NGSCheckMate

Version 1.0

1. Introduction

NGSCheckMate is a software package for identifying next generation sequencing (NGS) data files from the same individual. It analyzes various types of NGS data files including (but not limited to) whole genome sequencing (WGS), whole exome sequencing (WES), RNA-seq, ChIP-seq, and targeted sequencing of various depths. Data types can be mixed (e.g. WES and RNA-seq, or RNA-seq and ChIP-seq). It takes BAM (reads aligned to the genome), VCF (variants) or FASTQ (unaligned reads) files as input. NGSCheckMate uses depth-dependent correlation models of allele fractions of known single-nucleotide polymorphisms (SNPs) to identify samples from the same individual. Our alignment-free module is fast (e.g., less than one minute for RNA-seq using a single core) and we recommend it for a quick initial quality check, before pooling / aligning sequenced reads. The BAM and VCF modules can be used after the alignment and variant calling steps, respectively, to ensure correct sample annotation before further downstream analysis. Currently, it works only for human data.

2. Requirement

1. Software environment

- Unix/Linux System
- Python 2.6 or above
- R 3.1 or above (required to generate a PDF of sample clustering dendrogram and a xgmml graphical output for sample clustering; see Output and Supporting scripts)

For the BAM module (your input is in bam format), you need the following in addition.

- samtools (tested on version 0.1.19 and 1.3.1)
- bcftools 0.1.19 (a utility program included in samtools)

```
# How to install samtools
# For example, download samtools version 0.1.19 from
# https://sourceforge.net/projects/samtools/files/samtools/0.1.19/
tar xvf samtools-0.1.19.tar.bz2
cd samtools-0.1.19
make
```

2. Additional files

For the BAM module,

- Human reference genome FASTA file (hg19 or GRCh37)

- A bed file (.bed) that lists the locations of selected SNPs (included in the package)

For the VCF module, (VCF and GVCF generated by samtools or GATK are supported)

- A bed file (.bed) that lists the locations of selected SNPs (included in the package)

For the FASTQ module,

- A binary pattern file (.pt) that lists the flanking sequences of selected SNPs (included in the package)

3. Installation

1. Downloading NGSCheckMate

```
cd <installation_dir>
git clone https://github.com/parklab/NGSCheckMate.git
## set NCM_HOME according to you shell environment
## for example, when using bash, add the following in your .bashrc
export NCM_HOME=<installation_dir>/NGSCheckMate
```

2. Configuration (required only for the BAM module)

If your input is BAM/VCF files, add the following lines in your ncm.conf file in the package directory. If your input is FASTQ files, you can skip this step.

```
REF=<path for the reference FASTA file >
SAMTOOLS=<path for samtools>
BCFTOOLS=<path for bcftools>
```

3. Compilation of C codes

All the binary files are provided in the package, so in general there is no need to compile the source codes and you can skip this step. In case you need to do so, use *make* to create the binary files.

ngscheckmate_fastq : the main C program of the FASTQ module

```
cd ngscheckmate_fastq-source
make
cp ngscheckmate_fastq ../
```

patterngenerator : for creating your own pattern file for the FASTQ module)

```
cd patterngenerator
```

```
make
```

4. Usage

NGSCheckMate supports three types of input files: BAM, VCF and FASTQ. Each module is executed by a Python script. You can also run the FASTQ module using a C program in addition to a Python script.

1. BAM/VCF mode

Usage: python ncm.py <-B | -V> <–d INPUT_DIR | -l INPUT_LIST_FILE> <-bed BED_FILE>
<–O OUTPUT_DIR> [options]

Required arguments

-B | -V	A flag that indicates an input file type (B: BAM, V: VCF)
-d DIR	A directory that contains input files
or	
-l FILE	A text file that lists input files and sample names (one per line; see *Input file format*)
-bed FILE	A bed format file that lists the locations of selected SNPs (included in the package) SNP/SNP_GRCh37_hg19_wChr.bed if your reference genome fasta file contains ‘chr’ in a chromosome name (e.g. ‘chr10’). SNP/SNP_GRCh37_hg19_woChr.bed otherwise. Either file works for the VCF mode.
-O DIR	An output directory

