Accurity

Introduction

Accurity is a computational method that infers tumor purity and tumor cell ploidy from tumor-normal WGS (whole exome will probably work too) data by jointly modelling SCNAs and heterozygous germline single-nucleotide-variants (HGSNVs). Results from both *in silico* and real sequencing data demonstrated that Accurity is highly accurate and robust, even in low-purity, high-ploidy, and low-coverage (as low as **1X**) settings in which several existing methods perform poorly. Accounting for tumor purity and ploidy, Accurity significantly increased the signal/noise gaps between different copy numbers.

- Z. Luo*, X. Fan*, Y. Su, YS. Huang (2018). Accurity: Accurate tumor purity and ploidy inference from tumor-normal WGS data by jointly modelling somatic copy number alterations and heterozygous germline single-nucleotide-variants. *Bioinformatics*. PDF

0.5XAccurity

License

The license follows our institute policy that you can use the program for free as long as you are using Accurity strictly for non-profit research purposes. However, if you plan to use Accurity for commercial purposes, a license is required and please contact yuhuang@simm.ac.cn to obtain one.

The full-text of the license is included in the software package.

Get our software

Register to receive update emails.

Please register here to receive updates and bugfixes. The link included in the email is a standalone Accurity package without dependencies. If you have trouble installing dependencies, use the docker version instead.

Docker

Latest news

2019/3/11: We made available two versions of reference packages for different versions of the human genome (hs38, hs37). Software was also changed a bit. So please pull the latest docker image.

2019/1/11: We replaced Freebayes with Strelka2 to call SNPs. The latter is faster (chromosome-level parallel) and more accurate, https://www.nature.com/articles/s41592-018-0051-x. Strelka2 is included in the docker image, at /usr/local/strelka. NO need to install it.

NOTE Due to the difficulty (i.e. no root access to install required libraries or incompatible libraries) in running our binary software, we have made a docker image available at https://hub.docker.com/r/polyactis/accurity_ubuntu/ , which contains the **latest development version** of our software and all dependent libraries. Accurity inside the docker is newer than what you can download from this website and the source code at https://github.com/polyactis/Accurity.

1. Install Ubuntu package "docker.io" before you do anything below.
2. Download the refData package from section refData.

A docker session

```
yh@cichlet:~$ docker pull polyactis/accurity_ubuntu
Using default tag: latest
latest: Pulling from polyactis/accurity_ubuntu
04651435ae61: Pull complete
ccae121f92fd: Pull complete
7bb876499e21: Pull complete
444d1ce6037a: Pull complete
f02a7b59a9fd: Pull
complete
0970b4c9aeb0: Pull complete
dac06add9540: Pull complete
bc16ce130a5b: Pull complete
a43ae26492d8: Pull complete
bddaca7e4091: Pull complete
99af9d12ed4b: Pull complete
8c21c137bff5: Pull complete
63b88c619321: Pull
complete

fd6992ef54e0: Pull
complete
Digest: sha256:
a6f72af3114ba903f26b60265e10e6f13b8d943d25e740ab0a715d1a99000188
Status: Downloaded newer image for polyactis/accurity_ubuntu:latest
yh@cichlet:~$ docker images
REPOSITORY                  TAG                 IMAGE ID
CREATED             SIZE
polyactis/accurity_ubuntu   latest              a11fdb62c5d4        5
months ago        1.04GB

# Log into the docker image, without mounting. Useful just to look inside
the docker.
yh@cichlet:~$ docker run -i -t polyactis/accurity_ubuntu /bin/bash

# Log into the docker image.
# Mount /home/mydata, which contains your bam files and the reference
data, to /mnt inside the docker
yh@cichlet:~$ docker run -i -t -v /home/mydata:/mnt polyactis
/accurity_ubuntu /bin/bash
```

```
root@cc7807445e40:/$ cd /usr/local/Accurity/
/usr/local/Accurity /
root@cc7807445e40:/usr/local/Accurity$ ls
GADA         accurity   main.py             plot_autocor_diff.
py                  plot_snp_maf_peak.py
LICENSE      configure  plot.tre.autocor.R
plot_coverage_after_normalization.py  plot_tre.py
__init__.py  infer      plotCPandMCP.py     plot_snp_maf_exp.py
root@cc7807445e40:/usr/local/Accurity$ ./main.py
usage: main.py [-h] [-v] -c CONFIGURE_FILEPATH -t TUMOR_BAM -n NORMAL_BAM -
o
               OUTPUT_DIR [--snp_output_dir SNP_OUTPUT_DIR] [--clean CLEAN]
               [--segment_stddev_divider SEGMENT_STDDEV_DIVIDER]
               [--snp_coverage_min SNP_COVERAGE_MIN]
               [--snp_coverage_var_vs_mean_ratio
SNP_COVERAGE_VAR_VS_MEAN_RATIO]
               [--max_no_of_peaks_for_logL MAX_NO_OF_PEAKS_FOR_LOGL]
               [--nCores NCORES] [-s STEP] [-l LAM] [-d DEBUG] [--auto
AUTO]
main.py: error: argument -c/--configure_filepath is required
# modify file "configure" to reflect paths of input data and relevant
binaries
root@cc7807445e40:/usr/local/Accurity$ cat configure
reference_genome_name   hs37d5
read_length     101
window_size     500
reference_index_folder_path     /mnt/refData
reference_genome_fasta_path /mnt/refData/hs37d5.fa
samtools_path   /usr/local/bin/samtools
caller_path  /usr/local/strelka
accurity_path   /usr/local/Accurity
root@cc7807445e40:/usr/local/Accurity$ ls /usr/local/bin/
total 11640
drwxr-xr-x  2 root root    4096 Jul 13 05:18 ./
drwxr-xr-x 16 root root    4096 Jul 20 09:00 ../
-rwxr-xr-x  1 root root 7470576 Jul  7  2018 freebayes*
-rwxrwxr-x  1 root root 4436160 Jul  7  2018 samtools*
```

