May 07, 2021

NGS Protocol Automation

erick.rios. [1]

[1]California State University, Northridge

erick.rios. : Budding Data Scientist

Other This protocol is published without a DOI.

NoviceCodingErick

erick.rios.

ABSTRACT

Automation_v1_2.sh
Written by Erick (02/05/2021)

Hello to "v1_2".

The current protocol:

"A simplified workflow for the analysis of whole-genome sequencing data from mutant lines
with an application to the nematode Pristionchus pacificus"

written by Christian Rödelsperger.

Uses BWA, samtools, and bcftools which are tools or packages that can be readily downloaded and installed. These tools are "linux/Mac based" to put it simply.

Windows users need to download an "emulator" of linux terminal such as "ubuntu 20.04" from sources such as the Microsoft store. (I use ubuntu 20.04.)

Installing Tools

For Windows 10:
1. Install “ubuntu 20.04 LTS” (free) from the Microsoft store
2. Once installed correctly
3. Open windows command shell and type “bash”; it may take a few moments for first time running.
4. $ sudo apt-get install bwa
5. $ sudo apt-get install samtools
6. $ sudo apt-get install bcftools
7. You are now free to continue.

For Mac users:
1. Open terminal/console
2. $ brew install bwa
3. $ brew install samtools
4. $ brew install bcftools
5. You are now free to continue.

Some computer systems may require admin or even root permissions in order to run the bash script.

The simplest way ensure the script does not fail mid-way:

>sudo -i sudo -s

Bash script can now be ran with full permissions.

Initiating the bash script

>./Automation_v1_2.sh

PROTOCOL CITATION

erick.rios. 2021. NGS Protocol Automation. **protocols.io**
https://protocols.io/view/ngs-protocol-automation-butfnwjn

CREATED

May 06, 2021

LAST MODIFIED

May 07, 2021

PROTOCOL INTEGER ID

49735

DISCLAIMER:

