ISMB 2024 Tutorial IP2 (Friday, July 12, 2024 9:00 – 18:00 EDT, Room: 524c)

"Just-in-time compiled Python for bioinformatics research"
Know Before You Go (KBYG) information

by Johanna Schmitz, Jens Zentgraf & Sven Rahmann (Saarland University)

A. Preparations

We ask you to come with a prepared laptop to spend as little time as possible on setup in the morning. The preparation steps include:

1. forking and/or cloning our git repository (or, if you prefer, downloading a snapshot)
2. installing a Python distribution (recommended: miniforge)
3. creating a working environment (downloading and installing required packages)
4. downloading the t2t reference genome into the working directory using a script

In the following, we provide detailed instructions on each step.

A.1. Please clone (or otherwise obtain the contents of) our git repository that contains all materials. You can also fork it first if you want to commit and push your changes.

```
git clone https://gitlab.com/rahmannlab/numba-tutorial.git
```

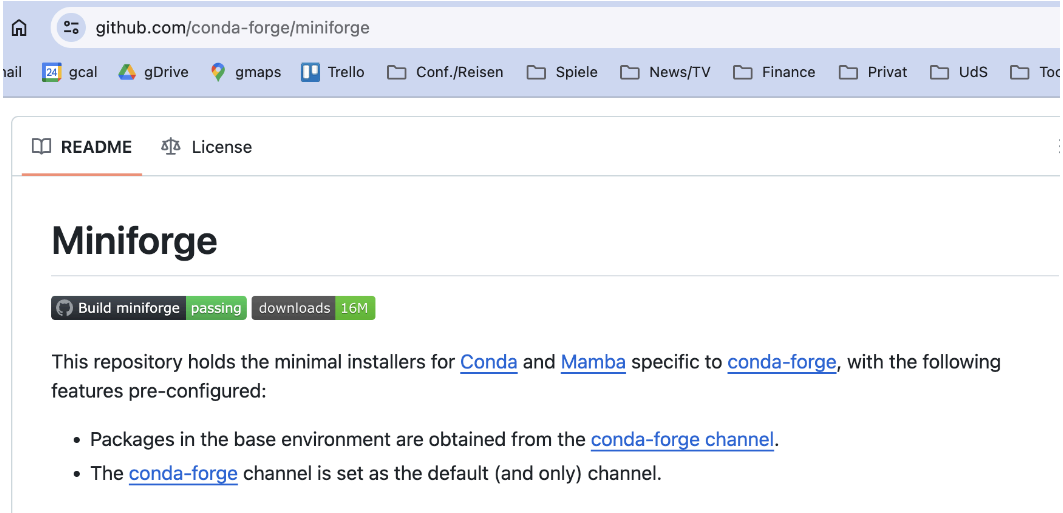
If you are not a user of git, alternatively download all files, maintaining the directory structure, from `https://gitlab.com/rahmannlab/numba-tutorial` (public).

A.2. To work through the tutorial, you need a recent Python version and several additional Python packages (they are also listed in the `environment.yml` file in the main directory of the repository)

We recommend that you use the **miniforge** Python distribution which uses the **conda** and **mamba** package managers and allows you to set up separate environments for different projects. It also has the advantage that it works purely from user directories, i.e. you do not need admin access to the computer to install anything. Everything works from your home directory.

We give some guidance for installing **miniforge** and using **mamba** here. If you use a different Python distribution, you have to install the required packages using your package manager. We recommend that you do not modify your system Python installation.

Miniforge can be found at www.github.com/conda-forge/miniforge.
You will find downloads for each major operating system and installation instructions (which mostly consist of running a shell script). You should allow the installer to modify your PATH. In the end, you should open a fresh terminal to execute the next step.



Miniforge

This repository holds the minimal installers for Conda and Mamba specific to conda-forge, with the following features pre-configured:

- Packages in the base environment are obtained from the conda-forge channel.
- The conda-forge channel is set as the default (and only) channel.

Figure: the miniforge page

A.3. Create a dedicated environment for the tutorial by going to the directory of the cloned git repository. There, you will find a file called `environment.yml`. It specifies which Python version and which packages should be installed for this tutorial, and it also specifies the name for the environment as "`numbatutorial`". When miniforge was installed correctly in step A.2, you can now just run the following command, and the environment will be created automatically.

```
mamba env create
```

This will download all required packages and create the environment, which then needs to be activated:

```
mamba activate numbatutorial   or   conda activate numbatutorial
```

Within the activated environment, you should run python and check that it is version 3.12; you should also try to import some of the modules.

```
python
```

you will see something like:

```
Python 3.12.0 | packaged by conda-forge | ...
```

Now try to see which package versions are installed:

```
import numpy
import numba
numpy.__version__, numba.__version__
```

This should print `('2.0.0', '0.60.0')`, or something close.

Similarly, please check that streamlit has been correctly installed: Quit Python and, back in the terminal, run

```
streamlit --version
```

This should show `Streamlit, version 1.36.0`.
For a more functional test, please run

```
streamlit hello
```

This will open a demo app in your browser (locally). To quit, close your browser window and hit CTRL-C in the terminal.
If everything works, your environment is all set up now.

A.4. Again in the top-level directory of the cloned repository, you will find a shell script to download the t2t reference genome. Please execute it:

```
./download_t2t.sh
```

This should work if your system supports either `wget` or `curl`. If this fails, please look at the URL in this file and download the genome (`chm13v2.0.fa.gz`) manually. The format is gzipped FASTA. As this is almost 1 GB in size, please do so while on a good fast WLAN before you travel.

B. Who will be giving the tutorial?

Johanna Schmitz has a Master in Bioinformatics from Saarland University (Saarbrücken, Germany) and is currently a PhD student in the Algorithmic Bioinformatics group at the same university.
Jens Zentgraf has a Master in Computer Science from TU Dortmund University and is currently a PhD student in the Algorithmic Bioinformatics group at Saarland University.
Sven Rahmann is professor of Algorithmic Bioinformatics at Saarland University and previously held positions as professor of Genome Informatics at University Hospital Essen, University of Duisburg-Essen, associate professor of Bioinformatics at TU Dortmund University and group leader at Bielefeld University (all in Germany).

The techniques presented in the tutorial are being actively developed and used by the group to create new methods and tools in sequence analysis, such as the fastest existing gapped *k*-mer counter (hackgap [1]: XXX) or xenograft sorting tool (xengsort [1]: XXX).
For more information, please visit our group website: https://www.rahmannlab.de

[1] Jens Zentgraf and Sven Rahmann. Fast gapped k-mer counting with subdivided multi-way bucketed cuckoo hash tables. 22nd International Workshop on Algorithms in Bioinformatics, WABI 2022, September 5-7, 2022, Potsdam, Germany, volume 242 of LIPIcs, pages 12:1–12:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022

[2] Jens Zentgraf and Sven Rahmann. Fast lightweight accurate xenograft sorting. Algorithms Mol. Biol., 16(1):2, 2021