FiSSAtVcf pipeline (Filter Snps and Samples,Add tags in the VCF file)

FiSSAtVcf is a Snakemake pipeline that can identify and remove unrelated samples, filter SNPs and add several customized "INFO" tags in the VCF files. The pipeline combines several open-souce tools and can also be run in parallel, therefore, the execution is relatively fast. Note that the pipeline has been designed specifically for Linux OS.

Utilities

FiSSAtVcf will remove unrelated samples, monomorphic SNPs and add the following tags in INFO field after its computation:

- alternate allele count (AC)
- alternate allele frequency (AF)
- minor allele frequency (MAF)
- total number of alleles (AN)
- average depth (AVG_DP), i.e. average value across FORMAT DP values
- number of heterozygous alternate allele carriers (N_ALT_HET), i.e. with genotypes 0/1 or 1/0
- number of homozygous alternate allele carriers (N_ALT_HOM), i.e. with genotype 1/1
- average depth's percentile value (AVG_DP_P),note that we defined percentile value as the percentage of SNPs with average depth less than that of a particular SNP.

Dependencies

FiSSAtVcf uses the following open-source tools/python modules:

- Snakemake- tested using 5.31.1 version.
- bcftools - tested using 1.9 version
- plink - tested using 1.9 version
- KING - tested using 2.2.5 version
- Fastindep
- python-tested using 3.9.1, but should work with any 3.x version
- conda-tested using 4.9.2 version.
- pysam- tested on 0.16.0.1

Installation

The installation instructions for above-mentioned tools are given on their respective home pages. In general, I recommend to use conda package manager for installation of Snakemake, bcftools and pysam (as the depdencies will be automatically taken care of). To install conda, refer to the instruction given on this page: https://docs.conda.io/projects/conda/en/latest/user-guide/install/linux.html. However, note that Fastindep should not be installed from Conda. I recommend to install it from its source codes following these steps:

```
git clone https://github.com/faustovrz/FastIndep.git
g++ -v -O -Wall main.cpp DataMethods.cpp -o fastindep
```

Plink and King doest not need to be compiled as its binary executable can directly be downloaded from its website.

Usage instructions

The pipeline requires a configuration file in Json format (example given below)

```
{
    "tools": {
        "PLINK": "/home/maulik/software/populationGenomics/Plink/plink",
        "KING": "/home/maulik/software/populationGenomics/King/king",
        "fastIndep":"/home/maulik/software/populationGenomics/FastIndep/fastindep"
    },
    "base_out": {
        "inputDir":"/home/maulik/skills_test_1/input/",
        "outputDir": "/home/maulik/skills_test_1/output/"
    },
    "params":{
        "chrmList":[19,20,21,22],
        "kinThreshold":"0.0442",
        "runs":"5",
        "seeds":"2345",
        "threads":"4"
    }
}
```

- Under the "tools" dictionary, replace the path of the respective program with the installation path of that program in your environment. The path should also include the name of the binary executable as well.
- Under the "base_out" dictionary, the "inputDir" should include the path of the input directory containing the zipped vcf files. Note that the name of the zipped vcf file should follow this nomenclature (for example): for chromosome 19, the file name should be "19.vcf.gz" and chromosome name in the vcf file should be "chr19". The "outputDir" should include the path to the output directory where the final results will be stored.
- Under the "params" dictionary, "chrmList" key should contains the list of chromosomes (only integers are allowed in this list). In "kinThreshold" key, enter the value (as string) of Kinship threshold. The pairs of individuals which have the kinship value (as calculated by King tool) above this threshold will be considered as related. The "runs" and "seeds" keys represent the parameters of Fastindep, refer to its documentation for further details. "Threads" key represent the number of cores to allot to run this pipeline.

Save this configuration file as "config.json".

Once the dependency is successfully installed and configuration file is ready, run the command like this:

```
$ snakemake --snakefile FiSSAtVcf.py --cores 4
```

For optimal use, keep the number of cores equal to the value specified in "config.json"

The resulting vcf files should be generated in the directory set in "outputDir" parameter of the config file. There should be one filtered file for each chromosome with suffix "unrelatedSamplesWithTags.vcf.gz". Additionally, "UnrelatedSamples.txt" will also be generated; this file contains the name of unrelated samples included in the filtered vcf files.

How does FiSSAtVcf works?

Refer to the "Steps.pdf" to view the overall schematic of the pipeline. In general, the working of FiSSAtVcf can be divided in following steps:

- Step 1: bcftools is used to concatenate the vcf files.
- Step 2: Plink is used to convert the merged vcf to binary format as required by KING.
- Step 3: KING is used to calculate kinship coefficient between every possible individul-pairs.
- Step 4: Fastindep is used to select the unrelated individuals.
- Step 5: bcftools is used to recode the vcf files keeping only the unrelated individuals selected in the previous step.
- Step 6: A customized python code embedded in the script will remove the SNPs with monomorphic varints and add (and compute) these customized tags in the INFO field: alternate allele frequency (AF), minor allele frequency (MAF),number of heterozygous alternate allele carriers (N_ALT_HET),number of homozygous alternate allele carriers (N_ALT_HOM), average depth's percentile value (AVG_DP_P).

Visualization of Directed Acyclic Graph (DAG) generated by Snakemake is included in "Steps.pdf" document.

License