So you want to generate PD SEDs:

But wait...there's so much set up work to be done??!!!

This doc will hopefully provide reasonably thorough instructions on how to filter out galaxies from a full snapshot of a cosmological simulation, assuming it's one of Illustris TNG, Simba, or Eagle, starting from downloading the simulation if need be. If you already have some of the data listed, you can skip to the appropriate step for the specific simulation you are working on.

The steps to setting up a simulation and powderday run are broadly (I'll go into further detail when breaking down each simulation below):

1. If using Illustris TNG or Eagle, download the snapshot you need from the simulation page. Because these snapshots are big (~1.7 T for any TNG), we can really only have one full snapshot on our /orange/ allocation at a time. You'll need both the raw particle data and the information from the associated FOF algorithm. Simba uses caesar, while TNG and Eagle use Subfind.
2. Selecting the galaxies you wish to filter from the full snapshot.
3. Filtering these selected galaxies into an individual snapshot for each galaxy.
4. Generating the auxiliary position .npz file and model.py scripts for powderday
5. Actually running powderday, but I leave that explanation for the powderday docs
6. Post-powderday, generate the files for use with mirkwood (or any general use case that needs synthetic broadband photometry and galaxy properties compiled into one place)

Each simulation has its own quirks on how to accomplish each step. Of course, even with these instructions, things are bound to mess up and be very confusing. So please don't hesitate to ask questions!

Happy RTing!

Simba

0. To work with simba you'll need to have installed caesar[1], powderday[2], yt[3], sl_simulation_tools[4], and h5py[5], with no specific hashes of any code (i.e. any version that isn't ancient will work)

To run powderday on a simba galaxy, first we need to filter galaxies from the particular box and then redshift you desire:

- If you need an SED from the m25 or m100 box at low redshift, chances are we have those in storage already. If you need another box size or an m100 snapshot at high redshift, we will probably need to download those.
- Simba is technically publicly available, but you'll need to email Romeel Dave, however it's best to ask Desika to be the middleman for this because it's more hassle to try and get you access to the Edinborough cluster than it is for Desika to just transfer it over for you.

Then assuming you have the snapshot(s) on disk that you need, we can start the filtering process:

1. Select the galaxies you want to filter. This again depends typically on the box size. You can easily filter every galaxy from an m25 snapshot, since there's typically <3000 identified by caesar. For larger box sizes, that may have >10,000 galaxies, you can limit the number you want to filter by selecting galaxies from the caesar snapshot against some criteria

```
import caesar
caesar_file = '/orange/narayanan/desika.narayanan/gizmo_runs/simba/m25n512/output/Groups/caesar_0305_z0.000.hdf5'
obj = caesar.load(caesar_file)

selection = []
for i in range(len(obj.galaxies)):
    if obj.galaxies[i].masses['stellar'] > 1e10:
        selection.append(i)
```

    a. e.g. the above snippet will select galaxies at z=0 with a stellar mass above 1e10 Msun. You can then filter only those galaxies instead of every galaxy.

2. Filter the galaxies you want. You'll do that by running sl_simulation_tools/filter_simba.py
    a. This script takes 4 command line arguments
        i. The snapshot path

[1] https://github.com/dnarayanan/caesar
[2] https://github.com/dnarayanan/powderday
[3] https://github.com/yt-project/yt
[4] https://github.com/smlower/sl_simulation_tools
[5] Command line install with `$conda install h5py`

ii. The snapshot number
iii. The galaxy to be filtered
iv. The output path

b. In the script, we access the caesar file and select the star and gas particles that belong to the galaxy according to caesar. We then use those particle IDs to index the full snapshot, copying over the star and gas properties for that galaxy into a new snapshot.

3. Once the galaxies are filtered, we can generate a file containing the center-of-mass positions that will be used with powderday. You'll do that by running sl_simulation_tools/galaxy_positions.py
   a. This script takes two command line arguments:
      i. The path to the filtered snapshots
      ii. The snapshot number
   b. This will generate an .npz file of positions that powderday will use
4. Now that we have the filtered galaxies and their positions, we can initialize our powderday run. We'll do this by running sl_simulation_tools/powderday_setup.py
   a. This script generates the 'model' scripts used by powderday, in tandem with the parameters_master script, that specific the galaxy position, etc.
   b. This script has a couple of lines at the top that need to be modified for each simulation/snapshot, to point it to where your pd runs will be saved to and where it can find the positions file we just generated.

