General instructions for data files and types iPyrad can run are found here; parameters are found here; and information about iPyrad's seven steps can be found here. There's also a great walkthrough for running iPyrad on a cluster here.

Log on to TACC.

```
ssh eac3496@frontera.tacc.utexas.edu
```

Navigate to your scratch directory.

```
cds
```

Download the Day3_iPyrad_worksheet_files folder directly from Anne's TACC to your scratch directory. This may take a few minutes as it's quite large.

```
scp -r
eac3496@frontera.tacc.utexas.edu:/scratch1/03123/eac3496/training_course/Day3_ipyr
ad_worksheet_files .
```

Installing iPyrad software

Anne will guide you through these steps

Install iPyrad. First, follow the directions found under "Linux install instructions for Conda" here:

```
# Fetch the miniconda installer with wget
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-
x86_64.sh

# Install miniconda into $HOME/miniconda3
#   * Type 'yes' to agree to the license
#   * Press Enter to use the default install directory
#   * Type 'yes' to initialize the conda install
bash Miniconda3-latest-Linux-x86_64.sh

# Refresh your terminal session to see conda
bash

# test that conda is installed. Will print info about your conda
install.
conda info
```

Now, follow the instructions under "Conda install":

```
conda install ipyrad -c conda-forge -c bioconda
```

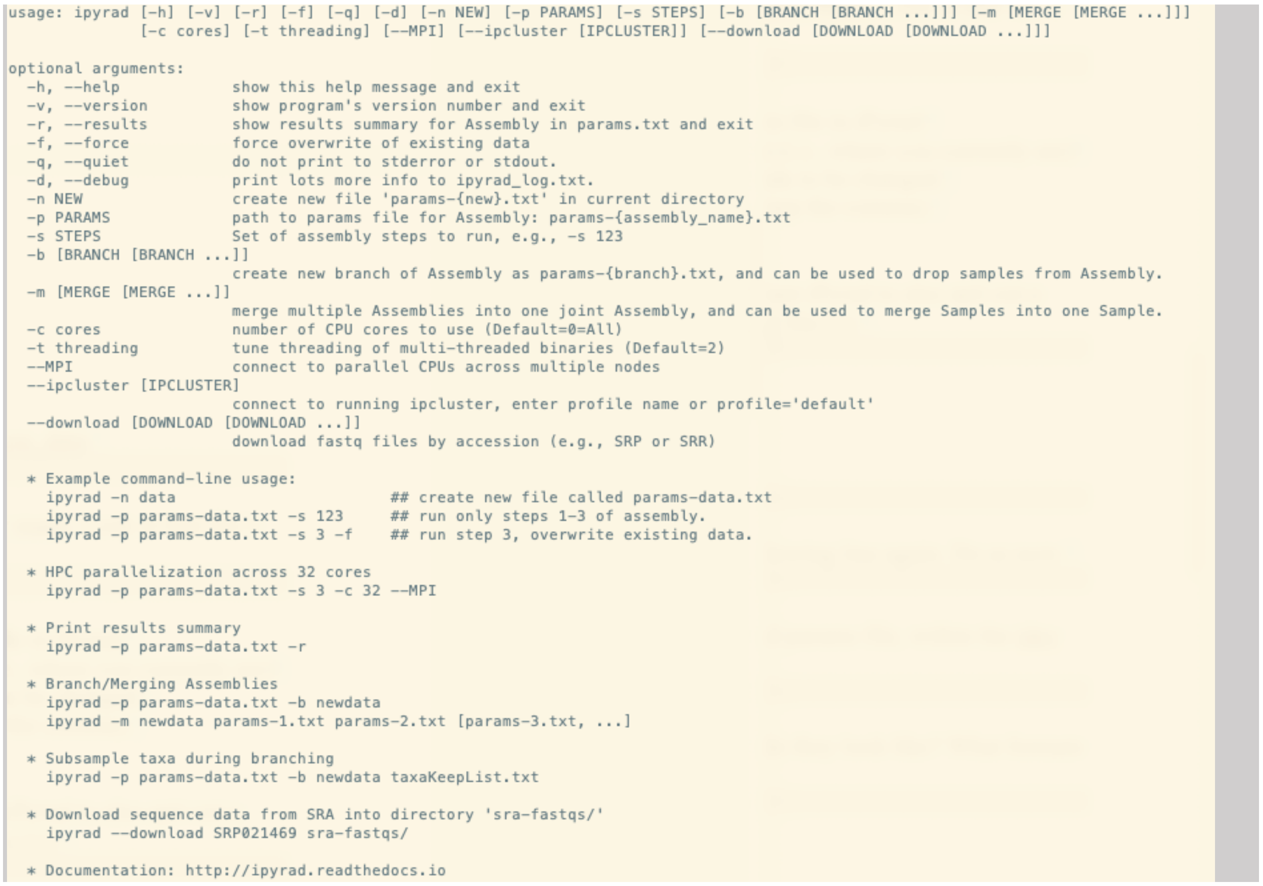
Check to see if iPyrad is working by typing the following:

```
ipyrad –h
```