Install piece by piece

Prerequisites

1. A computer with at least 32GB of memory (recommend 64GB).
2. Freebayes (A pre-compiled binary for Ubuntu 16.04). A variant caller that is used to call SNPs.
3. Python2
    1. matplotlib
    2. numpy

    3. pandas
    4. Pyflow https://github.com/Illumina/pyflow
4. samtools (A pre-compiled binary for Ubuntu 16.04)
5. libbz2-1.0 (a high-quality block-sorting file compressor library, install it via "apt install libbz2-1.0" in Debian/Ubuntu)
    1. If your OS (like CentOS) has this library installed but Accurity still fails to load it, you can do a symlink from the installed libarary file to "libbz2.so.1.0".
6. libgsl2
7. liblzma5 (XZ-format compression library)
8. libssl1.0.0
9. libboost-program-options1.58.0
10. libboost-iostreams1.58.0
11. (Only for building from source) pkg-config: used by Rust compiler to find library paths. i.e. "pkg-config --libs --cflags openssl"
12. (Optional) R packages ggplot2, grid, scales. Only needed if you obtain a development version of Accurity. Required to make one R plot.
    1. But the R plot is NOT a must-have, one python plot has similar content as the R plot.

Running Accurity requires a project-specific configure file, details below. Modify the path in the configure file according to your OS environment.

Install pyflow and other Python packages

```
git clone https://github.com/Illumina/pyflow.git pyflow
cd pyflow/pyflow
python setup.py build install
```

Other python packages can be installed through Python package system "pip install ..." or Ubuntu package system, dpkg/apt-get.

Register to download the Accurity binary package and receive update emails.

Please register here to receive an email that contains a download link. After finishing download, unpack the package via this:

```
tar -xvzf Accurity.tar.gz
```

Accurity is a package that contains a few binary executables and R/Python scripts. All binary executables were compiled for a Linux platform (Ubuntu 14 and 16 tested). It also contains a sample configure file. Denote the full path of the Accurity folder as accurity_path in the configure file (described below).

NOTE

1. If you are having difficulty in getting Accurity to work, please use docker instead.
2. This binary package is behind our docker release.

Compile source code (for advanced users)

Instead of downloading binary, you can also choose to compile the source code. Be forewarned, you may run into problems (missing packages, wrong paths, etc.) in compiling the C++ portion on non-Ubuntu platforms. Rust compiling is relatively easy.

Compiling Accurity requires those "lib..." packages in section 3.1 and their corresponding development packages (for example, libbz2-dev). In addition, it requires an installation of Rust, https://www.rust-lang.org/. We have compiled successfully on Ubuntu 16.04 and 18.04.

https://github.com/polyactis/Accurity

NOTE

1. The public source code on github (https://github.com/polyactis/Accurity) is an old stable version. We advise users to use the latest version that is encapsulated in the docker.

The reference genome package

The reference genome package is one of the required inputs of Accurity. The sub-folder, refData/1000g/, contains coordinates of common (allele frequency >10%) SNPs from the 1000 Genomes project. The chromosome coordinates are denoted as "chr1", not "1". We advise users to align reads against the genome file included in the package to re-generate their bam files, in order to minimize wrong alignments and more importantly, match the coordinates of the 1000Genomes SNP file. We provide three versions for different versions of the human genome.

