How good is your model?
Comparison with Observations

Objectives

Having created a synthetic population of exoplanets, we would like to evaluate whether our model is a reasonable representation of reality. In the following we will use some simple tools to compare our model against exoplanet observations. While the exercises considered here will not produce publishable results, parts of the code may be directly useful for your future work (e.g. the data archive access and plotting tools). The overall goal, however, is not for you to simply run the code, but to understand its strengths and limitations. Throughout the exercises, we encourage you to think about what the code is doing, what it is saying about the model, and what modifications you might want to make in order to improve the general analysis or to address a specific science question.

Outline

This exercise consists of three main steps:

1. First we will download data for observed exoplanets from the NASA Exoplanet Archive and plot the overall distribution of mass and semi-major axis.
2. Next we will add simulated data to the plot and roughly quantify whether the synthetic population is consistent with the observations.
3. Finally we will consider selection effects within the observations, focusing first on radial-velocity planets, then on Kepler-detected planets.

Prerequisites

You must have an account on IPAC's server (same as for the previous hands-on sessions). The required python files will already be in your `wednesday_handsOn` directory. If you have made modifications and want to start over from scratch, the original files can be copied from `/ssw/wednesday_handsOn`.

1. Introduction

If you only have a few minutes to spend on this, here's the short version.

Quick start

To run the program:

```
python archiveComparison.py
```

To change the input parameters:

```
xemacs params.py
```

To view the results:

```
xv *.png
```

2. Planet Data from the Exoplanet Archive

We start by downloading tables of exoplanet data from the NASA Exoplanet Archive.

The archive website stores the data in two distinct tables - one for all confirmed planets and another for all Kepler planets, both confirmed and candidates. The tables currently contain 1858 and 4661 planets respectively, of which 5521 are unique and 998 are duplicates. Running the program (`python archiveComparison.py`) downloads these two tables into the `archiveData` directory (as `confirmedPlanets.csv` and `keplerPlanets.csv`). The overlapping tables are then combined together to create a master list of all observed planets. Fields that exist in both tables - e.g. `pl_orbper` and `koi_period` which both refer to a planet's orbital period - are merged into a single non-redundant field. During this merging procedure, priority is given to information from the confirmed planet table, which comes directly from published works and has been vetted to a greater degree.

The first time the program is run (`python archiveComparison.py`), it downloads the tables of observed exoplanets and puts this data into the `archiveData` directory. Subsequent runs will access the files in that directory. If you want to update the tables with the latest archive data, simply delete the saved files and they will be downloaded once again.

After downloading and saving the data for observed exoplanets, the program will create a plot showing the two top-level planet characteristics - semi-major axis and mass. For now only the confirmed planets are shown. Look at the outputted file `semiaVSmass.png`. The data has been divided into three categories - radial-velocity discoveries ($\triangle$ markers), transit discoveries ($\circ$ markers), and the rest ($+$ markers), including those discovered via microlensing, direct imaging, pulsar timing, and astrometry. We will consider these samples separately below.

To help distinguish between the different symbols, you may want to give them each a unique color. Edit the parameters `rvColor`, `transitColor`, and `otherColor` to your liking in the parameter file `params.py`. You can also adjust the size of the points (`pointSize`) in the parameter file.

Question2a: Where do radial-velocity detections fall in the planet mass vs semi-major axis plot? Transit detections? Can you guess the detection method for the remaining ``other'' detections?

Bear in mind that the different detection methods do not all observe the same exoplanet characteristics. In particular, one of the plotted quantities - exoplanet mass - is not generally a directly observed quantity. Radial velocity variations, for example, measure $M_p \sin(i)$, planet mass times the orbit inclination. Unless the planet is also transiting, this orbital inclination is completely unknown. Transit observations without accompanying radial-velocity measurements, on the other hand, do not give any information on planet mass, but instead measure the planet radius. *Some assumptions have to be made in order to plot all planets on the same scale.*

For non-transiting radial-velocity planets, we simply assume an orbital inclination of 60° from face-on, such that M_p = 1.15 $M_p \sin(i)$. This is the median inclination for a randomly oriented orbit. Much larger planet masses are possible if an orbit happens to be face-on.