If that doesn't work, enter the following and then try the above line again.

```
module unload python3
```

What does the iPyrad help (`ipyrad -h`) output look like (this may take a few seconds to run)?



```
usage: ipyrad [-h] [-v] [-r] [-f] [-q] [-d] [-n NEW] [-p PARAMS] [-s STEPS] [-b [BRANCH [BRANCH ...]]] [-m [MERGE [MERGE ...]]]
              [-c cores] [-t threading] [--MPI] [--ipcluster [IPCLUSTER]] [--download [DOWNLOAD [DOWNLOAD ...]]]

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         show program's version number and exit
  -r, --results         show results summary for Assembly in params.txt and exit
  -f, --force           force overwrite of existing data
  -q, --quiet           do not print to stderror or stdout.
  -d, --debug           print lots more info to ipyrad_log.txt.
  -n NEW                create new file 'params-{new}.txt' in current directory
  -p PARAMS             path to params file for Assembly: params-{assembly_name}.txt
  -s STEPS              Set of assembly steps to run, e.g., -s 123
  -b [BRANCH [BRANCH ...]]
                        create new branch of Assembly as params-{branch}.txt, and can be used to drop samples from Assembly.
  -m [MERGE [MERGE ...]]
                        merge multiple Assemblies into one joint Assembly, and can be used to merge Samples into one Sample.
  -c cores              number of CPU cores to use (Default=0=All)
  -t threading          tune threading of multi-threaded binaries (Default=2)
  --MPI                 connect to parallel CPUs across multiple nodes
  --ipcluster [IPCLUSTER]
                        connect to running ipcluster, enter profile name or profile='default'
  --download [DOWNLOAD [DOWNLOAD ...]]
                        download fastq files by accession (e.g., SRP or SRR)

  * Example command-line usage:
    ipyrad -n data                       ## create new file called params-data.txt
    ipyrad -p params-data.txt -s 123     ## run only steps 1-3 of assembly.
    ipyrad -p params-data.txt -s 3 -f    ## run step 3, overwrite existing data.

  * HPC parallelization across 32 cores
    ipyrad -p params-data.txt -s 3 -c 32 --MPI

  * Print results summary
    ipyrad -p params-data.txt -r

  * Branch/Merging Assemblies
    ipyrad -p params-data.txt -b newdata
    ipyrad -m newdata params-1.txt params-2.txt [params-3.txt, ...]

  * Subsample taxa during branching
    ipyrad -p params-data.txt -b newdata taxaKeepList.txt

  * Download sequence data from SRA into directory 'sra-fastqs/'
    ipyrad --download SRP021469 sra-fastqs/

  * Documentation: http://ipyrad.readthedocs.io
```

Make a new project in iPyrad

Generate a new parameters file for your data with the project name `ranaddrad`.

```
ipyrad -n ranaddrad
```

