2022 Sagan Summer Workshop

Exoplanet Science in the Gaia Era

Getting Started Guide: Python, Orvara, and Gaia

For the workshop hands-on sessions, participants can choose to run the exercises from their own installation of Python or use Google Colaboratory notebooks. These instructions are for the Python notebooks. In addition, participants should install the orvara package (see page 3) and create a Gaia Archive user account (see page 6).

Please note that *you should set up your Python distribution in advance of the workshop*; just follow the instructions below. In this document, we use the `Courier New` font to indicate a command you type into the command prompt.

The Python Distribution

In order to run the Jupyter notebooks used in the hands-on session, you first need a working installation of Python3. Even if your operating system comes with a system level Python distribution installed (e.g. most flavors of Linux include this), we recommend you follow these instructions to install the standalone Anaconda Python distribution. System level distributions are often far out of date, and are difficult to upgrade without breaking the entire operating system.

You can download the free Anaconda here: https://www.anaconda.com/products/distribution

Versions exist for Windows, Mac, and Linux. *Choose the Python 3.9 download appropriate for your operating system.* Note that the Individual Edition is free, while the Team and Enterprise distributions require license agreements. For the Sagan Workshop, and most data analysis you would do for independent research, the Individual Edition is sufficient.

These downloads are between 300 and 600 MB, depending on operating system, and need a minimum 5 GB of disk space for the download and installation. If your internet is limited in bandwidth, you can utilize the Miniconda distribution, which is much smaller (~50 MB; requires minimum 400 MB of disk space) at https://docs.conda.io/en/latest/miniconda.html. This gives you the same core packages as Anaconda, but strips out many extra packages that may not be needed. If you do need missing components later on, it is straightforward to add them individually (see page 4 for a list of the main packages you would have to install).

Note: If you already have Python installed on your system, but would like to install the new packages in their own virtual environment, you can choose to install Python in a custom working environment (see the instructions on pages 3-4).

Windows 🪟 MacOS 🍎

The Windows and Mac versions of Anaconda and Miniconda are installed in the normal way by double-clicking the downloaded installer. Follow the prompts to install the Python distribution. Be patient as the installation can take as much as 15-20 min to complete.

You will interact with Python from a terminal prompt. On macOS you can reach the terminal from your Applications/Utilities directory. On Windows, use the search feature next to the start menu to search for Anaconda Navigator. From there open the Powershell to access the command line.

Linux 🐧

The Linux version is installed by running the 'sh' script from the command line in a terminal: `bash Anaconda3-2022.05-Linux-x86_64.sh`

After accepting the license agreement, you will be asked for a location into which to install the distribution (e.g. the default /home/username/anaconda3). All of the Python packages we will be using will be installed under the directory you specified. This will take quite some time to complete. When you are asked "Do you wish the installer to initialize Anaconda3 by running conda init?", answer "yes". This will add code to your `.bashrc` file which initializes Anaconda Python whenever you open a new terminal window.

Now close the terminal and open another (so that the new `.bashrc` takes effect).

Update to the Latest Version 🪟🍎🐧

Next update the core conda package to the latest version by typing:
`conda update conda`

Answer "y" when asked if you wish to update, and wait for the update to finish.

If you would like to update *all* of the packages that come with Anaconda to their latest versions, type:
`conda update --all`

Note that this will take some time. This step is optional, since packages can also be updated individually as needed.

You now have a working Python distribution installed in your account.

Install Packages for the Workshop 🪟🍎🐧

We will use a few additional Python packages that may not be included in the Anaconda distribution. If not already installed, you should install them by typing:

```
conda install astropy
conda install -c astropy astroquery
conda install -c glueviz glueviz
conda install -c conda-forge lmfit
```

Installing Orvara

We will use the orvara package to fit orbits of substellar companions. The code can be found here:

https://github.com/t-brandt/orvara

Here are the installation instructions:

Choose a directory where you want to install the orvara package. *Please make sure that the directory path does not contain any names with spaces*; for example, "/home/user/Sagan Workshop/orvara" should not be used (it will cause problems when orbital fits are run). In this directory type:

```
git clone https://github.com/t-brandt/orvara
```

Then install cython by typing:

```
pip install cython
```

The first command will create a directory called "orvara". Change into that directory, then type

```
pip install -e .
```

To test that your installation worked, while in the "orvara" directory, type

```
pytest -sv -m "not e2e"
```

This will run some quick tests. If successful (a few warning messages are o.k), you are all set!

