720625, 1818187)

## Training materials:

- Administrative supplement to NCI U01CA232137 (Year 2)

## Other Funding:

- NCI / DOE / Frederick National Lab for Cancer Research (21X126F)
- DOD / Defense Threat Reduction Agency (HDTRA12110015)
- NIH Common Fund (3OT2OD026671-01S4)