1. hs38d1 (NCBI hs38 is equivalent to UCSC hg20, about 866MB)
2. hs37d5 (NCBI hs37 is equivalent to UCSC hg19, about 871MB)
3. an older version hs37d5 (about 1.7GB) that suits the Accurity version downloaded from this site. NOT for the docker version of Accurity.

Configuration

Input bam files

For an example, you have a pair of matched tumor and normal samples.

sample_ 1_cancer.bam

sample_1_cancer.bam.bai

sample_1_normal.bam

sample_1_normal.bam.bai

***.bam.bai** files (bam index) are not required. Accurity will call samtools to generate them if they are found to be missing.

Setup the configure file

Copy the sample configure file (tab-delimited) from the Accurity package into your project folder and modify it accordingly. An example looks like this:

```
reference_genome_name    hs37d5
read_length    101
window_size    500
reference_index_folder_path    /simm/program/refData
reference_genome_fasta_path    /simm/huangyulab/genome/hs37d5_namechr.fa
samtools_path  /program/bin/samtools
caller_path    /program/bin/freebayes
accurity_path  /home/hello/src/purity_03_29/Accurity
```

All the fields in the configure file:

reference_genome_name the version of the reference genome to which the bam files are aligned to. It is assumed that all samples under the same directory are aligned to the same reference genome.

read length the length in base pair of the read.

window_size the window size in base pair for segmentation. The segmentation program (GADA) first calculates the number of reads for each window and then perform segmentation over the genome. A small window size often leads to a large number of small segments. The recommended window size is 500bp.

reference_index_folder_path the path to the 1000 genome bi-allele SNPs file, this path should have a subdirectory 1000g/

reference_genome_fasta_path the path to the reference genome fasta file

samtools_path the path to the samtools program.

caller_path the path to the 3rd-party variant calling program. For freebayes, it's the path to the binary. For Strelka2, it is the path of the folder that contains all Strelka2 code/executables, i.e. /usr/local/strelka.

accurity_path the path to the Accurity software. See section 3.2

Run Accurity

Accurity consists of several binary executables. To make everything easy, we have written a Python program **main.py** (inside the "Accurity" folder) which wraps all binary executables in a workflow.

./main.py –h gives you an explanation of all the arguments:

```
yh@hello:~/Accurity$ ./main.py  -h
usage: main.py [-h] -c CONFIGURE_FILEPATH -t TUMOR_BAM -n NORMAL_BAM -o
               OUTPUT_DIR [--snp_output_dir SNP_OUTPUT_DIR] [--clean CLEAN]
               [--segment_stddev_divider SEGMENT_STDDEV_DIVIDER]
               [--snp_coverage_min SNP_COVERAGE_MIN]
               [--snp_coverage_var_vs_mean_ratio
SNP_COVERAGE_VAR_VS_MEAN_RATIO]
               [--max_no_of_peaks_for_logL MAX_NO_OF_PEAKS_FOR_LOGL]
               [--nCores NCORES] [-s STEP] [-l LAM] [-d DEBUG] [--auto
AUTO]

optional arguments:
  -h, --help            show this help message and exit
  -c CONFIGURE_FILEPATH, --configure_filepath CONFIGURE_FILEPATH
                        the path to the configure file.
  -t TUMOR_BAM, --tumor_bam TUMOR_BAM
                        the path to the tumor bam file. If the bam is not
                        indexed, an index file will be generated
  -n NORMAL_BAM, --normal_bam NORMAL_BAM
                        the path to the normal bam file. If the bam is not
                        indexed, an index file will be generated
  -o OUTPUT_DIR, --output_dir OUTPUT_DIR
                        the output directory path.
  --snp_output_dir SNP_OUTPUT_DIR
                        the directory to hold the SNP calling output.
Default
                        is the same folder as the bam file.
  --clean CLEAN         whether to remove the existing output folders and
                        files? 0 No, 1 Yes. Default is 0.
  --segment_stddev_divider SEGMENT_STDDEV_DIVIDER
                        A factor that reduces the segment noise level. The
                        default value is recommended. Default is 20.
  --snp_coverage_min SNP_COVERAGE_MIN
                        the minimum SNP coverage in adjusting the expected
SNP
                        MAF. Default is 2.
  --snp_coverage_var_vs_mean_ratio SNP_COVERAGE_VAR_VS_MEAN_RATIO
                        Instead of using the observed SNP coverage variance
```

```
                        (not consistent), use coverage_mean X this-
parameter
                        as the variance for the negative binomial model
which
                        is used in adjusting the expected SNP MAF. Default
is
                        10.
  --max_no_of_peaks_for_logL MAX_NO_OF_PEAKS_FOR_LOGL
                        the maximum number of peaks used in the log
likelihood
                        calculation. The final logL is average over the
number
                        of peaks used. Default is 3
  --nCores NCORES       the max number of CPUs to use in parallel. Increase
                        the number if you have many cores. Default is 2.
  -s STEP, --step STEP  0: start from the very begining (Default).
                        1: obtain the read positions and the major allele
fractions.
                        2: normalization. 3: segmentation. 4: infer purity
and
                        ploidy only.
  -l LAM, --lam LAM     lambda for the segmentation algorithm. Default is
4.
  -d DEBUG, --debug DEBUG
                        Set debug value. Default is 0, which means no debug
                        output. Anything >0 enables several plots being
made.
  --auto AUTO           The integer-valued argument that decides which
method
                        to use to detect the period in the read-count ratio
                        histogram. 0: the simple auto-correlation method.
1: a
                        GADA-based algorithm (recommended). Default is 1.
```