Custom Working Environment (*Optional*) 🪟 🍎 🐧

If you are developing code or running a program that requires you to install custom packages, it can be useful to create a virtual environment dedicated to that project. This environment allows you to create local installations of your Python environment for specific purposes; if you do something undesirable in a virtual environment, you can delete it and start over, without having to reinstall conda.

Note: This step is optional for the Sagan Workshop. It can be useful for users who are already using Python for other projects, and want to keep the workshop materials separated out.

To create a virtual Python3 environment for use with the Sagan Workshop, type the following:
```
conda create --name sagan python=3.9
```

This creates a directory ~/anaconda3/envs/sagan (C:\Users\username\anaconda3\envs\sagan under Windows), which contains a local copy of your Python directory. You can modify the Python environment in here without risk of damaging the base install.

To 'activate' your custom environment, type:
```
conda activate sagan
```

Note: You may have to initialize your shell first before this command can be run. To do this, type
```
conda init <Shell_name>
```
where <Shell_name> should be either bash, fish, tcsh, xonsh, zsh, or powershell, then close and restart your shell.

Once you have activated your custom environment, the beginning of your terminal prompt should change to [sagan], and any packages that you install now will end up in ~/anaconda3/envs/sagan. Note that each new terminal you open needs to have the activate command run on it.

To 'deactivate' a custom environment, type:
```
conda deactivate    or    conda activate
```
(with no environment specified). That will return your terminal to the base environment.

By default, conda environments contain only a minimal set of packages. If you are working in an environment, you will need to install packages as necessary for the workshop; here are the main ones:
```
conda activate sagan        conda install jupyter
conda install numpy         conda install matplotlib
conda install scipy         conda install -c glueviz glueviz
conda install astropy       conda install PySide2
conda install pandas        conda install -c conda-forge lmfit
```
Note: If any of the above commands does not work, instead of using
```
conda install package_name try conda install -c conda-forge package_name
```


Python Jupyter Notebooks

Jupyter notebooks are a handy, interactive way to work through programming problems in a GUI environment in your web browser. They allow you to easily document your work (just like a real lab notebook) and visualize the results of each step in a computer program. The activities for the Sagan Workshop are in the form of Jupyter notebooks (files ending in 'ipynb'), so if you are not familiar with them, read the following instructions.

Jupyter comes included with the full Anaconda distribution.

If you installed miniconda, run the following to add the Jupyter notebook package:
```
conda install jupyter
```

You can start running a Jupyter notebook by typing the following command in a terminal window (you should type this in the directory where the Jupyter notebook is located, or any directory above it):

```
jupyter notebook
```

■ *Note: Windows users can use Anaconda Navigator to access the Jupyter notebook interface.*

This will launch the Jupyter kernel and load an interface in your local web browser where you can see the content of the directory where you started Jupyter. To open a Jupyter notebook, simply click on the file (it ends in 'ipynb'), and the notebook will open up in a new browser window. To execute code in a Jupyter notebook cell, type CTRL-Enter (or Shift-Enter). Note that Jupyter note-books are automatically saved, so it is helpful to create a backup copy of any notebook before working with it.

When you are done, click on the Jupyter icon ⌣ in the upper left. This will return you to the di-rectory interface, where you can load additional notebooks.

To close a Jupyter notebook, in the directory interface select the notebook and click on "Shut-down"; close the browser window. Then, if you launched the notebook from a terminal window, press Control-C twice in that window to shut down the notebook server. Another option to shut down the server is to click on "Quit" in the top right corner of the directory interface page and close the browser window.

Glue ■■🐧

Glue is a visualization tool to explore datasets. You can find more information here:

https://glueviz.org/index.html

If not already done, you can install Glue with:

```
conda install -c glueviz glueviz
```

Additional installation instructions can be found here (in particular, there are standalone applica-tions for MacOS X and Windows – not needed if you have already installed it with 'conda install' in the previous step):

https://glueviz.readthedocs.io/en/stable/installation/installation.html

Once installed, Glue is launched from the command line by typing:

```
glue
```

Basic GUI features are described here:

https://docs.glueviz.org/en/stable/getting_started/index.html

Creating a Gaia Archive User Account

Some of the exercises in this hands-on session will require you to be registered as a user of the Gaia Archive (if you plan on making use of Gaia data regularly, this is in any case a good idea). The registration is easy and quick (and free) and offers the following features:

- 20 GB of space to store query results
- the option to create and share your own tables ('user tables')
- longer timeout for large or complex query jobs

Follow this link to register: https://www.cosmos.esa.int/web/gaia-users/register.