For transiting planets without radial-velocity measurements, we have to convert planet radius to planet mass. We use a simple equation originally developed by Wes Traub. This formula has three key characteristics: 1) planets smaller than Earth have Earth density, 2) between Earth and Jupiter mass the density decreases, and 3) the maximum size is Jupiter radius. Note that there is no physical modeling behind this adopted equation; it is based on a fit to the planets of the Solar System. Still, it does contain the important physical effect of degenerate electron pressure, which limits planets to radii about equal to Jupiter's. To view the equation and see how it compares to the Solar System, set `plotMassRadius` to `True` (in `params.py`), resulting in a plot called (`MassvsRadius.png`).

Question2b: Beyond semi-major axis and planet mass, what other exoplanet characteristics might be worth exploring?

Question2c: Does the Exoplanet Archive provide enough information to calculate detection rates for different types of planet, e.g. eta_Earth, the frequency of habitable Earth-like planets?

Optional: modify the downloaded dataset

The code provided to you is currently set up to download all information needed for the following exercises. Should you wish to include additional data fields that are more relevant for your own research (e.g. `st_dist`, the distance to each star), the data fields that are downloaded are specified in `params.py` (as are the names of the downloaded files).
The names and descriptions of all possible columns within the the two tables are listed online at confirmed data fields and KOI candidate fields.
Note that you should not select the archive's default columns for the two tables; they do not contain all of the fields used for this exercise, e.g. R.A. and declination, which are used to cross-match the two tables.

3. Comparison between model results and real data

Now that we have the latest exoplanet data from the NASA archive, let's compare against the synthetic population produced by our model. This exercise is designed to take the output from yesterday's session as in input. *You must tell the program which simulation file you want to use, and where it is located.* The input file should be a snapshot containing data for many planets at a single epoch; such files are named with the prefix `ref_red` followed by age of the systems. Set the parameter `popsynthFile` in `params.py` to the name of the synthetic population file that you would like to use, e.g. `ref_red1e10.dat`. You can either copy this file into the `popsynth` directory here or you can set the parameter `popsynthDir` to the current file location.

Once you've identified the model results file name and location within the `params.py` file, rerunning the main program (`python archiveComparison.py`) will redraw the semi-major axis vs planet mass plot, but now with model results plotted alongside the real exoplanet data.

Question3a: How do the observed and simulated planet distributions compare by eye?

The overall goal of this exercise is to evaluate the strengths and weaknesses of the model and to potentially identify new physics that should be added to it. The best way to evaluate the model is to compare against observations and the simplest way to make this comparison is with a Kolmogorov-Smirnov (K-S) test. This generic test quantifies the likelihood that two distributions are distinct, in this case the observed and the simulated distributions.

When you run the program with both the observed and simulated data included, a K-S test is performed for two quantities - semi-major axis and planet mass. The p-value for each is printed out. A low p-value means a low probability that the two distributions come from the same underlying distribution.

Question3b: Do the results of this statistical comparisons match your intuition?

Question3c: Is it important to consider the uncertainty of each observation?

Question3d: Why are there so many simulated planets with Earth-like mass and orbital location, but so few observed?

4. Selection effects

All methods to detect exoplanets have limitations. Direct imaging requires planets that are bright relative to their parent star and that are orbiting at relatively large angular separation. Microlensing is most sensitive to planets near the Einstein ring radius, which typically corresponds to a few AU for galactic bulge observations. Transit detections require a sufficiently large planet to be observed over multiple orbits, such that short-period planets are favored. Radial velocity signals are largest for massive short-period planets.