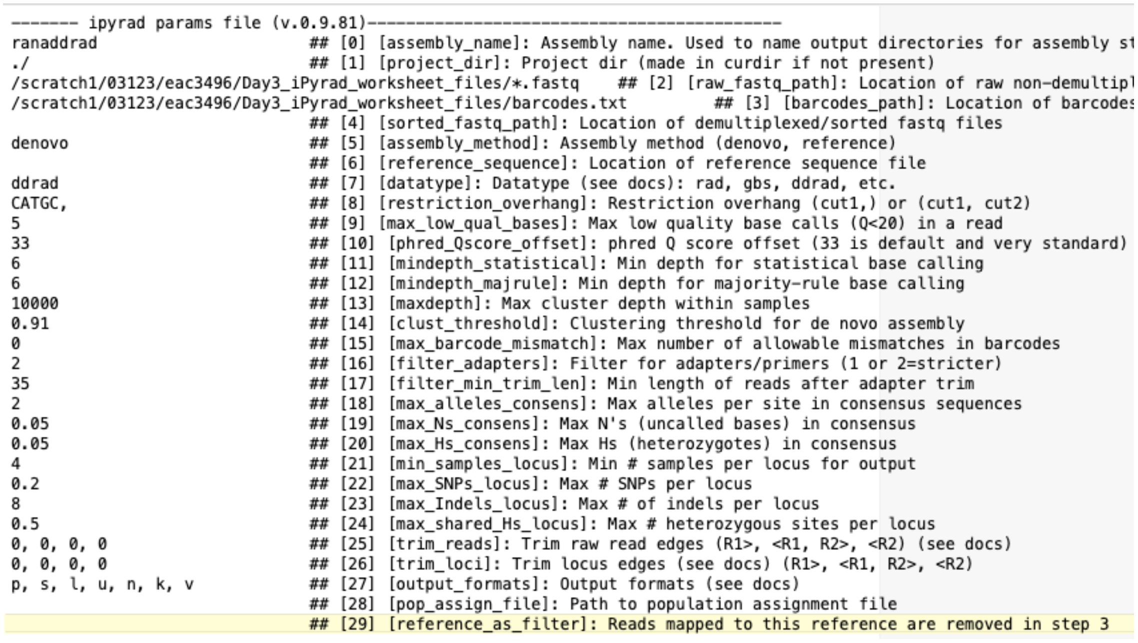
Now, check your current directory to see if a params file was actually made. Look at that file's contents now.

```
ls # there should be a file called params-ranaddrad.txt
```

Modify the following parameters in your newly created parameters file in iPyrad:

[2]: make the raw fastq path the path to the two .fastq files (i.e., where you currently are)
[3] add the path to the barcodes file (with the name of the barcodes file) here.
[7]: the data are paired ddRAD data, but **we're only going to use the forward reads** for the sake of saving time. If you wanted to use both reads, you'd put `pairddrad` here and iPyrad automatically detects R1 and R2 in the file names.
[8]: the restriction overhang is `CATGC,` (make sure you keep the comma).
[14]: clustering threshold should be set to 0.91.
[21]: should be set to 4 by default.
[27]: in addition to the default output file types, we also want iPyrad to also spit out a usnps file (u), a nexus file (n), a structure file (k), and a vcf file (v).

```
nano params-ranaddrad.txt
```



```
------- ipyrad params file (v.0.9.81)-------------------------------------------
ranaddrad                      ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly st
./                             ## [1] [project_dir]: Project dir (made in curdir if not present)
/scratch1/03123/eac3496/Day3_iPyrad_worksheet_files/*.fastq    ## [2] [raw_fastq_path]: Location of raw non-demultipl
/scratch1/03123/eac3496/Day3_iPyrad_worksheet_files/barcodes.txt        ## [3] [barcodes_path]: Location of barcodes
                               ## [4] [sorted_fastq_path]: Location of demultiplexed/sorted fastq files
denovo                         ## [5] [assembly_method]: Assembly method (denovo, reference)
                               ## [6] [reference_sequence]: Location of reference sequence file
ddrad                          ## [7] [datatype]: Datatype (see docs): rad, gbs, ddrad, etc.
CATGC,                         ## [8] [restriction_overhang]: Restriction overhang (cut1,) or (cut1, cut2)
5                              ## [9] [max_low_qual_bases]: Max low quality base calls (Q<20) in a read
33                             ## [10] [phred_Qscore_offset]: phred Q score offset (33 is default and very standard)
6                              ## [11] [mindepth_statistical]: Min depth for statistical base calling
6                              ## [12] [mindepth_majrule]: Min depth for majority-rule base calling
10000                          ## [13] [maxdepth]: Max cluster depth within samples
0.91                           ## [14] [clust_threshold]: Clustering threshold for de novo assembly
0                              ## [15] [max_barcode_mismatch]: Max number of allowable mismatches in barcodes
2                              ## [16] [filter_adapters]: Filter for adapters/primers (1 or 2=stricter)
35                             ## [17] [filter_min_trim_len]: Min length of reads after adapter trim
2                              ## [18] [max_alleles_consens]: Max alleles per site in consensus sequences
0.05                           ## [19] [max_Ns_consens]: Max N's (uncalled bases) in consensus
0.05                           ## [20] [max_Hs_consens]: Max Hs (heterozygotes) in consensus
4                              ## [21] [min_samples_locus]: Min # samples per locus for output
0.2                            ## [22] [max_SNPs_locus]: Max # SNPs per locus
8                              ## [23] [max_Indels_locus]: Max # of indels per locus
0.5                            ## [24] [max_shared_Hs_locus]: Max # heterozygous sites per locus
0, 0, 0, 0                     ## [25] [trim_reads]: Trim raw read edges (R1>, <R1, R2>, <R2) (see docs)
0, 0, 0, 0                     ## [26] [trim_loci]: Trim locus edges (see docs) (R1>, <R1, R2>, <R2)
p, s, l, u, n, k, v            ## [27] [output_formats]: Output formats (see docs)
                               ## [28] [pop_assign_file]: Path to population assignment file
                               ## [29] [reference_as_filter]: Reads mapped to this reference are removed in step 3
```