In the debug mode (-d 1), Accurity will produce several intermediate plots, offering insights into how well it is handling the input data.

Run Accurity from scratch given two input bam files, use 30 cores, output to folder sample_1_infer, enable debug mode

```
./main.py –c configure_file --nCores 30 -t sample_1_cancer.bam -n
sample_1_normal.bam -o sample1_output -d 1
```

Resume Accurity from step 2 and change the snp output folder (default was the bam file folder)

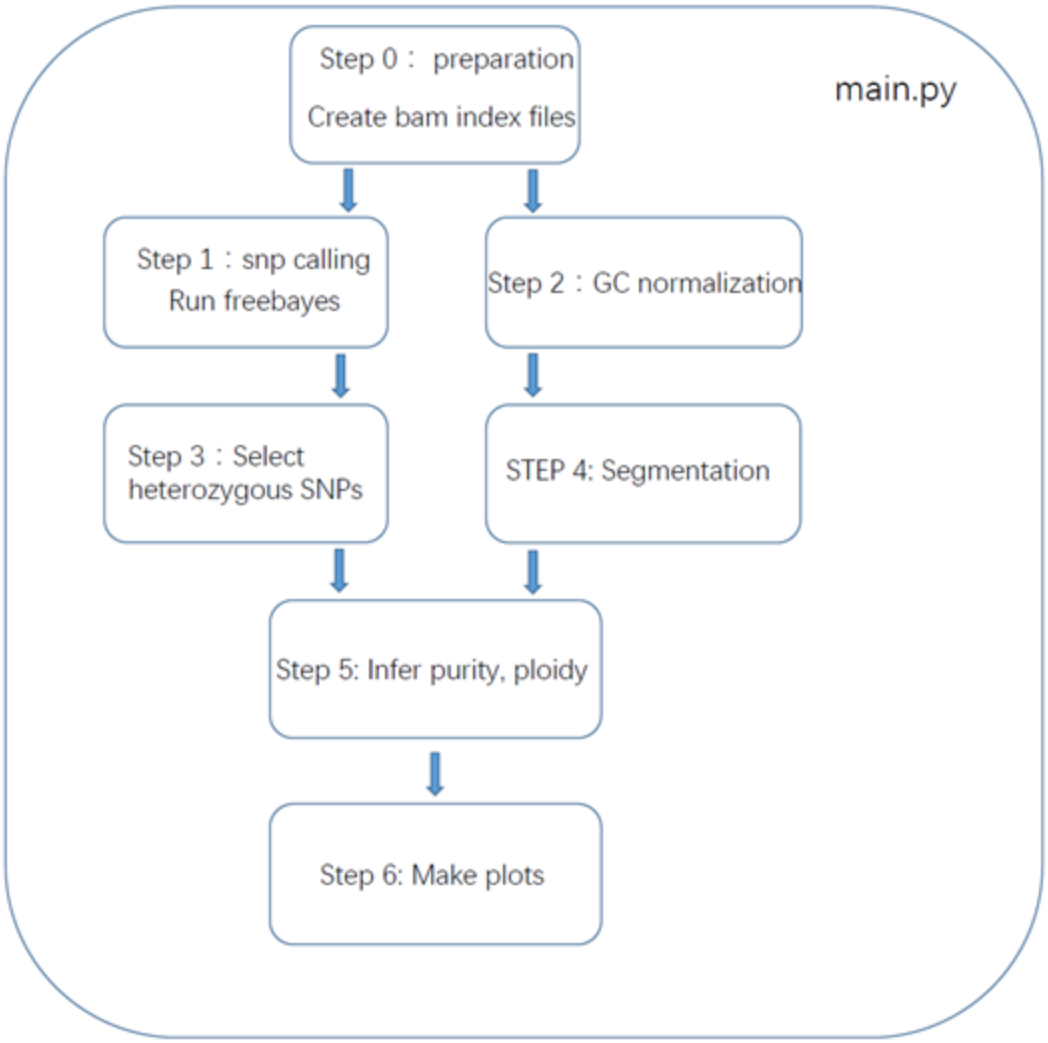
```
./main.py -c configure_file --nCores 20 -t sample_1_cancer.bam -n
sample_1_normal.bam -o sample1_output --snp_output_dir sample1_output -d 1
--step 2
```

