ETCHING

Version 1.3.6c

Efficient Detection of Chromosomal Rearrangements Using a Scalable k-mer Database of Multiple Reference Genomes and Variations

ETCHING takes about 3 hours for WGS data with 30X normal and 50X tumor on 30 threads on DELL 930 server.
You can also find codes, k-mer set, and DEMO files in our website.

http://big.hanyang.ac.kr/ETCHING/

The demo is complete within 10 min on a desktop (AMD Ryzen 7 3700X 8-Core Processor).

Change history of recent versions

1.3.6

c. Virtual environment is implemented to solve dependencies. Simple installation guide.

b. etching debug (line 882). Indentation error fixed (Sorter/scorer_XGBoost). README updated.

a. File names of final result modified

v1.3.6

--target-filter and --miscall-kmer-cutoff options were added.

v1.3.5

Bug fixed (etching line 1283)

See CHANGE.md for older updates.

Requirements

System

- 64-bit LINUX with >=32GB RAM (at least >=16GB).
- Tested on Fedora workstation, Centos, and Ubuntu

Software

- Required to compile
  - gcc, g++ (>=4.7.0), make, Python3 (3.6, 3.7, or 3.8)
  - python3-venv (Ubuntu/Debian/Mint)
- Required to run
  - BWA, samtools
- Optional (but recommended)
  - pyenv

Guide to ETCHING

We prepared a simple guide for CentOS/Fedora or Ubuntu/Debian/Mint users. You can skip this this step if all requirements were installed.

Note: We tested this guide on Fedora32/33/34, CentOS7/8, Ubuntu16.04/18.04/20.04, Mint19/20, Debian11, and MX linux.

1. Requirements

- **CentOS/Fedora (or other Red Hat-based linux distros)**

```
# Required programs
sudo yum install -y epel-release # CentOS
sudo yum install -y gcc gcc-c++ make bwa samtools
```

- **Ubuntu/Debian/Mint (or other Debian-based distros)**

```
## Required programs
sudo apt install -y gcc g++ make bwa samtools

# You can skip this if you will use pyenv.
# Unless, python3-venv should be installed.
sudo apt install -y python3-venv
```

* Optional: `pyenv`

Highly recommended if the default version of your Python3 is >=3.9 or <=3.5.

```
# dependencies of pyenv
# For Fedora/CentOS
sudo yum install make gcc zlib-devel bzip2 bzip2-devel readline-devel sqlite
sqlite-devel openssl-devel tk-devel libffi-devel xz-devel
# For Ubuntu/Debian/Mint
sudo apt-get update
sudo apt install make build-essential libssl-dev zlib1g-dev libbz2-dev
libreadline-dev libsqlite3-dev wget curl llvm libncursesw5-dev xz-utils tk-dev
libxml2-dev libxmlsec1-dev libffi-dev liblzma-dev
```

```
# Install pyenv
curl https://pyenv.run | bash
echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.bashrc
echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.bashrc
echo 'eval "$(pyenv init -)"' >> ~/.bashrc
echo 'eval "$(pyenv init --path)"' >> ~/.bashrc
exec $SHELL
```

2. Installation

Once, requirements were solved, you can install ETCHING as follows.

```
# Download ETCHING
git clone --depth=1 https://github.com/ETCHING-team/ETCHING.git

# Move to /path/to/ETCHING
cd ETCHING

# Optional for pyenv users
pyenv install 3.7.12 # any version from 3.6.0 to 3.8.12
pyenv local 3.7.12

# Compile and install ETCHING
make
echo "export ETCHING_HOME=$PWD" >> ~/.bashrc
echo "export PATH=$PWD/bin" >> ~/.bashrc
exec $SHELL
```

As long as you keep `/path/to/ETCHING/lib`, virtual environment automatically sets `LD_LIBRARY_PATH` while running ETCHING.

3. DEMO

```
# Change directory
cd /wherever/you/want/

# Download and decompress DEMO
wget http://big.hanyang.ac.kr/ETCHING/DEMO.tar.gz
tar zxvf DEMO.tar.gz
cd DEMO

# Run demo
etching -1 tumor_1.fq -2 tumor_2.fq -1c normal_1.fq -2c normal_2.fq -g
small_genome.fa -a small_genome.gtf -f demo_PGK -o example -t 8
```

Pan-Genome k-mer set

If you have no matched normal data, our pan-genome k-mer set (PGK) will be helpful to select tumor specific reads.

```
# Move to etching directory
cd /somewhere/you/want/

# Download
wget http://big.hanyang.ac.kr/ETCHING/PGK.tar.gz

# Decompress
tar zxvf PGK.tar.gz

# Then, you will see PGK_20200103.kmc_pre and PGK_20200103.kmc_suf in PGK:
# Here, PGK_20200103 is the name of k-mer set to be used for ETCHING.
ls PGK
```

Alternatively, you can make your own k-mer set as follows:

```
make_pgk -i reference.list -o my_pgk -v dbSNP.vcf -g hg19.fa
deactivate
```

Here, `reference.list` is a file of file names of reference genomes in fasta format.

ETCHING on a ship (docker)

Requirement

docker

Download docker image

```
# Download ETCHING docker image
wget http://big.hanyang.ac.kr/ETCHING/etching_docker.tar

# Load the image
docker load -i etching_docker.tar

# Check the image
docker images
```

You can see like this

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
etching	1.3.6	16647cac9a99	40 hours ago	4.3 GB

Demo for docker user

Download our DEMO

```
# Download and decompress DEMO
wget http://big.hanyang.ac.kr/ETCHING/DEMO.tar.gz
tar zxvf DEMO.tar.gz
```

Run ETCHING with docker

```
docker run -i -t --rm -v /path/to/DEMO/:/work/ etching:1.3.6 etching -1
tumor_1.fq -2 tumor_2.fq -1c normal_1.fq -2c normal_2.fq -g small_genome.fa -a
small_genome.gtf -f /work/demo_PGK -o example_1 -t 8
```

Here, `etching:1.3.6` is `REPOSITORY` and `TAG` of ETCHING docker image.

Replace `/path/to/DEMO` with `/your/data/path/`.

Note: Keep `/work/` in the above command line.

Alternatively, you can run ETCHING inside docker container

```
docker run -i -t --rm -v /path/to/DEMO/:/work/ etching:1.3.6 /bin/bash

etching -1 tumor_1.fq -2 tumor_2.fq -1c normal_1.fq -2c normal_2.fq -g
small_genome.fa -a small_genome.gtf -f /work/demo_PGK -o example_2 -t 8
```

Contributors

Jang-il Sohn, Min-Hak Choi, Dohun Yi, A. Vipin Menon, and Jin-Wu Nam

Bioinformatic and Genomics Lab., Hanyang University, Seoul 04763, Korea

---

Contact

If you have any issues, please contact us

Jang-il Sohn (sohnjangil@gmail.com)

Jin-Wu Nam (jwnam@hanyang.ac.kr)
