README For Variant Calling by VarScan on TACC (Stampede2)

Written by : Taslima Haque
Last Modified: 03/02/2023

Contact : taslima@utexas.edu

Expected parametes to set:

Here is the example header of each script which expect the following variables:
(Please set this with our own personal work directory on Stampede2)

```
refDir=/some/path/ #directory where the reference genome file will be
ref=/some/file # Name of reference genome file
outDir=/some/path #output directory. It must be created before running the scrip
met=/some/path/Pvirg_48_midwest_metadata_mod.csv
TMP=/some/path/Temp
```

Here is the sample of Meta file that is tab separated with fields of unique
sample ID, sample name, and other meta info:

```
IELF,J020.A,Natural Collection,Grown from GRIN Seed Packet In Austin,\
Florida,27.197548,-80.252826,Unknown,Midwest,Midwest
IELJ,J036.A,Natural Collection,Grown from GRIN Seed Packet In Austin,\
South Dakota,44.520755,-99.200387,Upland,Midwest,Midwest
IELK,J037.A,Cultivar,Grown from GRIN Seed Packet In Austin,\
South Dakota,46.388289,-100.98355,Upland,Midwest,Midwest
IEMC,J254.A,Natural Collection,Supplied by Mike Cassler,\
Wisconsin,42.496,-87.808,Unknown,Midwest,Midwest
```

Tools requried:

```
  -- bwa-mem2 (install locally)
  -- samtools
  -- picard
  -- varscan (install locally)
  -- bcftools (install locally)
```

The pipe expect the output directory is the 1st level directory that already exists &
have following directories:

```
ls  outDir/

          RAW
          MAP_SORTED
          MAP_SORTED_DEDUP
          VarScan
          VarScan_Filter
          MergedVCF
```

Each of these steps create a specific param file which you need to run by slurm.
The slurm file is provide with the pipe named as "slurm.sh"

Step 1: Map and filter

```
sh 01-BWA2-Mapping-Filter-Sort.sh
```

This step will generate a param file named as "bwa2-sort.param". We are going to run this on "skx-normal" queue with 12 hours limit while each job will take one entire node. Using "skx-normal" instead on "normal" queue will be quick due to faster clock speed.The max it took me to run the largest sample was 8 hours therefore 12 hours should be a safe limit. The following command should work for 48 samples:

```
sbatch -t 12:00:00 -N 48 -n 48 --ntasks-per-node=1 -p skx-normal slurm.sh \
bwa2-sort.param
```

Step 2: Remove Duplicates

```
sh 02-Dedup.sh
```

This step will generate a param file named as "dedup.param". We are going to run this on "normal" queue with 15 hours limit while each job will take one entire node. This step is limited mostly for memory than CPU therefore on "normal" queue it would be cheaper and not much less faster than "skx-normal" queue. The max it took me to run the largest sample was 11 hours therefore 15 hours should be a safe limit. The following command should work for 48 samples:

```
sbatch -t 15:00:00 -N 48 -n 48 --ntasks-per-node=1 -p normal slurm.sh \
dedup.param
```

Step 3: Call variants

```
sh 03-Samtools-Varscan.sh
```

This step will generate a param file named as "varscan.param". We are going to run this on "long" queue with 120 hours limit. We will run 6 jobs on a single node. The max it took me to run the largest sample was 62 hours but we will set it to the max upper limit. The following command should work for 48 samples:

```
sbatch -t 120:00:00 -N 8 -n 48 --ntasks-per-node=6 -p long slurm.sh \
varscan.param
```

Step 4: Filter variants

```
sh 04-VCFFilter-Rename.sh
```

This step will generate a param file named as "filvcf.param". We are going to run this on "normal" queue with 9 hours limit. We will run 8 jobs on a single node. The max it took me to run the largest sample was 6 hours therefore 9 hours should be a safe limit. The following command should work for 48 samples:

```
sbatch -t 09:00:00 -N 6 -n 48 --ntasks-per-node=8 -p normal slurm.sh \
filvcf.param
```

Step 5: Merge variants

```
sh 05-MergeVCF.sh
```

This step will generate three param file named as "bcfindex1.param", "bcfmerge.param", and "bcfindex1.param". This entirely depends on the number of samples we are merging and may require optimization