Optional arguments

-N PREFIX	A prefix of output files (default: *output*)
-f	Use strict VAF correlation cutoffs. Recommended when your data may include related individuals (parents-child, siblings)
-nz	Use non-zero mean depth of target loci as reference correlation. (default: Use mean depth of all target loci)

2. Speed up to analyze multiple large BAM files

You may need to analyze a large number of large BAM files. For example, you may want to identify the proper pairing of 100 cancer WGS data with their matched blood WGS data sequenced at high depth. In this case, it would take a long time to run NGSCheckMate on the set of BAM files, and we recommend the following procedures.

STEP1: Generate a VCF file for each BAM file as follows. This step can be parallelized depending on your computing system. For example, the LSF-based system can perform this step in parallel using ‘bsub’ command.

```
# an example for generating sample.vcf from sample.bam mapped to hg19

samtools mpileup –I –uf hg19.fasta –l SNP_GRCh37_hg19_woChr.bed sample.bam |
bcftools view –cg - > sample.vcf
```

STEP2: Run NGSCheckMate on the set of VCF files as input.

```
python ncm.py -V …
```

3. FASTQ mode

Usage: python ncm_fastq.py <-l INPUT_LIST_FILE> <-pt PT_FILE> <–O OUTPUT_DIR> [options]

Required arguments

-l FILE A text file that lists input fastq (or fastq.gz) files and sample names (one per line; see *Input file format*)

-pt FILE A binary pattern file (.pt) that lists flanking sequences of selected SNPs (included in the package; SNP/SNP.pt)

-O DIR An output directory

Optional arguments

-N PREFIX A prefix for output files (default: “output”)

-f Use strict VAF correlation cutoffs. Recommended when your data may include related individuals (parents-child, siblings)

-nz Use non-zero mean depth of target loci as reference correlation (default: Use mean depth of all target loci)

-s FLOAT The read subsampling rate (default: 1.0)

or

-d INT The target depth for read subsampling. NGSCheckMate calculates a subsampling rate based on this target depth.

-R INT The length of the genomic region with read mapping (default: 3x10^9) used to compute subsampling rate. If your data is NOT human WGS and you use the *-d* option, it is highly recommended that you specify this value. For instance, if your data is human RNA-seq, the genomic length with read mapping is ~3% of the human genome (1x10^8).

-L INT The length of the flanking sequences of the SNPs (default: 21bp). It is not recommended that you change this value unless you create your own pattern file (.pt) with a different length. See *Supporting Scripts* for how to generate your own pattern file.

-p INT The number of threads (default: 1)

4. FASTQ mode (alternative way)

A C program, ngscheckmate_fastq, can be directly called to generate a VAF file from one FASTQ file (single-end sequencing) or two FASTQ files(paired-end sequencing). Then, another script, vaf_ncm.py is used to read a set of VAF files to complete the downstream analysis. When you need to analyze many FASTQ files, the first VAF file generation using ngscheckmate_fastq can be parallelized.

ngscheckmate_fastq

Usage: ngscheckmate_fastq <-1 FASTQ_FILE1> [-2 FASTQ_FILE2] <PT_FILE (.pt)> [options] >
_output.vaf

Required arguments

-1, --fastq1 FILE FASTQ file for single-end or the first FASTQ file for paired-end. File can be gzipped (auto-detect).

PT_FILE A binary pattern file (.pt) that lists flanking sequences of selected SNPs (included in the package; SNP/SNP.pt)

Optional arguments

-2, --fastq2 FILE The second FASTQ file for paired-end. File can be gzipped (auto-detect)

-s, --ss FLOAT The read subsampling rate (default: 1.0)
or
-d, --depth INT The target depth for read subsampling. NGSCheckMate calculates a subsampling rate based on this target depth.

-R, --reference_length INT The length of the genomic region with read mapping (default: 3x10^9) to compute a subsampling rate. If your data is NOT human WGS and you use the *-d* option, it is highly recommended that you specify this value. For instance, if your data is human RNA-seq, the genomic length with read mapping is ~3% of the human genome (1x10^8).

