Prosper Pipeline and Genotype Matrix on Hoffman2

1) Clone the prosper processing pipeline to your hoffman2 working directory https://github.com/NuttyLogic/ProsperProcessingContainer
2) Remove lines 120 to 135 from ProcessingScripts/ProcessingPipeline.py

```
120     # gather microbiome counts
121     mb_counts = get_microbiome_counts(f'{output}.dup.bam', microbiome_bed,
122                                       min_qual=int(config_kwargs['CallVariation']['MQ']))
123
124     # Biomarker Model Predictions
125     bm_vals = process_bms(cgmap_path=f'{output}.CGmap.gz',
126                           biomarkers_file=biomakers_file,
127                           cell_type_ref=cell_type_file,
128                           snp_calls_path=f'{output}.variant.bed.gz',
129                           snp_annotation_bed=snp_annotation_bed)
130     bm_vals['SampleID'] = output.split('/')[-1]
131     bm_vals['MicroBiomeCounts'] = mb_counts
132     bm_vals['SampleQC'] = qc_info
133     with open(f'{output}.bmpredictions.json', 'w') as out:
134         json.dump(bm_vals, out)
135     log.update_log(11, 'bm_predictions')
```

(if you don't want CGmap files created you can also remove lines 98-106 as well):

```
108     # Variant Calling
109
110     call_args = config_kwargs['CallVariation']
111     call_args['DB'] = bsb_db
112     call_args['I'] = f'{output}.dup.bam'
113     call_args['O'] = f'{output}.variant'
114     if probe_target_bed is not None:
115         call_args['BR'] = probe_target_bed
116
117     variant_cmd = extract_bsbolt_cmd(call_args)
118     run_subprocess(variant_cmd, output, 10, 'CallVariation', log)
```

3) Set up environment and dependencies:
    a) cd ProsperProcessingContainer
    b) Module load python
    c) pip3 install wheel pyyaml cutadapt joblib scipy git+https://github.com/NuttyLogic/BSBolt@prosper
    d) export PATH="${PATH}:/u/home/{first letter of username}/{username}/.local/bin"
    e) Module load samtools
    f) Module load bedtools
4) Generate Index for species
    a) Download reference fasta file for species
    b) Run BSBolt Index to generate index, here is the bash script I used:

```
 GNU nano 2.3.1                                              File: canFam4_bac_index.sh

#!/bin/bash
#$ -pe shared 8
#$ -l h_rt=20:00:00
#$ -l h_data=4G

echo "@@@ Starting bsb Index on integrated_genomes.fna $(date)"

. /u/local/Modules/default/init/modules.sh
module load python

python3 -m bsbolt Index -G integrated_genomes.fna.gz -DB canFam4-bac-Index 1> bsbIndex.stdout 2> bsbIndex.stderr

echo "@@@ Script reached end. $(date)"
```

5) Create line separated text file of sample names you will be processing
    a) My fastq files were formatted like this:

    PK9-31201050607829_R1_trimmed.fastq.gz
    PK9-31201050607829_R2_trimmed.fastq.gz

    So for this specific sample I would just write PK9-31201050607829 in the samples.txt file
6) Place all fastq inputs into a new directory, I called mine fastqInputs
7) Create work and launch scripts, here are mine

```
 GNU nano 2.3.1                                              File: ProsperWork.sh

#!/bin/bash
#$ -pe shared 4
#$ -l h_rt=15:00:00
#$ -l h_data=7G


while getopts f:d:o:i: option
do
 case "${option}"
 in
 f) files_to_process=${OPTARG};;
 d) file_directory=${OPTARG};;
 o) output_folder=${OPTARG};;
 i) index=${OPTARG};;

 esac
done

mkdir ${output_folder}

. /u/local/Modules/default/init/modules.sh
module load python
pip3 install wheel pyyaml cutadapt joblib scipy
pip3 install git+https://github.com/NuttyLogic/BSBolt@prosper
module load samtools
module load bedtools
export PATH="${PATH}:/u/home/a/aborujer/.local/bin"

sample=$(cat $files_to_process | head -${SGE_TASK_ID} | tail -1 )

python3 ProcessingScripts/ProcessSample.py -O ${output_folder}${sample} \
-F1 ${file_directory}${sample}_R1_trimmed.fastq.gz -F2 ${file_directory}${sample}_R2_trimmed.fastq.gz -DB ${index}  > ${output_folder}${sample}.log
```

```
 GNU nano 2.3.1                                              File: ProsperLaunch.sh

files_to_process=samples.txt

# get the number of lines in txt file
number_files=$(cat $files_to_process | wc -l)


# set variables to pass to processing script
file_directory=fastqInputs/
output_folder=processedFiles/
index=canFam4-bac-Index


# of jobs to process simultaneously
JOBS=50


# submit jobs

qsub -cwd -r no -m as -N dataProcessing -t 1-$number_files -tc $JOBS ProsperWork.sh -f $files_to_process -d $file_directory -o $output_folder  -i $index
```