Override all previous output:

```
./main.py -c configure_file -t sample_1_cancer.bam -n sample_1_normal.bam -
o sample1_output -d 1 --clean 1
```

Accurity workflow

Accurity contains 7 major components. First, it will check whether the bam index files exist, if not, Accurity will create them. Then, it carries out SNP calling and the coverage normalization (in parallel, one job per chromosome). Next, call heterozygous SNPs and segment each chromosome (in parallel, one job per chromosome). After that, Accurity infers the purity and ploidy. Last, it will plot some results. The whole workflow structure is as follows.



Accurity output

These are the output files that matters.

infer.out.tsv

A summary output that contains purity and ploidy estimates, and some other statistical measures. Probably the most important file to a user.

infer.out.tsv example

```
purity  ploidy
0.66735 2.0612
logL    period  best_no_of_copy_nos_bf_1st_peak first_peak_int
9.9811e+06      327     2       980
no_of_segments  no_of_segments_used     no_of_snps      no_of_snps_used
539     539     1333539 933576
```

infer.out.details.tsv

This contains lots of internal model output, useful for developers.

cnv.output.tsv

This contains preliminary copy number alteration predictions.

The important columns are chr, start, end.

"cp" is the predicted copy number.

"copy_no_float" is the raw copy number outputted by our program, which will be converted to an integer (the "cp" column) if our model deems it a clonal (shared by all cancer cells) CNV. Some "cp" will stay as "float" because our model thinks they are subclonal (some cancer cells in one CNV state, some cancer cells in another).

cnv.output.tsv example

```
chr     start      end        cp      major_allele_cp copy_no_float
oneSegment.stddev       maf_mean        maf_stddev      maf_expected
cumu_start   cumu_end
5       8215001 8363001       2       1       1.76147 0.00987896
0.622568        0.0685194       0.632948        895215001       895363001
3       16591001        16751001       2       1       1.76758
0.00891515      0.62856 0.063834        0.632948        511591001
511751001
...
```

plot.cnv.png

A genome-wide CNV plot.

plot.tre.jpg

A period plot. Check if the model fits data well.

plot.tre.autocor.jpg

A plot for developers.

A clean-data example

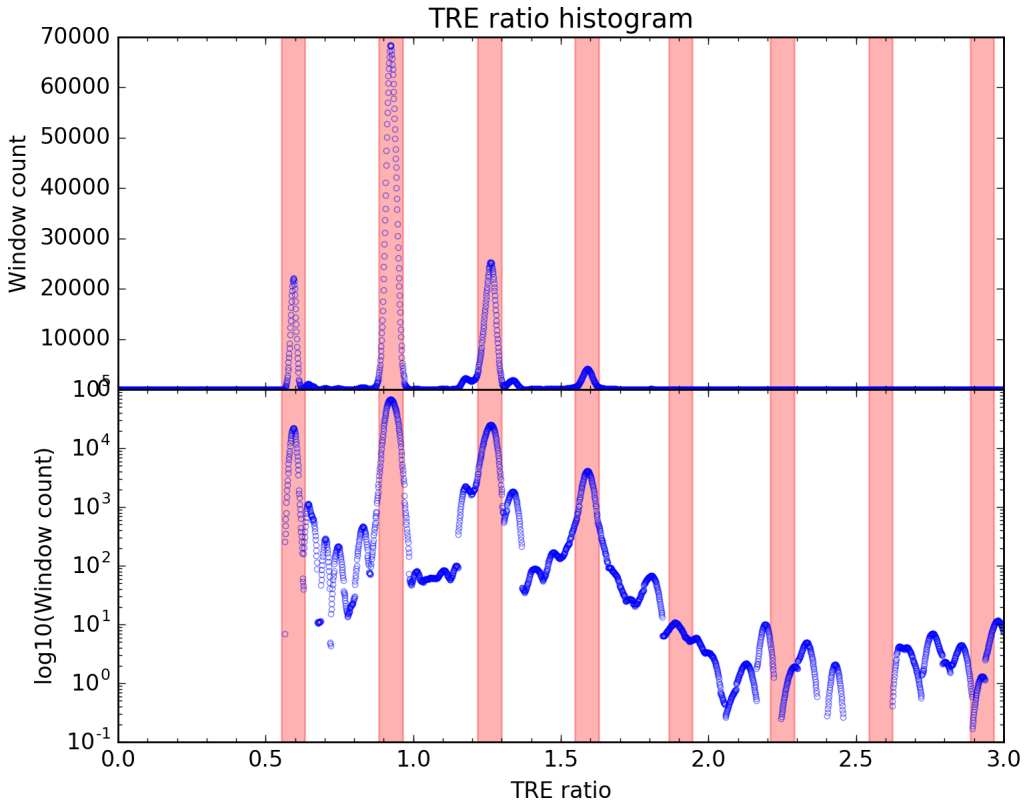
All major results are stored in the output directory. File sample_1_infer/**infer.out.tsv** contains the purity and ploidy estimates. Here is an example. (Viewing in Excel is a lot nicer) :

```
purity        ploidy       purity_naive     ploidy_naive
rc_ratio_of_cp_2      rc_ratio_of_cp_2_corrected
segment_stddev_divider         snp_maf_stddev_divider  snp_coverage_min
snp_coverage_var_vs_mean_ratio    period_discover_run_type
0.7246      2.2428      0.72078    2.2282      924  919.13      10
20     2       10     1
logL period       best_no_of_copy_nos_bf_1st_peak first_peak_int
1.5204e+07      333  1       585
no_of_segments       no_of_segments_used      no_of_snps
no_of_snps_used
697  697  1517851  1062619
```