A comparison between observed and simulated planets is meaningless unless the synthetic population has been passed through selection effects similar to those in the observed population. In the following we will attempt to filter the simulated data based on two types of selection effects - radial velocity and transits.

Radial-velocity selection

The radial velocity signal generated as a planet orbits its parent star has a magnitude of

$$K = 9\,\mathrm{cm/s}\left(\frac{M_p}{M_\oplus}\right)\left(\frac{a_p}{\mathrm{AU}}\right)^{-1/2}\left(\frac{M_\star}{M_\odot}\right)^{-1/2}\sin(i). \quad (1)$$

Extreme radial-velocity precision is needed to reach this Earth-detecting level. Past surveys have typically only achieved accuracies of $\sim$1 m/s.

Here we apply a radial-velocity selection effect on the model results and compare the resulting planet distribution against all planets discovered by radial-velocity signal. To turn on this option, set `RVComparison` to `True` in `params.py`.

While real surveys have an accuracy that varies from night to night and from star to star, here we set a uniform detection threshold of 1 m/s for the simulated data. We also set a maximum time baseline for observations at 5 years; planets with longer periods are not observed for a full orbit, significantly reducing their detectability. Both of these parameters (`RVcut` and `RVtime`) can be adjusted (in `params.py`).

After running the program with the radial-velocity selection effect turned on, the resulting plot is saved as `RVComparison.png`. Note that the y-axis of the plot is now the observable $M_p \sin(i)$, not M_p. Rather than assume a median inclination for the observed planets (as in the preview two sections), we assign a random inclination to each simulated planet.

As for the previous case without selection effects included, the K-S test results are printed out comparing the observed and simulated distributions, but now for semi-major axis and planet mass $\times$ sin(inclination).

Question4a: How do the observed and simulated planet distributions compare by eye, now that we've included radial-velocity selection effects?

Question4b: What is the biggest discrepancy between the observations and the synthetic population?

Question4c: How might the model be modified to create a better fit?

Kepler selection

Finally we consider the selection effects from a transit survey. The transit depth depends only on the ratio of star and planet radii.

$$\text{transitdepth} = 83\,\text{ppm}\left(\frac{R_p}{R_\oplus}\right)^2\left(\frac{R_\star}{R_\odot}\right)^{-2} \quad (2)$$

Similar to the radial-velocity selection, we set a threshold detecting the simulated data and a timespan for the synthetic observations (`transitCut` and `tranTime` in `params.py`).

To turn on a Kepler based selection effect, set `KeplerComparison` to `True` (in `params.py`). The resulting plot (`keplerComparison.png`) now shows planet radius on the y-axis. Radii of the model planets are calculated with the simple equation described earlier. For the observed planets, rather than combine the results from many surveys, we include only Kepler detections.

Question4d: What is the biggest discrepancy between the observations and the synthetic population?

Again, the K-S test results are printed out comparing the observed and simulated distributions, in this case for semi-major axis and planet radius.

Question4e: Based on the K-S test, is the match better for semi-major axis or for planet radius?

Question4f: How might the model be modified to create a better fit?

Optional: include candidate planets

All of the above plots can be recalculated with Kepler's candidate planets included. Simply set `onlyConfirmed` to `False` (in `params.py`) and rerun the program.

Question4g: Do you notice any issues with the candidate planet radii? Semi-major axis?

5. Discussion

We have taken a pure simulated distribution and filtered it with selection effects in a simple attempt to match the observations. We have not considered the reverse - trying to remove selection effects from the observed data and thereby trace back to the true underlying distribution of exoplanets. Calculating the completeness of various observational surveys is well beyond the scope of this exercise, but is a crucial topic for the community to explore.

Also, while this exercise did not go into any advanced statistical methods, note that Penn State offers regular week-long courses on statistics for astronomers including Bayesian analysis packages such as emcee.

Lastly, remember that the top level goal of this exercise is to use observations to improve our understanding of how planets form. Tomorrow, while working on your group projects, use the tools provided here to evaluate the success of your model and to suggest improvements to it.