Run steps 1 and 2 in a development node

Start a development node for two hours.

```
idev -t 2:00:00 -A Phylogenomics
```

Before doing any of the steps in iPyrad, you'll need to run the following line again. Do so now.

```
module unload python3
```

Using the sequence data, the barcodes file, and your newly created params file, within the idev node, run the first ***two*** steps of iPyrad.

```
ipyrad -p params-ranaddrad.txt -s 12
```

Once this is complete, check your output files for this run. What do they look like? What formats do they have?

```
ls
# two folders should be created from steps 1 and 2: a
ranaddrad_fastqs/ folder (containing demultiplexed .fastq files) and a
ranaddrad_edits/ folder (containing trimmed .fastq files)
```

Run steps 3-6 as a job

Steps 3 and 6 are both clustering steps and are very computationally intensive, so we can't run these steps within an idev node but will have to submit them as a job.

Copy the `skeletonslurm` file you made on Day 2 into your current directory and save it as `ranaddrad_s3-7.slurm`. Note that this file name specifies both the data that will be run and the steps that TACC will be running. This helps us stay organized.

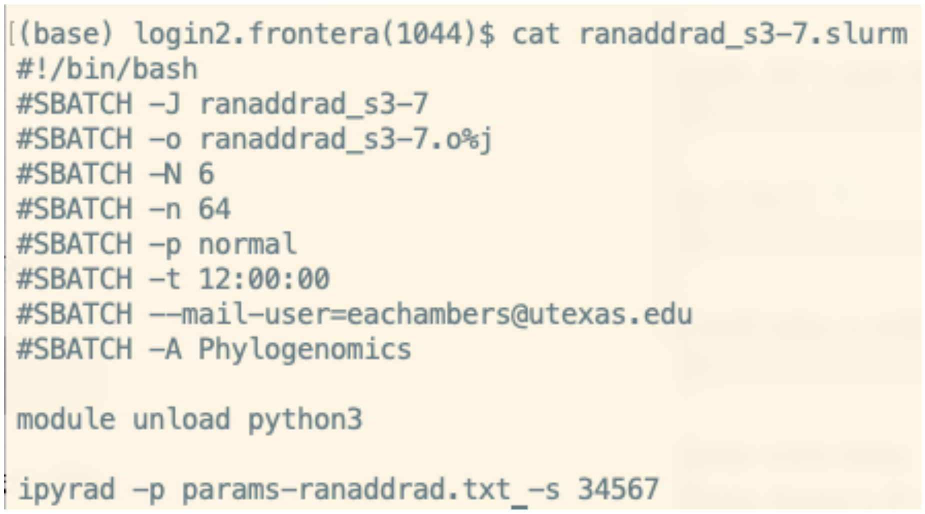
```
cp ../Day2_TACC_worksheet_files/skeletonslurm ranaddrad_s3-7.slurm
```