In the output above, the column ‘purity’ is the final purity estimate, and ‘purity_naive’ is the pre-adjusted estimate which can be ignored. ‘logL’ is the maximum likelihood of the hierarchical Gaussian Mixture model. ‘period’ is the 1000 X period of the tumor-read-enrichment (TRE) histogram (=333 in this case), which is detected by auto-regression. 'no_of_snps' and ‘no_of_segments’ is the result of step 3 and step 4. Other columns are values of commandline arguments.

There are other important output files, such as all_segments.tsv.gz and het_snp.tsv.gz, which are output of step 4 and step 3 respectively. If the sample is abnormal, we can usually see an unreasonable number of segments and SNPs in these two files.

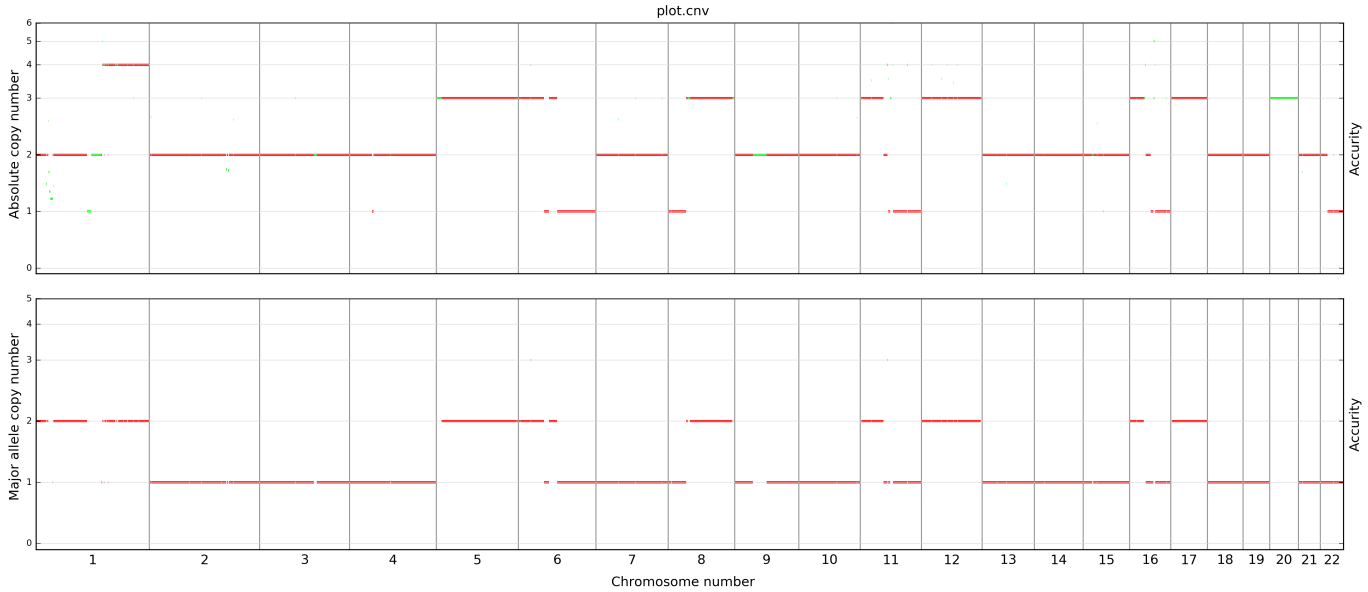
Besides the text output, Accurity will produce some graphic output. One of the the most important plots is plot.tre.png, **available only in debug mode (-d 1)**:



TRE stands for Tumor Read Enrichment. You can think of it as a normalized version of the read count ratio between the tumor and normal samples for one chromosome window. More details can be found in our paper. The Y axis in the two panels is the window count. The lower panel is in the log scale. A clean TRE histogram leads to a confident purity estimate.

In this clean-data example, the tumor read enrichment (TRE) histogram displays a beautiful periodic pattern. That means we can confidently infer the period (=0.333) from the TRE data and the ensuing maximum likelihood estimates will be more robust. The CNV estimates (a by-product of Accurity), in plot.cnv.png, also demonstrates a clean copy number variation (CNV) profile.

plot.cnv.png:



This is the estimated CNV profile for the example. The top plot is the estimated absolute copy number for each segment. For a normal sample, the absolute copy number should be 2 throughout the genome. The lower plot shows the major allele copy number for each segment.

There are cases where purity and ploidy can not be inferred:

1. The cancer genome contains too few somatic copy number alterations.
2. The noise level is too high, or the noise level is moderate but the sample purity is very low (<0.05).

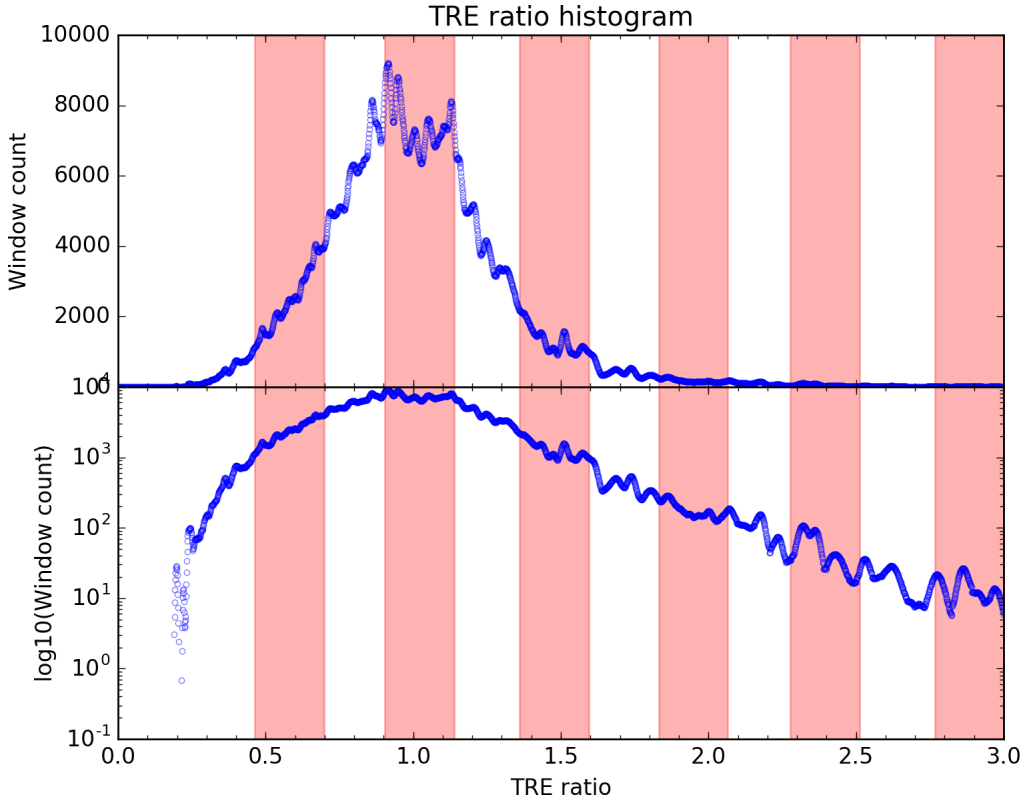
A noisy-data example

Occasionally, a user will encounter extremely noisy data. The user should learn to identify the noisy data from plots and do NOT use the estimates made by Accurity. In the future, we will probably stop Accurity from making any estimate. But for the time being, here is a noisy-data example.