Illustris TNG

0. To work with Illustris TNG, you'll need powderday[6], the yt[7] 'desika_octree' branch from Ash Kelly's yt4.0 fork, sl_simulation_tools[8], and h5py[9]. The stable yt hash is db35aa511619845fce57b5694e37be9ff8821723 which is confirmed to work with the 9c5d94383bb7102e48a4d662201a162204566a56 powderday hash.

I highly recommend making a separate conda environment for the TNG stuff since Arepo needs to be handled so much differently than Eagle or Simba.

---

[6] https://github.com/dnarayanan/powderday
[7] https://github.com/AshKelly/yt/tree/desika_octree
[8] https://github.com/smlower/sl_simulation_tools
[9] Command line install with `$conda install h5py`

First, we need to download the snapshot we need from the TNG public data archive. Since the TNG snapshots are so big (~1.7T), chances are the snapshot you need is not idly on hpg.

1. Go to https://www.tng-project.org/data/ and register to get access. Usually takes less than a full business day.
2. Log in and select the simulation you need (presumable the TNG 100 Mpc box, which is at https://www.tng-project.org/data/downloads/TNG100-1/)
3. These snapshots take a while to download, so I recommend using a job script to download them instead of doing it interactively. You'll need to download the snapshot, the group catalogue, and the offsets

Download Snapshot	**Download FoF & Subfind**	**Download Offsets**
[Snapshot] (494.1 GB)	[Groupcat] (12.5 MB)	

4. If this is the first time you're filtering for TNG, you'll also need to download the 'snapshot' file. I won't pretend to understand the ins-and-outs of this file, but it essentially provides a link system to quickly parse through the >400 hdf5 snapshot files. You'll find this file under simulation.hdf5 file near the bottom of the TNG data page.

Once you have the TNG data on disk, we'll need to do some prep work on your directories to organize the data the way the simulation.hdf5 file expects things to be. In your base TNG directory, which might look something like

$ /orange/narayanan/s.lower/TNG/

create an "output" directory containing a "snapdir_099" subdirectory (for snap number 99, this will contain the snapshots) and a "group_099" subdirectory (which will contain the fof files), along with a "postprocessing" directory containing an "offsets" subdirectory (these will be the files outlining the particles in each galaxy). This will then allow us to parse the TNG data much faster than trying to manually go through each hdf5 subfile.

Now we can start to filter galaxies.

1. First, we need to select the galaxies we want to filter. We need to do this (as opposed to simba where this was ~optional) for two reasons.
    a. Assuming we're using the 100 Mpc box size, there are roughly 4,000,000 identified galaxies….which is a lot. And all of hpg would struggle for months to run radiative transfer on 4,000,000 galaxies
    b. Not every galaxy identified is what we would consider a galaxy. This is because the Subfind FOF algorithm identifies a halo as any 32 particles grouped together, even if there are no gas or stars.
    c. Like simba, we can do this beforehand easily, by asking for galaxies that meet a stellar mass and gas mass criteria, just so we know we're dealing with things we'd consider galaxies.

```
import yt
sim_file = h5py.File('/orange/narayanan/s.lower/TNG/simulation.hdf5', 'r')
fof_file = yt.load('/orange/narayanan/s.lower/TNG/output/groups_033/fof_subhalo_tab_033.0.hdf5')
galaxy_list = []
glengths = sim_file['/Groups/33/Subhalo/SubhaloLenType']
slengths = sim_file['/Groups/33/Subhalo/SubhaloLenType']
num_galaxies = len(sim_file['Groups']['33']['Subhalo']['StellarMasses']['StellarMasses_in_r100ckpc'])
for i in range(num_galaxies):
    glength = glengths[i, 0]
    slength = slengths[i, 4]
    gmass = np.sum(glength) * 1e10/0.6774 #unit conversion, code_mass to physical
    smass = np.sum(slength) * 1e10/0.6774
    if (gmass > 1e3) & (smass > 5e7):
        galaxy_mass_list.append(i)
```

Even this selection might not be strict enough, so feel free to randomly sample from this subsample to cut down on computation.

2. With our galaxy selections, we can now filter the galaxies. We'll use sl_simulation_tools/filter_tng.py for this.
    a. The script takes 4 command line arguments
        i. The index to filter the galaxy selection list
            1. Right now, this assumes it's a csv with a column named 'ID'
        ii. The snapshot number
        iii. The path to the galaxy selection list
        iv. The path you want the galaxies written to
    b. Briefly, this code parses over the full snapshot, which is broken down into multiple files, finding the particles that belong to each galaxy, and copying over those particle properties. The TNG snapshots are set up such that each galaxy is assigned an 'offset', which tells us the offset, from the beginning of file 0, where the galaxy's particles are located. We use this in tandem with the SubhaloLenType entry, which tells us how many particles of each type (i.e. gas and stars) the galaxy has. We then access the file that has the location of the galaxy's offset and copy over

those particles until we reach the SubhaloLenType number for each particle type. (Trust me, before this simulation.hdf5 file, it was even worse).