In the appropriate part of this file, add the line of code you to unload python3 (`module unload python3`), followed by the line of code that you want TACC to run to specify running iPyrad steps 3–7 for your data and parameters file.

```
nano ranaddrad_s3-7.slurm # to edit this file
# add on to end of file the following two lines:
module unload python3
ipyrad -p params-ranaddrad.txt -s 34567
```

Now, before we submit this as a job, let's just make sure that everything is in place to run.

```
cat ranaddrad_s3-7.slurm
```



```
[(base) login2.frontera(1044)$ cat ranaddrad_s3-7.slurm
#!/bin/bash
#SBATCH -J ranaddrad_s3-7
#SBATCH -o ranaddrad_s3-7.o%j
#SBATCH -N 6
#SBATCH -n 64
#SBATCH -p normal
#SBATCH -t 12:00:00
#SBATCH --mail-user=eachambers@utexas.edu
#SBATCH -A Phylogenomics

module unload python3

ipyrad -p params-ranaddrad.txt -s 34567
```

Submit your slurm file as a job to TACC.

```
sbatch ranaddrad_s3-7.slurm
```

Check the status of the job. This will take a while to actually run in TACC.

```
squeue -u eac3496
```

So that you can work on this in your own time, download the relevant data files (and final output from step 7) by secure copying from Anne's Frontera account:

```
scp -r
eac3496@frontera.tacc.utexas.edu:/scratch1/03123/eac3496/training_cour
se/ranaddrad_output_files/ .
```

Once you've finished running iPyrad

TACC likely hasn't had time to finish running through all of iPyrad, but you hopefully understand the general gist of it. To expedite our subsequent analyses, download the relevant output files from iPyrad, available here. *These will also be the files you'll use to build trees in RAxML-ng.* Take a look at the structure of this directory. Keep in mind that the stats files *do not* have accompanying interim data files; I've done this just to simplify things.

Move into the copied folder (`Day3_iPyrad_output_files`).

```
cd Day3_iPyrad_output_files
```

Make a *new* folder (using the `mkdir` command) called `ranaddrad_output_files` and move *all* files within your copied folder into this directory.

```
mkdir ranaddrad_output_files
mv ranaddrad* ranaddrad_output_files
mv -r s123456_stats_files ranaddrad_output_files
```

Copy your parameters file into the `ranaddrad_output_files` directory.

```
cd ranaddrad_output_files
cp ../../Day3_iPyrad_worksheet_files/params-ranaddrad.txt .
```

How many output files are there?

```
ls ranaddrad* | wc -l
# there are 12 files
```

How do these output files correspond to parameter [27] in the iPyrad parameters file?

```
# parameter 27 settings were p, s, l, u, n, k, v. These correspond to:
# p: .phy file
# s: .snps.phy file with SNPs only
# l: .loci (iPyrad-specific) file
# u: .u.snps.phy file with one SNP per locus
# n: .nexus file
# k: .str file (structure)
# v: .vcf file
```

