##
####

(c) 2015-2017 Merly Escalona merlyescalona@uvigo.es

Phylogenomics Lab. University of Vigo.

#

Description:

=============

Pipelines for data simulation for variant calling assesement

Running @ft2.cesga.es

##
####
#!/bin/bash -l
##
####

Previous to running the wrapper I had

to set up the perl env.

< o conf mbuildpl_arg '--install_base /home/uvi/be/mef/perl'
cpan> o conf commit
cpan> q
cpan install Math::GSL
MODULE_INSTALL_PERL
##

Folder paths

##
source $HOME/vc-benchmark-cesga/src/vcs.variables.sh
simphyReplicateID=1
##

0. Folder structure

##

git clone https://merlyescalona@github.com/merlyescalona/vc-benchmark-cesga.git $HOME/vc-benchmark-cesga

mkdir $folderDATA $folderOUTPUT $folderERROR $folderINFO

##

STEP 1. SimPhyvc

##
sbatch -a $simphyReplicateID $folderJOBS/vcs.1.simphy.sh | awk '{ print $4}'
##

STEP 2. INDELible wrapper

##

After the running of SimPhy, it is necessary to run the INDELIble_wrapper

to obtain the control files for INDELible. Since, is not possible to

run it for all the configurations, it is necessary to modify the name of the

output files in order to keep track of every thing

##
####
sbatch -a $simphyReplicateID $folderJOBS/vcs.2.wrapper.sh | awk '{ print $4}'
##
####

3. INDELIBLE CALLS

##
####

Need to figure out the folder from where I'll call indelilble

Need to filter the species tree replicates that do not have ninds % 2==0

find $LUSTRE/data/ -mindepth 2 -maxdepth 2 -type d | grep ssp | sort > $HOME/vc-benchmark-cesga/files/ssp.3.indelible.folders.txt
sbatch -a 11-50 $folderJOBS/vcs.3.indelible.array.sh | awk '{ print $4}'
##
####

4. ngsphy

##
####
sbatch $folderJOBS/vcs.4.ngsphy.sh

Possible - Generate Folder structure

for art

##
####

4. 0

#---

Compress gene tree files of the replicates into a single gtrees file.

The file will be a tab separated file with the id and the gtree

##
####
replicateID=$(printf "%05g" $simphyReplicateID)
replicateFOLDER="$LUSTRE/data/$pipelinesName.$replicateID"
for item in $()
##
####

4.1 ART

##
####

Need to split the command file. This is because the slurm sysmtem does not

allow me to launch jobs over 1K.

##

Moved info to triploid

<<RSYNC

This takes like an hour

rsync -rP $LUSTRE/data/ngsphy.data/NGSphy_ssp.00002/ merly@triploid.uvigo.es:/home/merly/data/NGSphy_ssp.00002

Had to change the names of the paths for the files that were used, since I'm no longer at cesga

cat ssp.00002.sh | sed 's/\/mnt\/lustre\/scratch\/home\/uvi\/be\/mef\/data\/ngsphy.data/\/home\/merly\/data/g' | sed 's/\/home\/uvi\/be\/mef\/vc-benchmark-cesga\/files/\/home\/merly\/csNGSProfile/g' > ssp.00002.triploid.sh

Way better and faster to run on triploid sequentially

RSYNC
module load gcc/5.2.0 bio/art/050616
triploidART="/home/merly/data/NGSphy_ssp.00002/scripts/ssp.00002.triploid.sh"
for item in $(seq 500001 537000); do
command=$(awk -v x=$item 'NR==x' $triploidART)
echo -e "$item"

$command
done

##
####
<<SPLIT_COMMANDS

If staying at LUSTRE, LUSTRE does not allow to launch more than 1000 jobs.

So, I had to split the files and wait for all the jobs to finish to launch

the following 1000 jobs.

split -l 1000 -d -a 3 ssp.00002.sh ssp.00002.art.commands.
for file in $(ls ssp.00002.art.commands*); do mv $file "$file.sh"; done for item in $(find /mnt/lustre/scratch/home/uvi/be/mef/data/ngsphy.data/NGSphy_ssp.00002/scripts -name "ssp.00002.art.commands*" | sort); do
sbatch -a 1-1000 $HOME/vc-benchmark-cesga/jobs/vcs.5.art.param.sh $item;
done
SPLIT_COMMANDS
##
####

5. Reference Loci Selection

##
####
refselector -p -ip data outgroup -op -o outgroup -m 0 -nsize 250
refselector -p -ip data ringroup -op -o rndingroup -m 2 -nsize 250
##

####

6. Organize and compress read files (ssp.regroup.ngs.individuals)

#---
replicateNum=$1
pipelinesName="ssp"
replicateID="$(printf "%0${replicatesNumDigits}g" $replicateID)"
replicatesNumDigits=5
ngsphyReplicatePath="$LUSTRE/data/ngsphy.data/NGSphy_${pipelinesName}.${replicateID}"

reads/1/03/testwsimphy_1_03_data_7_R2.fq

numReplicates=10

NGSMODE=("PE150OWN" "PE150DFLT" "PE250DFLT" "SE150DFLT" "SE250DFLT")

MODE=("PAIRED", "SINGLE")

NGSMODE="PE150OWN"
MODE="PAIRED"
for replicateST in $(seq 1 $numReplicates); do
numIndividuals=$(cat
$ngsphyReplicatePath/ind_labels/${pipelinesName}.${replicateST}.individuals.csv | tail -n+2 | wc
-l)
let numIndividuals=numIndividuals-1
mkdir -p $ngsphyReplicatePath/$NGSMODE/$replicateST
for individualID in $(seq 0 $numIndividuals); do
fqFilesR1=($(find $ngsphyReplicatePath/reads/$replicateST -name "*${individualID}_R1.fq")) for*

item in ${fqFilesR1[@]}; do cat $item >>
$ngsphyReplicatePath/$NGSMODE/$replicateST/${pipelinesName}${replicateST}${individualID}
_R1.fq gzip $item done gzip
$ngsphyReplicatePath/$NGSMODE/$replicateST/${pipelinesName}${replicateST}${individualID}
_R1.fq if [[MODE -eq "PAIRED"]]; then fqFilesR2=($(find
$ngsphyReplicatePath/reads/$replicateST -name "${individualID}*R2.fq")) for item in*
${fqFilesR2[@]}; do cat $item >>
$ngsphyReplicatePath/$NGSMODE/$replicateST/${pipelinesName}${replicateST}*${individualID}*
_R2.fq gzip $item done gzip
$ngsphyReplicatePath/$NGSMODE/$replicateST/${pipelinesName}${replicateST}_${individualID
}_R2.fq
fi
done
done

rm -f reads

##
####

6. stats

##
####

STEP 9. FASTQC

##
####

fqFiles="$fqReadsFolder/${pipelinesName}.allfiles.fastq"
find $fqReadsFolder -name *.fq | xargs cat > $fqFiles

st=1
echo -e "#! /bin/bash
#$ -o $outputFolder/$pipelinesName.8.$st.o
#$ -e $outputFolder/$pipelinesName.8.$st.e

```
#$ -N $pipelinesName.8.$st

INPUTBASE=$(basename $fqFiles .fastq)

cd $qcFolder/\$INPUTBASE
$fastqc $fqFiles -o $qcFolder/$INPUTBASE

"> $scriptsFolder/$pipelinesName.8.$st.sh
qsub -l num_proc=1,s_rt=0□00,s_vmem=2G,h_fsize=1G,arch=haswell
$scriptsFolder/$pipelinesName.8.$st.sh
```