Question5a: Despite the limiting assumptions of this analysis, does it provide any insight toward the physics of planet formation?

Question5b: How would you improve the code?

Appendix

Input Parameters

The following parameters can be easily changed within `params.py`:

Variable	Default	Description
confirmedDataFile	exoplanetArchive.csv	saved file of confirmed planets
confirmedDataFields	(see params.py)	desired columns from the confirmed planet table
keplerDataFile	keplerCandidate.csv	saved file of Kepler planets
keplerDataFields	(see params.py)	desired columns from the Kepler planet table
archiveDir	archiveData	directory for the above files
popsynthFile	ref_red1e10.dat	file with synthetic population results
popsynthDir	popsynth	directory for the simulation files
onlyConfirmed	True	only plot confirmed planets?
plotPhaseSpace	True	plot the overall planet phase space?
plotMassRadius	False	plot the function used to convert R to M?
RVComparison	False	compare RV-selected population against RV data?
RVcut	1	minimum detectable RV signal (m/s)
RVtime	5	timespan for synthetic RV observations (years)
KeplerComparison	False	compare transit-selected population against Kepler data?
tranScale	10	scaling for transit-like selection effect
transitCut	100	minimum transit signal (ppm)
tranTime	4	timespan for synthetic transit observations (years)
modelColor	orange	color for simulated exoplanets
rvColor	purple	color for RV-detected exoplanets
transitColor	purple	color for transit-detected exoplanets
otherColor	purple	color for other-detected exoplanets
pointSize	10	size of plot points

Exoplanet Archive Data

Interactive tables of the data used in this exercise can be accessed from click boxes near the top of the NASA Exoplanet Archive main page.
Alternately you can go directly to the confirmed planet table via this link for confirmed planets while the Kepler candidate table is at this link for Kepler planets.
There is extensive documentation available on the archive site. In particular the names of all the possible data fields within the two tables are described at confirmed data fields and KOI candidate fields.

Model Data

The output files from the population synthesis model are described in the Monday/Tuesday instructions, §3.7, where the column definitions are given for the type data used in this exercise (a file listing all the planets at a single point in time). Columns 5 and 10, for example, are the planet mass and semi-major axis. The column names can also be viewed as `fieldList` in the `readSyntheticPopulation` function in `readers.py`.

Figures

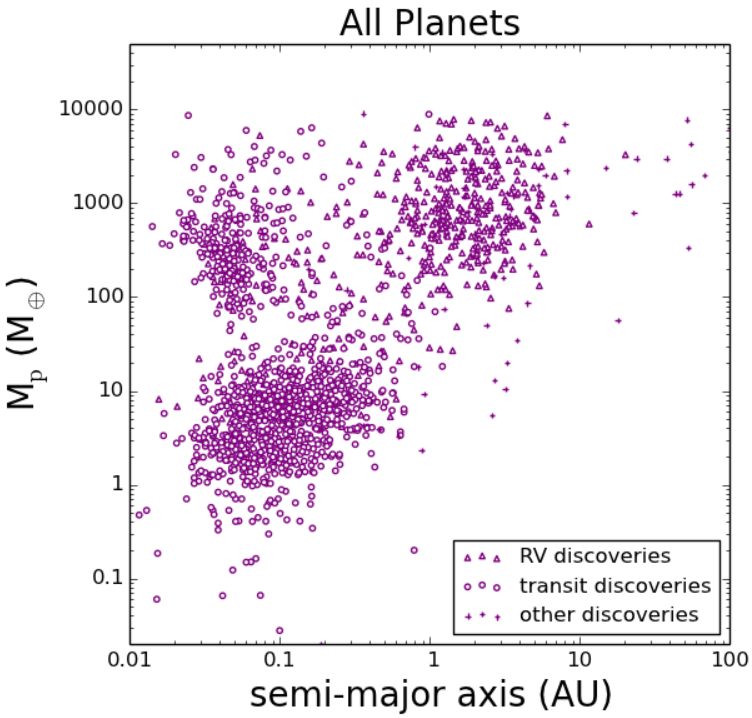

Figure 1: Mass and semi-major axis distributions for confirmed planets in the NASA exoplanet archive.