Take a look at the final stats file (`ranaddrad_stats.txt`).

```
cat ranaddrad_stats.txt
## The number of loci caught by each filter.
## ipyrad API location: [assembly].stats_dfs.s7_filters

                            total_filters  applied_order  retained_loci
total_prefiltered_loci              65098              0          65098
filtered_by_rm_duplicates             668            668          64430
filtered_by_max_indels                 92             92          64338
filtered_by_max_snps                  461             60          64278
filtered_by_max_shared_het           1003            889          63389
filtered_by_min_sample              42648          42362          21027
filtered_by_max_alleles              2412            541          20486
total_filtered_loci                 20486              0          20486
```

```
## The number of loci recovered for each Sample.
## ipyrad API location: [assembly].stats_dfs.s7_samples

             sample_coverage
Rber_T1113a            15266
Rber_T1113b            12427
Rber_T1114              6409
Rbla_D2864             10638
Rbla_D2865              8451
Rchi_T2034a             2607
Rchi_T2034b             3523
Rchi_T2049              2544
Rneo_T480              14527
Rneo_T527              15052
Rsph_T25870             9221
Rsph_T26064             9056


## The number of loci for which N taxa have data.
## ipyrad API location: [assembly].stats_dfs.s7_loci

    locus_coverage  sum_coverage
1                0             0
2                0             0
3                0             0
4             8061          8061
5             4843         12904
6             3358         16262
7             2068         18330
8             1215         19545
9              600         20145
10             244         20389
11              86         20475
12              11         20486


## The distribution of SNPs (var and pis) per locus.
## var = Number of loci with n variable sites (pis + autapomorphies)
## pis = Number of loci with n parsimony informative site (minor allele in >1
sample)
## ipyrad API location: [assembly].stats_dfs.s7_snps

     var  sum_var   pis  sum_pis
0   2075        0  5695        0
1   2423     2423  4384     4384
2   2588     7599  3314    11012
3   2520    15159  2348    18056
4   2318    24431  1631    24580
5   2033    34596  1084    30000
6   1727    44958   818    34908
7   1381    54625   493    38359
8   1048    63009   336    41047
9    775    69984   185    42712
10   537    75354   102    43732
11   354    79248    46    44238
12   271    82500    33    44634
13   157    84541     6    44712
14   103    85983     5    44782
```

```
15     77     87138      5     44857
16     43     87826      0     44857
17     26     88268      1     44874
18     17     88574      0     44874
19      7     88707      0     44874
20      6     88827      0     44874


## Final Sample stats summary

             state  reads_raw  reads_passed_filter  clusters_total  clusters_hidepth  hetero_est  error_est  reads_consens
loci_in_assembly
Rber_T1113a      7    1325769              1325165          322924             52046    0.014998   0.005683          34309
15266
Rber_T1113b      7     990429               990042          274337             39378    0.015587   0.006248          23762
12427
Rber_T1114       7     526937               526750          155871             20028    0.016128   0.006471          10846
6409
Rbla_D2864       7    1013594              1013181          211491             42789    0.010616   0.004770          30964
10638
Rbla_D2865       7     783243               782909          187525             33110    0.011303   0.005506          22094
8451
Rchi_T2034a      7     748006               747668          208201             19182    0.016474   0.007543          10514
2607
Rchi_T2034b      7     893997               893551          231457             25511    0.014364   0.007388          15088
3523
Rchi_T2049       7     718310               717980          202394             18530    0.016205   0.007663          10125
2544
Rneo_T480        7    1080926              1080515          283418             43132    0.010911   0.005109          30778
14527
Rneo_T527        7    1079884              1079458          246068             45492    0.010514   0.004669          33543
15052
Rsph_T25870      7    1095559              1095093          320685             43811    0.014733   0.005476          28633
9221
Rsph_T26064      7    1102849              1102357          329863             43283    0.014912   0.005502          28143
9056
```

How many loci are there in the final assembly? (*two places in the stats file state this number)*

```
20486 # under "retained loci" and also "sum_coverage"
```

How many SNPs are there in the final assembly?

```
88827 # under "sum_var"
```

How many parsimony-informative sites (`pis`) are there in the final assembly?

```
44874 # under "sum_pis"
```

Which sample has the highest number of ***consensus*** reads? How many are there? Is this the same sample as the one with the highest number of loci?

```
# under Final sample stats summary
# Rber_1113a has the most consensus reads (34309)
# yes, the same sample has the most loci (15266)
```

Which sample has the lowest number of ***consensus*** reads? How many are there? Is this the same sample as the one with the lowest number of loci?

```
# under Final sample stats summary
# Rchi_T2049 has the fewest consensus reads (10125)
# it also has the fewest number of loci (2544)
```

Which stats file contains information on the depth of coverage for each of your samples?

```
# s3 stats file (ranaddrad_s3_cluster_stats.txt)
```

Which stats file contains information on the heterozygosity estimates for your samples?

```
# in the final stats file: hetero_est column, but more info is
contained within s4 stats file (ranaddrad_s4_joint_estimate.txt)
```