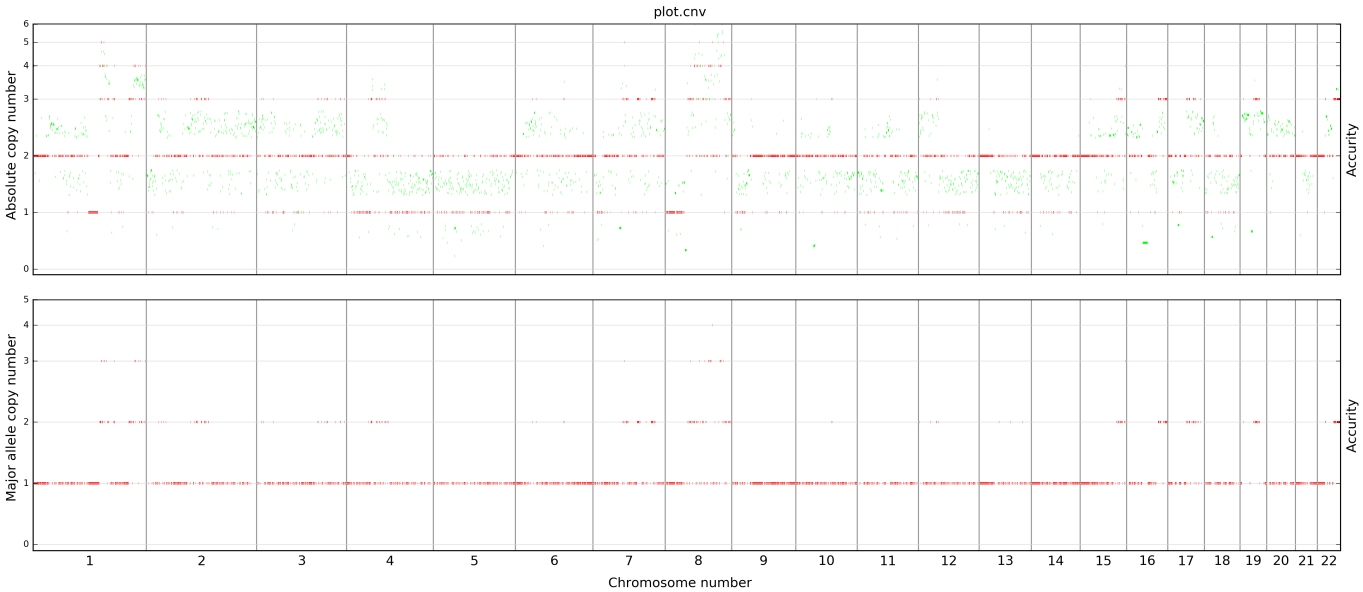
Content of infer.out.tsv for a noisy-data example. The **high number of segments** is a red flag.

```
purity        ploidy       purity_naive     ploidy_naive
rc_ratio_of_cp_2      rc_ratio_of_cp_2_corrected
segment_stddev_divider   snp_maf_stddev_divider  snp_coverage_min
snp_coverage_var_vs_mean_ratio         period_discover_run_type
0.90938    1.9375      0.91675    1.9551      1021 1029.3      10
20     2       10     2
logL period       best_no_of_copy_nos_bf_1st_peak first_peak_int
4.3681e+06      468  1       555
no_of_segments       no_of_segments_used      no_of_snps
no_of_snps_used
19909       19909       1559676  1092048
```

Its tumor-read-enrichment (TRE) histogram (plot.tre.png) has **one big and unclean peak** (its landscape looks like being cut through by a lousy jigsaw). It makes it really difficult to accurately estimate its period. The period estimate (0.468, 468 in the 2nd cell of the 4th line is 1000Xperiod.) is probably far from the truth. All ensuing maximum likelihood estimates are questionable. The estimated CNV profile further confirms the great amount of noise in this data.



plot.cnv.png:



Feedback

If you encounter any issues, please email polyactis@gmail.com or file an issue at https://github.com/polyactis/Accurity/issues (so that everyone can learn).

Attachments

File	Modified
PNG File clean.plot.cnv.png	May 08, 2018 by Yu Huang
PNG File clean.plot.tre.png	May 08, 2018 by Yu Huang
File freebayes v1.1.0-54-g49413aa	May 08, 2018 by Yu Huang
PNG File image2018-5-7_3-12-13.png	May 08, 2018 by Yu Huang
PNG File noise.plot.cnv.png	May 08, 2018 by Yu Huang
PNG File noise.plot.tre.png	May 08, 2018 by Yu Huang
File samtools samtools 1.4-19-g8bd76fe Using htslib 1.4-22-gaf89ccb Copyright (C) 2017 Genome Research Ltd.	May 08, 2018 by Yu Huang

Download All