3. With the galaxies filtered, we can then generate their positions file. This is done with the same script that we used above for simba, so refer to the instructions there.
4. We can also generate the powderday setup files, with the same sl_simulation_tools/powderday_setup.py script as before, so refer to those instructions as well.

Eagle

0. To work with eagle, you'll need powderday[10], yt[11], the read_eagle routine[12] sl_simulation_tools[13], and h5py[14]. Any hash of these codes should work.

First, like TNG, we'll need to download the snapshot we need. Again, because of storage, we most likely don't have the raw snapshot (i.e. unfiltered) just sitting around.

1. First, go to http://icc.dur.ac.uk/Eagle/database.php and register for an account under the 'galaxy database' section. It typically takes a business day to get the username and password.
2. Download using the link under the particle data section. Like TNG, these snapshots are spread across multiple files and they're big so best to download by submitting a SLURM job rather than doing it interactively.

In the meantime, we can thankfully select the galaxies we'll want to filter before we even have the snapshot data. This is done using Eagle's galaxy database query system. You can access that at the link under the galaxy database section. Assuming we're making a stellar mass cut to avoid any weird FOF artifacts (i.e. 'galaxies' with no stars but 32 dark matter particles), your query should look something like below. This will generate a list of galaxies, identified by SubGroupNumber (basically galaxy #) and GroupNumber (basically halo #), that meet the stellar mass and gas mass criteria. Save

[10] https://github.com/dnarayanan/powderday
[11] https://github.com/yt-project/yt
[12] https://github.com/jchelly/read_eagle
[13] https://github.com/smlower/sl_simulation_tools
[14] Command line install with `$conda install h5py`

this list as a csv and put it somewhere you'll remember on hpg.

```
SELECT
        gal.SubGroupNumber,
        gal.GroupNumber

FROM
        RefL0050N0752_Subhalo as gal
WHERE
        gal.SnapNum = 27 and
        gal.MassType_Star > 5e7 and
        gal.MassType_Gas > 0
```

1. Once the galaxies are filtered and we have our galaxy selection list, we can get to filtering the galaxies. We'll use the sl_simulation_tools/filter_eagle.py for this.
    a. This script takes 4 command line arguments
        i. Path to full snapshot files
        ii. Path to the galaxy selection list
        iii. The path you want the galaxy hdf5s written to
        iv. The number to index the galaxy selection list
        v. The script also tries to infer the box size of the snapshot, but that may need to be manually modified depending on what your snapshot file path looks like
    b. This script makes use of the read_eagle routine, which uses MPI to parallel search the snapshot subfiles for the particles that belong to the selected galaxy.
2. With the galaxies filtered, we can generate the positions npz like above and the powederday setup. Refer to the instructions in the simba section for more detail.

Congrats -- you've successfully filtered your snapshots and are ready to run powderday !! Please refer to the powderday docs if this is your first time running.

Epilogue, Or: how to live with powderday data & use it with mirkwood

Now that you've successfully run powderday, we can compile the output from both powderday and the snapshots themselves to make two files for use with mirkwood. You can of course use this for any purpose, but the scripts have been somewhat tailored to suit mirkwood's needs.

The script we want is called sl_simulation_tools/get_properties_and_seds.py

- To run this script, you'll need to have the following packages (just listing less commonly used ones)
  - Yt
    - Follow what version you need according to which simulation you're using
  - Fsps & Hyperion
    - Both should already be installed from powderday
  - sedpy
  - astropy
- There's 5 command line arguments to run the code
  - The directory to the powderday output
  - The directory to the filtered snapshots
  - The redshift of this snapshot
  - The type of simulation (simba, TNG, or eagle) to differentiate between simba with manual dust and TNG/Eagle with dtm
  - A switch for generating the SEDs in observed or rest frame. If True, it will redshift the rest frame SED into observer frame
- The script itself is pretty self explanatory, it generates broadband photometry and pulls the galaxy properties. The photometry is generated according to the list of filters in the beginning, synthesized with sedpy.
- As a warning, this script can take a very long time (read: 2 days for 10,000 galaxies), mostly due to having to rely on fsps to calculate the SFR. Highly recommend tossing this script into a priority queue job and letting it run for a couple days.