-L, --pattern_length INT The length of flanking sequences of SNPs (default: 21bp). It is recommended not to change this value unless you create your own pattern file (.pt) with a different length. see *Supporting Scripts* for how to generate your own pattern file.

-p, --maxthread INT The number of threads to use (default : 1)

vaf_ncm.py

Usage: python vaf_ncm.py -f -I <INPUT_DIR> -O <OUTPUT_DIR > <-N PREFIX>

-I DIR Input directory that contains the output VAF files of ngscheckmate_fastq

-O DIR Output directory

-N PREFIX Ouput file prefix

5. Input file list format

1. BAM/VCF mode

The input file that lists input BAM or VCF files (-l) needs to list one file name per line (single column).

2. FASTQ mode

The input file that lists input FASTQ files (-l) should follow the format below.

Paired-end data

FASTQ_FILE1 (tab) FASTQ_FILE2 (tab) SAMPLE_NAME (\n)

Single-end data

FASTQ_FILE1(tab) SAMPLE_NAME (\n)

6. Examples

1. Test sample pairing using BAM input

python ncm.py -B -f -d /data/wgs_download/LUAD/ -O LUAD_WGS/ -N LUAD -bed SNP/SNP_GRCh37_hg19_woChr.bed

2. Test sample pairing using VCF input

python ncm.py -V -f -d /data/wgs_download/LUAD/ -O LUAD_WGS/ -N LUAD -bed SNP/SNP_GRCh37_hg19_woChr.bed

3. Test sample pairing using FASTQ input

python ncm_fastq.py -l fastq_list.txt -O output -N ChIP_batch -p 4 -pt SNP/SNP.pt

7. Output

1. PREFIX_all.txt

This output file lists both matched and unmatched sample pairs with VAF correlation coefficients and representative sequencing depths.

Format

Sample1 (tab) matched/unmatched (tab) Sample2 (tab) Correlation (tab) Depth

6216-01A	matched	6216-10A	0.7957	4.9
6216-01A	unmatched	6324-10A	0.2153	15

2. PREFIX_matched.txt

This output file lists sample pairs that were predicted to be matched based on our depth-dependent VAF correlation model.

Format

Sample1 (tab) matched_or_unmatched (tab) Sample2 (tab) Correlation (tab) Depth

6216-01A	unmatched	6216-10A	0.7957	4.9

3. PREFIX.pdf

This pdf file shows a dendrogram image of hierarchical clustering of samples based on VAF correlation coefficients.

8. Supporting scripts

1. Patterngenerator

The set of scripts in the patterngenerator folder in the package generate the .pt file used by the FSTQ module, in cases when the user wants to generate a custom .pt file. It requires a bed file containing a set of SNP positions, a genome reference file (both FASTA and bowtie 1 index) and the bowtie alignment program (http://bowtie-bio.sourceforge.net/index.shtml).

```
Usage: makesnvpattern.pl bedfile genomefasta genome(bowtie)index outdir
outprefix
```

2. Graph generator (Rscript)

This script with a set of xgmml templates is used for generating a graph representing matching files as connected nodes. The output format is in .xgmml, which can be read by Cytoscape.

```
source("graph/ngscheckmate2xgmml.R")

create.xgmml.from.ngscheckmateout(label.file,ngscheckmateoutput.file,output.x
gmml)
```

Label file: a tab-delimited text file containing a BAM file name (1st column), an individual identifier (2nd column) and optionally, a file identifier (3rd column) for each line. An individual identifier must be unique to a subject (e.g. both tumor and normal samples from the same individual must have the same individual identifier). A file identifier must be unique to a file name.

ngscheckmateoutput.file: the output text file of NGSCheckMate. It is a tab-delimited text file containing two BAM file names (1st and 2nd columns), VAF correlation (3rd column) and average depth (4th column). It may contain either all pairs or matched pairs, depending on the option used to run NGSCheckMate. Both options may be used to run this program.

Sample label file (sample.label.txt) and ngscheckmateouput.file (sample.input.txt) can be found in the subdirectory graph/.