1
```
ENTIRE=n
 ALIGNMENTS=n
 REGIONS=n
 MUTATIONS=n
 INDEX=n
 echo -n "Perform ENTIRE Protocol? (Y/n): "
read ENTIRE
if [[ “$ENTIRE” == Y ]]; then
 echo “WILL perform ENTIRE protocol.”
 fi
if [ $ENTIRE != n -a $ENTIRE != Y ]; then
 echo “You said $ENTIRE, assuming ‘NO’”
 ENTIRE=n
 fi
 ###################################
 if [[ “$ENTIRE” == n ]]; then
 echo -n "Perform candidate regions protocol? (Y/n): "
 read REGIONS
if [ $REGIONS != n -a $REGIONS != Y ]; then
 echo “You said $REGIONS, assuming ‘NO’”
 REGIONS=n
 fi
echo -n "Perform candidate mutations protocol? (Y/n): "
 read MUTATIONS
 if [ $MUTATIONS != n -a $MUTATIONS != Y ]; then
 echo “You said $MUTATIONS, assuming ‘NO’”
 MUTATIONS=n
 fi
 fi
 ###################################
 if [ $ENTIRE == Y -o $ALIGNMENTS == Y -o $REGIONS == Y -o $MUTATIONS == Y ]; then
 echo -n "Enter representative genome (include file extension): "
 read GENOME
 echo -n "Index your representative genome? (Y/n): "
 read INDEX
 if [ $INDEX != Y -a $INDEX != n ]; then
 echo “You said $INDEX, assuming ‘NO’”
 echo “Will NOT index genome.”
```

```
INDEX=n
fi
fi
###################################
if [ "$ENTIRE" == Y -o $MUTATIONS == Y -o $ALIGNMENTS == Y ]; then
echo -n "Enter suppressor read 1 (include file extension): "
read SUPP_1
echo -n "Enter suppressor read 2 (include file extension): "
read SUPP_2
echo -n "Enter mutant read 1 (include file extension): "
read MUT_1
echo -n "Enter mutant read 2 (include file extension): "
read MUT_2
fi
if [ "$ENTIRE" == Y -o $MUTATIONS == Y ]; then
echo -n "Candidate mutations; Enter annotation file (include file extension): "
read ANNOTATION
fi
if [ "$ENTIRE" == Y -o $REGIONS == Y -o $ALIGNMENTS == Y ]; then
echo -n "Enter wild read 1 (include file extension): "
read WILD_1
echo -n "Enter wild read 2 (include file extension): "
read WILD_2
echo -n "Enter map read 1 (include file extension): "
read MAP_1
echo -n "Enter map read 2 (include file extension): "
read MAP_2
if [ "$ENTIRE" == Y -o $REGIONS == Y ]; then
echo -n "Candidate regions; Enter 'Source Code 1' perl script (include file extension): "
read $PERL
fi
fi
###################################
if [ $ENTIRE == Y -o $ALIGNMENTS == Y -o $REGIONS == Y -o $MUTATIONS == Y ]; then
echo -n "How many CPU threads to dedicate for alignments? (Minimum: 1; Max: 5): "
read THREAD
fi
#########################################################################################
#####
if [[ "$INDEX" == Y ]];then
bwa index $GENOME
fi
######################## REGIONS
#########################################################
if [ "$ENTIRE" == Y -o $REGIONS == Y ]; then
bwa mem -t THREAD -o WILD.sam $GENOME $WILD_1 $WILD_2
bwa mem -t THREAD -o MAP.sam $GENOME $MAP_1 $MAP_2
samtools view -S -b MAP.sam > MAP_unsorted.bam
samtools view -S -b WILD.sam > WILD_unsorted.bam
samtools sort -o MAP.bam MAP_unsorted.bam
samtools sort -o WILD.bam WILD_unsorted.bam
samtools index MAP.bam
samtools index WILD.bam
rm MAP_unsorted.bam MAP.sam
rm WILD_unsorted.bam WILD.sam
bcftools mpileup -Ou -f $GENOME WILD.bam | bcftools call -mv -Ov -o WILD.vcf
awk '{if($6>100) print}' WILD.vcf | grep -v INDEL |awk '{print $1 "\t" $2}' > positions.txt
samtools mpileup -f $GENOME -l positions.txt MAP.bam > pileup_data.txt
perl count_allele_frequencies.pl pileup_data.txt > mapping_data.txt
echo "Candidate regions finished; Look for 'mapping_data.txt' file."
fi
```

```
################################ MUTATIONS
##########################################
if [ "$ENTIRE" == Y -o $MUTATIONS == Y ]; then
bwa mem -t THREAD -o SUPP.sam $GENOME $SUPP_1 $SUPP_2
bwa mem -t THREAD -o MUT.sam $GENOME $MUT_1 $MUT_2
samtools view -S -b SUPP.sam > SUPP_unsorted.bam
samtools view -S -b MUT.sam > MUT_unsorted.bam
samtools sort -o SUPP.bam SUPP_unsorted.bam
samtools sort -o MUT.bam MUT_unsorted.bam
samtools index SUPP.bam
samtools index MUT.bam
rm SUPP_unsorted.bam SUPP.sam
rm MUT_unsorted.bam MUT.sam
bcftools mpileup -Ou -f $GENOME SUPP.bam | bcftools call -mv -Ou | bcftools view -i '%QUAL>=20' -Oz > SUPP.vcf.gz
bcftools mpileup -Ou -f $GENOME MUT.bam | bcftools call -mv -Ou | bcftools view -i '%QUAL>=20' -Oz > MUT.vcf.gz
bcftools index SUPP.vcf.gz
bcftools index MUT.vcf.gz
bcftools filter -O z SUPP.vcf.gz > SUPP_filtered1.vcf.gz
bcftools index SUPP_filtered1.vcf.gz
bcftools isec -C SUPP_filtered1.vcf.gz MUT.vcf.gz > pos.txt
bcftools filter -O z -R pos.txt SUPP_filtered1.vcf.gz > SUPP_filtered2.vcf.gz
bcftools sort -O z SUPP_filtered2.vcf.gz > SUPP_filtered2_sorted.vcf.gz
bcftools index SUPP_filtered2_sorted.vcf.gz
bcftools csq -p s -f $GENOME -g $ANNOTATION SUPP_filtered2_sorted.vcf.gz >  SUPP_variants_annotated.vcf
grep missense SUPP_variants_annotated.vcf > Candidate_Mut.txt
grep stop_gained SUPP_variants_annotated.vcf >> Candidate_Mut.txt
grep synonymous SUPP_variants_annotated.vcf >> Candidate_Mut.txt
grep INDEL SUPP_variants_annotated.vcf >> Candidate_Mut.txt
echo "Candidate mutations protocol finished; Look for 'Candidate_Mut.txt' file"
fi
################################################################################
#####
if [[ $ENTIRE == n ]]; then
if [[ "$REGIONS" == n ]]; then
echo "Candidate regions protocol was not performed."
fi
if [[ "$MUTATIONS" == n ]]; then
echo "Candidate mutations protocol was not performed."
fi
fi
if [ $ENTIRE == n -a $REGIONS == n -a $MUTATIONS == n -a $INDEX == n -a $ALIGNMENTS == n ];then
echo "Nothing was accomplished this day."
else
echo "Something was accomplished!"
fi
```