**Preprocessing ddRAD Sequences Workflow**

1. Initial quality checks
   * Use fastqc (see usage script below)
2. Remove duplicates based on DBR
   * Use ParseDBR\_ddRAD.py (available on github, see usage script below, requires Python2) <https://github.com/Eljensen/ParseDBR_ddRAD>
   * The amount of memory to request depends on how many reads are in your input fastq files. If you have ~300M reads, request 112 GB, if ~250M, request 72GB
   * If you are using Graham, the amount of time required for the job script can vary from 1 day to 3 days, depending on the type of node that runs the job. Request 3 days to be safe.
   * The command line flags are as follows:

-r read 1 fastq file

-R read 2 fastq file

-i index sequence

-e R2 enzyme sequence (should be AATT)

-n read 1 output fastq file name

-N read 2 output fastq file name

--drop file name for the list of dropped reads (e.g. dropped\_reads.txt)

-Z indicates the output files should be fastq, not fastq.gz (optional, but if you include it the job will run much faster))

-l the length of the R2 adapter anchor sequence

* + Some of the commands will depend on which DBR adapter you used.

DBR01: -i ATCACG -l 0

DBR08: -i ACTTGA -l 1

DBR10: -i TAGCTT -l 2

DBR11: -i GGCTAC -l 3

It is a good idea to run fastqc on the deduplicated fastq files, just to make sure the R2 reads all have been trimmed properly.

**FastQC Script Example**

#!/bin/bash

#SBATCH -J ddRAD\_fastqc

#SBATCH -c 1

#SBATCH --mem=520

#SBATCH -t 0-4:0:0

#SBATCH --mail-type=ALL

#SBATCH --mail-user=USERNAME@queensu.ca

#SBATCH -o fastqc\_output\_%j.o

module load fastqc

fastqc ddRAD\_R1.fastq.gz

fastqc ddRAD\_R2.fastq.gz

**ParseDBR\_ddRAD Script Example**

#!/bin/bash

#SBATCH -J dbr\_dedup

#SBATCH -c 1

#SBATCH --mem=112000

#SBATCH -t 3-1:0:0

#SBATCH --mail-type=ALL

#SBATCH --mail-user=USERNAME@queensu.ca

#SBATCH -o dedup\_output\_%j.o

python ParseDBR\_ddRAD.py -r ddRAD\_R1.fastq.gz \

-R ddRAD\_R2.fastq.gz \

-n dedup\_ddRAD\_R1.fastq -N dedup\_ddRAD\_R2.fastq -i GGCTAC -e AATT -l 3 --drop dropped.txt