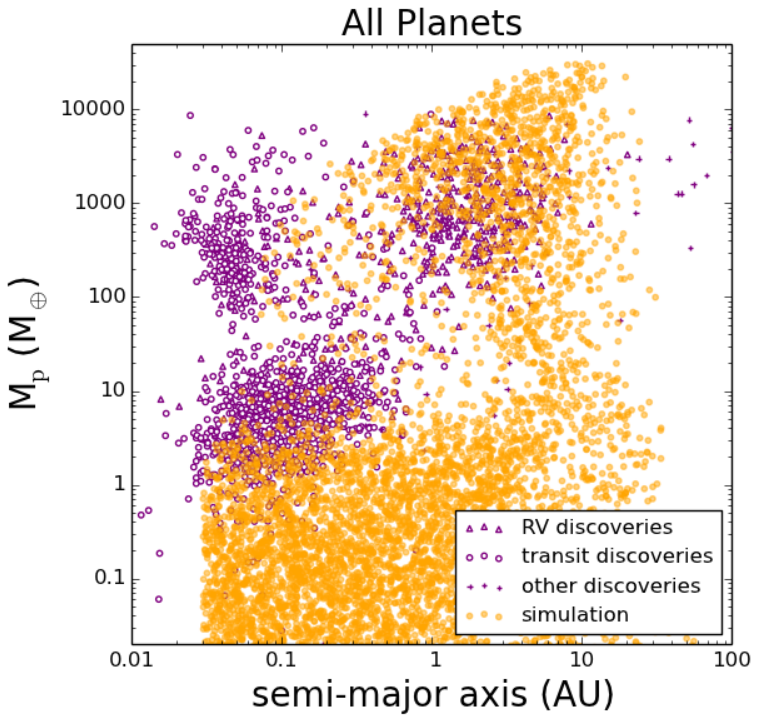

Figure 2: Same as previous figure, but now with the simulated planets also shown.

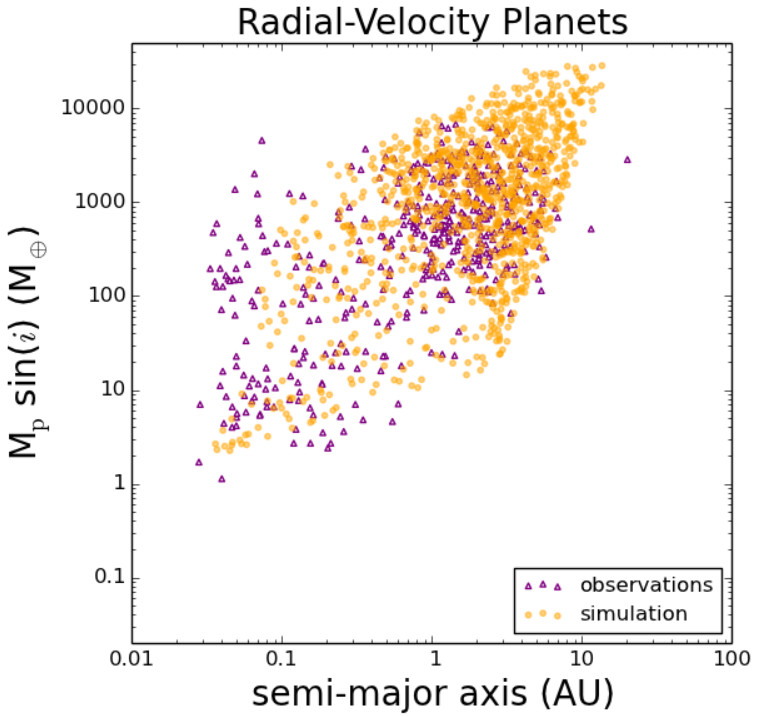

Figure 3: Observed and simulated radial-velocity detections.

