iPyrad tutorial

Installing iPyrad

- All written in Python
- Requires conda (anaconda or miniconda)
- Installation can be tricky! Instructions are on the worksheet but may not actually work ☺

Let's get started!

- What do we need to run iPyrad?

You've already seen what these data look like

.fastq.gz files

JR0197_KS	AACCA
JR0198_KS	CGATC
MLT64_TX	TCGAT
TBG36_TX	TGCAT
TJH1365_TX	CAACC
TJH1366_TX	GGTTG
TJH1646_TX	AAGGA
TJH2082_TX	AGCTA
TJH2083_TX	ACACA
TJH2480_TX	AATTA
TJH3008_TX	ACGGT

Extremely important; this file sets all the parameters to run your data through iPyrad and specifies paths to find data files, etc.

Parameters (params-*.txt) file

name for your assembly & where files are outputted

paths in TACC to your data and barcode file

```
------- ipyrad params file (v.0.9.80)-------------------------------------------
                               ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
                               ## [1] [project_dir]: Project dir (made in curdir if not present)
                               ## [2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
                               ## [3] [barcodes_path]: Location of barcodes file
                               ## [4] [sorted_fastq_path]: Loca
denovo                         ## [5] [assembly_method]: Assem
                               ## [6] [reference_sequence]: Loca
pairddrad                      ## [7] [datatype]: Datatype (see docs): rad, gbs, ddrad, etc.
GAATT,                         ## [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
0.85                           ## [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
12                             ## [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, 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
```

Parameters (params-*.txt) file

```
------- ipyrad params file (v.0.9.80)-------------------------------------------
                               ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
                               ## [1] [project_dir]: Project dir (made in curdir if not present)
                               ## [2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
                               ## [3] [barcodes_path]: Location of barcodes file
                               ## [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
pairddrad                      ## [7] [datatype]: Datatype (see docs): rad, gbs, ddrad, etc.
GAATT,                         ## [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
0.85                           ## [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
12                             ## [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, 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
```

A general command in iPyrad

```
$ ipyrad –n [params file]
```

Creates a parameters file that you'll edit

```
$ ipyrad –p [params file] –s [step number]
```

Accesses the params file to run sequential steps

You can run multiple steps in the same line of code:

```
$ ipyrad –p #name -s 4567
```

Viewing progress

- Can use command:

```
$ ipyrad –p paramsfilename -r
```

```
## -r fetches informative results from currently executed steps
$ ipyrad -p params-anolis.txt -r

Summary stats of Assembly anolis
------------------------------------------------
                    state  reads_raw
punc_IBSPCRIB0361       1     250000
punc_ICST764            1     250000
punc_JFT773             1     250000
punc_MTR05978           1     250000
punc_MTR17744           1     250000
punc_MTR21545           1     250000
punc_MTR34414           1     250000
punc_MTRX1468           1     250000
punc_MTRX1478           1     250000
punc_MUFAL9635          1     250000


Full stats files
------------------------------------------------
step 1: ./anolis_s1_demultiplex_stats.txt
step 2: None
step 3: None
step 4: None
step 5: None
step 6: None
step 7: None
```

https://radcamp.github.io/AF-Biota/02_ipyrad_partI_CLI.html

Step 1. Demultiplexing files

Let's take a look at an example sequence:

```
[Annes-MacBook-Pro-2:ipsimdata eac$ gunzip -c ./pairddrad_example_R1_.fastq.gz | head -n 12
@lane1_locus0_2G_0_0 1:N:0:
CTCCAATCCTGCAGTTTAACTGTTCAAGTTGGCAAGATCAAGTCGTCCCTAGCCCCCGCGTCCGTTTTTACCTGGTCGCGGTCCCGACCCAGCTGCCCCC
+
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
@lane1_locus0_2G_0_1 1:N:0:
CTCCAATCCTGCAGTTTAACTGTTCAAGTTGGCAAGATCAAGTCGTCCCTAGCCCCCGCGTCCGTTTTTACCTGGTCGCGGTCCCGACCCAGCTGCCCCC
+
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
@lane1_locus0_2G_0_2 1:N:0:
CTCCAATCCTGCAGTTTAACTGTTCAAGTTGGCAAGATCAAGTCGTCCCTAGCCCCCGCGTCCGTTTTTACCTGGTCGCGGTCCCGACCCAGCTGCCCCC
+
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
```

```
[Annes-MacBook-Pro-2:ipsimdata eac$ head pairddrad_example_barcodes.txt
1A_0    CATCATCAT
1B_0    CCAGTGATA
1C_0    TGGCCTAGT
1D_0    GGGAAAAAC
2E_0    GTGGATATC
2F_0    AGAGCCGAG
2G_0    CTCCAATCC
2H_0    CTCACTGCA
3I_0    GGCGCATAC
3J_0    CCTTATGTC
```