Downloading Files for the Workshop

For the Jupyter notebooks below, simply save them from your browser window by using the menu option File → Save Page As, and then save the file with the ipynb extension for the links below.

Another option is to download all notebooks (except the Orvara Tutorial notebook available with the git clone of orvara) at:
https://catcopy.ipac.caltech.edu/ssw2022/hands-on/SSW2022_jupyter_notebooks.zip

Gaia Catalog and Visualization Hands-on Session

SSW2022_Jupyter_GaiaHandsOn.ipynb notebook
https://catcopy.ipac.caltech.edu/ssw2022/hands-on/SSW2022_Jupyter_GaiaHandsOn.ipynb

> ➤ **Here is the notebook with answers (SSW2022_Jupyter_GaiaHandsOn_Answers.ipynb):**
> https://catcopy.ipac.caltech.edu/ssw2022/hands-on/SSW2022_Jupyter_GaiaHandsOn_Answers.ipynb

SSW2022_Jupyter_GaiaHandsOn-TransitingPlanetsDR3.ipynb notebook
https://catcopy.ipac.caltech.edu/ssw2022/hands-on/SSW2022_Jupyter_GaiaHandsOn-TransitingPlanetsDR3.ipynb

> ➤ **Here is the notebook with answers (SSW2022_Jupyter_GaiaHandsOn-TransitingPlanetsDR3_Answers.ipynb):**
> https://catcopy.ipac.caltech.edu/ssw2022/hands-on/SSW2022_Jupyter_GaiaHandsOn-TransitingPlanetsDR3_Answers.ipynb

Glue visualizations: Gaia Catalog of Nearby Stars and Exoplanet Archive
- https://catcopy.ipac.caltech.edu/ssw2022/hands-on/GCNS_cat.fits.zip
- https://catcopy.ipac.caltech.edu/ssw2022/hands-on/GCNS-Matched-DR3.fits.zip
- https://catcopy.ipac.caltech.edu/ssw2022/hands-on/Exoplanet-Archive-GCNS-Matched.fits.zip

Download these three files and unzip them (note that the first two files are fairly large, 100-300 MB).

Orbital Fits using Orvara Hands-on Session

Orvara Tutorial notebook
The Tutorial.ipynb notebook is included with the git clone of orvara.

SSW2022_Jupyter_Orvara_Projects.ipynb notebook
https://catcopy.ipac.caltech.edu/ssw2022/hands-on/SSW2022_Jupyter_Orvara_Projects.ipynb

Orvara Data Sets for Projects:
You can find the links to the data sets in the SSW2022_Jupyter_Orvara_Projects.ipynb notebook and also below:

- RV and astrometry data:
  https://catcopy.ipac.caltech.edu/ssw2022/hands-on/RV_astrometry.zip
  or
  https://catcopy.ipac.caltech.edu/ssw2022/hands-on/RV_astrometry.tar.gz
  Unzip/unpack these files and copy them to orvara/tests/data.
- Hipparcos data:
  https://catcopy.ipac.caltech.edu/ssw2022/hands-on/Hipparcos_Gaia_Data.zip
  or
  https://catcopy.ipac.caltech.edu/ssw2022/hands-on/Hipparcos_Gaia_Data.tar.gz
  Unzip/unpack these files and
  - copy Hipparcos_Gaia_Data/gaia files to orvara/tests/data/gaia
  - copy Hipparcos_Gaia_Data/hip1 files to orvara/tests/data/hip1
  - copy Hipparcos_Gaia_Data/hip21 files to orvara/tests/data/hip2

Challenge: Orbital Fitting of Gaia-like Data
https://catcopy.ipac.caltech.edu/ssw2022/hands-on/SSW2022_Jupyter_Orbit_Fit_Challenge.ipynb

➤ **Here is the notebook with answers (SSW2022_Jupyter_Orbital_Fit_Challenge_Answers.ipynb):**
https://catcopy.ipac.caltech.edu/ssw2022/hands-on/SSW2022_Jupyter_Orbit_Fit_Challenge_Answers.ipynb