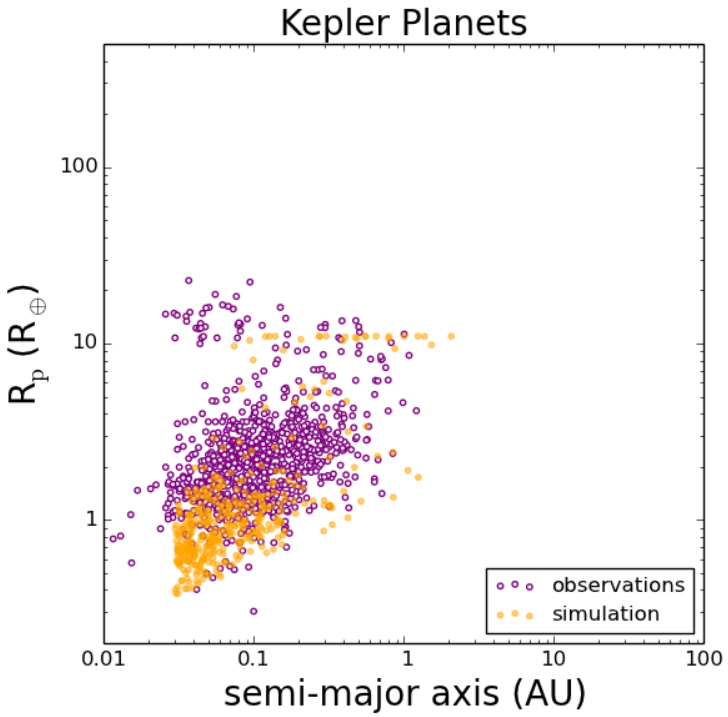

Figure 4: Observed and simulated Kepler detections.

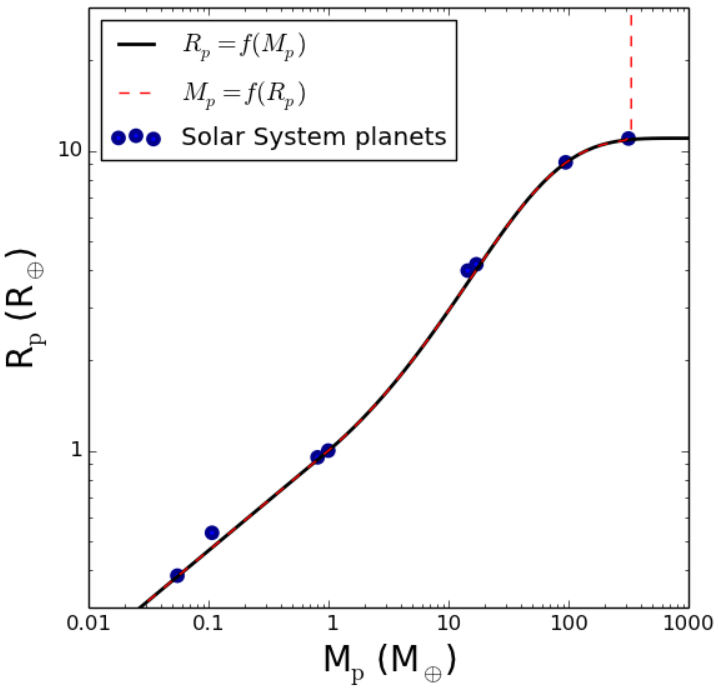

Figure 5: The assumed function for calculating planet radius as a function of mass (black line). The reverse formula (mass as a function of radius) is shown as a red dotted line. Solar System planets are shown as blue circles, except for Pluto, which is of course also a planet.