After you've done this, you'll have a separate .fastq.gz file for each individual

Let's look at the data we'll be working on!

- Barcodes file (barcodes.txt)

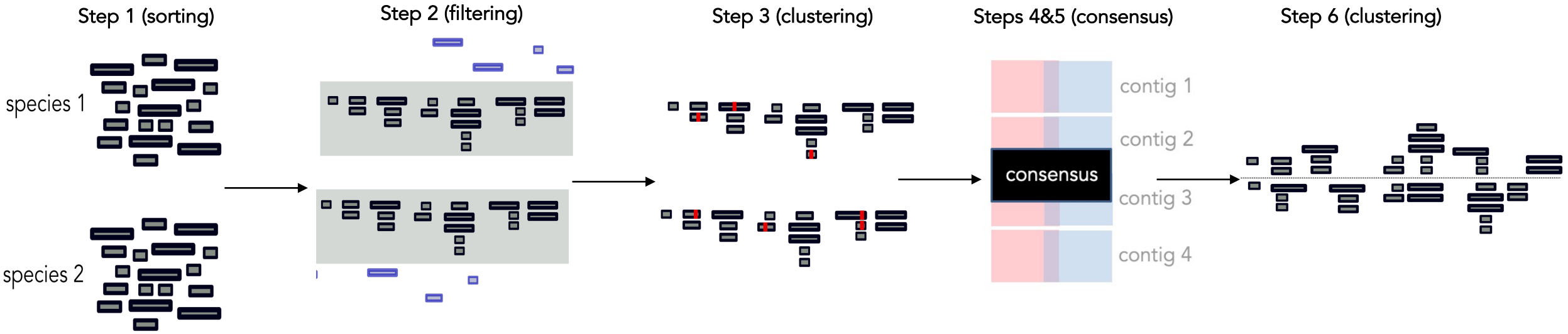
```
Rbla_SD_1        ACTGG
Rbla_SD_2        ACTTC
Rneo_Jalisco_1   ATACG
Rneo_Jalisco_2   ATGAG
Rber_Tam_1a      ATTAC
Rber_Tam_1b      CATAT
Rber_Tam_2       CGAAT
Rchi_AZ_1a       CGGCT
Rchi_AZ_1b       CGGTA
Rchi_AZ_2        CGTAC
Rsph_TX_1        CGTCG
Rsph_TX_2        CTGAT
```

- Files:

`64T64_P29_S1_L005_R1_001.fastq`

`64T64_P29_S1_L005_R2_001.fastq`

Species	Locality	Barcode file ID
Rana blairi	South Dakota, USA	Rbla_D2864
Rana blairi	South Dakota, USA	Rbla_D2865
Rana neovolcanica	Jalisco, Mexico	Rneo_T480
Rana neovolcanica	Jalisco, Mexico	Rneo_T527
Rana berlandieri	Tamaulipas, Mexico	Rber_T1113a Rber_T1113b
Rana berlandieri	Tamaulipas, Mexico	Rber_T1114
Rana chiricahuensis	Arizona, USA	Rchi_T2034a Rchi_T2034b
Rana chiricahuensis	Arizona, USA	Rchi_T2049
Rana sphenocephala	Texas, USA	Rsph_T25870
Rana sphenocephala	Texas, USA	Rsph_T26064


Step 1 (sorting)
Step 2 (filtering)
Step 3 (clustering)
Steps 4&5 (consensus)
Step 6 (clustering)
species 1
species 2
consensus
contig 1
contig 2
contig 3
contig 4