8) Launch pipeline through bash scripts (./ProsperLaunch.sh)

9) To generate genotype matrix:
    a) Move your .variant.bed files into a single directory, I called mine just_Variant_files. I used this script:

```
  GNU nano 2.3.1                                                File: getVariants.sh

processed_files=samples.txt
number_files=$(cat $processed_files | wc -l)


for i in $(seq 1 $number_files)
do
    sample=$(cat $processed_files | head -$i | tail -1 )
    cd ${sample}
    cp ${sample}.variant.bed.gz ../../just_Variant_files
    cd ..
done
```

    b) You can clone my repository that has the genotype matrix method https://github.com/aborujerdpur/BSBolt-VariantMatrix-Test
    c) cd BSBolt
    d) Create filenames.txt inside BSBolt that contains the path to your variant files that you want to generate a matrix from, mine looks like this:

```
  GNU nano 2.3.1                                                File: filenames.txt

/u/scratch/a/aborujer/ProsperProcessingContainer/just_Variant_files/DogWolf00108_S84.variant.bed.gz
/u/scratch/a/aborujer/ProsperProcessingContainer/just_Variant_files/DogWolf00109_S110.variant.bed.gz
/u/scratch/a/aborujer/ProsperProcessingContainer/just_Variant_files/DogWolf00110_S105.variant.bed.gz
/u/scratch/a/aborujer/ProsperProcessingContainer/just_Variant_files/DogWolf00111_S89.variant.bed.gz
/u/scratch/a/aborujer/ProsperProcessingContainer/just_Variant_files/DogWolf00112_S88.variant.bed.gz
/u/scratch/a/aborujer/ProsperProcessingContainer/just_Variant_files/DogWolf00113_S82.variant.bed.gz
/u/scratch/a/aborujer/ProsperProcessingContainer/just_Variant_files/DogWolf00114_S95.variant.bed.gz
/u/scratch/a/aborujer/ProsperProcessingContainer/just_Variant_files/DogWolf00115_S56.variant.bed.gz
/u/scratch/a/aborujer/ProsperProcessingContainer/just_Variant_files/DogWolf00116_S77.variant.bed.gz
/u/scratch/a/aborujer/ProsperProcessingContainer/just_Variant_files/DogWolf00117_S48.variant.bed.gz
/u/scratch/a/aborujer/ProsperProcessingContainer/just_Variant_files/DogWolf00118_S58.variant.bed.gz
/u/scratch/a/aborujer/ProsperProcessingContainer/just_Variant_files/DogWolf00119_S76.variant.bed.gz
/u/scratch/a/aborujer/ProsperProcessingContainer/just_Variant_files/DogWolf00120_S102.variant.bed.gz
/u/scratch/a/aborujer/ProsperProcessingContainer/just_Variant_files/DogWolf00121_S70.variant.bed.gz
/u/scratch/a/aborujer/ProsperProcessingContainer/just_Variant_files/DogWolf00122_S45.variant.bed.gz
/u/scratch/a/aborujer/ProsperProcessingContainer/just_Variant_files/DogWolf00123_S59.variant.bed.gz
/u/scratch/a/aborujer/ProsperProcessingContainer/just_Variant_files/DogWolf00124_S107.variant.bed.gz
/u/scratch/a/aborujer/ProsperProcessingContainer/just_Variant_files/DogWolf00125_S75.variant.bed.gz
/u/scratch/a/aborujer/ProsperProcessingContainer/just_Variant_files/DogWolf00126_S103.variant.bed.gz
/u/scratch/a/aborujer/ProsperProcessingContainer/just_Variant_files/DogWolf00127_S114.variant.bed.gz
/u/scratch/a/aborujer/ProsperProcessingContainer/just_Variant_files/DogWolf00128_S93.variant.bed.gz
```

    e) Create a launch script, mine looks like this:

```
  GNU nano 2.3.1                                          File: launchBsbGenotypeMatrix.sh

#!/bin/bash
#$ -pe shared 4
#$ -l h_rt=12:00:00
#$ -l h_data=10G

echo "@@@ Starting genotype matrix generation. $(date)"

. /u/local/Modules/default/init/modules.sh
module load python

python3 -m bsbolt GenotypeMatrix -F filenames.txt -O testMatrix.txt -min-sample 0.8 -t 4 -verbose

echo "@@@ Script reached end. $(date)"
```

    f) Launch the command using ./launchBsbGenotypeMatrix.sh