RVD2.7 command line program (CLI) instructions

Contents
I. The overall Flowchart of RVD2 program .................................................................................................................. 1
II. The overall Flowchart of Test functions ..................................................................................................................... 2
III. RVD2 CLI syntax ....................................................................................................................................................... 3
IV. RVD2 CLI demo ......................................................................................................................................................... 5

I. The overall flowchart of RVD2 program

Figure 1 provides the overall flowchart for RVD2 algorithm. We start from bam files, and then use samtools mpileup to convert bam files to pileup files. Next, we use a pileup2dc program to convert pileup files(.pileup) to depth chart files (.dc). Depth charts are tab-delimited text files that record the count of the number of {A,C,T,G} in columns and genomic position in rows. Finally, we feed depth chart files to rvd27.py to call variants.

RVD2 has several positional arguments after the command. The positional arguments are detailed in the RVD2 CLI syntax section. Briefly, gen is used to generate data from the model, gibbs is used to perform MCMC sampling to estimate the posterior distributions of the latent variables in the model, and various test functions are used to run specific hypothesis tests to identify variants. The specific hypothesis test run depends on whether case and control data is available and whether you are looking for somatic or germline variants.

The subcommand gibbs is applied to the depth chart, to fit the RVD2 statistical model and generate hdf5 model files. Then input the hdf5 files to one of the test functions and call variants. The results are output in hdf5 vcf format.

Figure 1 The overall Flowchart of RVD2 program
II. The overall flow chart of test functions

Figure 2 shows the four types of test functions available in RVD2 program, and Figure 3 summarizes the difference between them. **One_sample_test** is a Bayesian posterior distribution test, which reports the positions where $1 - \alpha$ percent of the samples are within the interval of interest, intvl in one single sample. **Germline_test** uses an optional chi2 test with the one_sample_test, to improve specificity. **Paired_difference_test** is a one-sided Bayesian posterior distribution test. This test requires both control and case samples. This test reports if the error rate in case sample is significantly higher than the control sample. **Somatic_test** is a two-sided Bayesian posterior distribution test, which reports positions where control sample are significantly different from case sample.

![Test functions diagram](image)

**Figure 2** The structure of hypothesis testing functions in RVD2

<table>
<thead>
<tr>
<th>Test Type</th>
<th>one-sided</th>
<th>two-sided</th>
</tr>
</thead>
<tbody>
<tr>
<td>one sample</td>
<td>Germline test</td>
<td>NA</td>
</tr>
<tr>
<td>two samples (control-case paired)</td>
<td>Paired difference test</td>
<td>Somatic test</td>
</tr>
</tbody>
</table>

**Figure 3** Overview of three test functions. The chi square test can improve specificity and is optional in all the tests. In the paper we performed paired difference test on the synthetic dataset, and germline test and somatic test on the clinical dataset due to the specific way those experiments were conducted for sequencing.
III. RVD2 CLI syntax

We provide a python module containing the core functionality of RVD2. This module depends on the following modules: numpy, scipy, itertools, h5py tempfile, logging, datetime, os, subprocess, re, pdb and time. To run in multithreaded mode, you also need the multiprocessing module.

RVD2 can be run as a command line program or imported into an existing python script as a module.

CLI Syntax:

usage: rvd27 [-h] [--version] [-v]
       {gen,gibbs,one_sample_test,germline_test,paired_difference_test,somatic_test}

RVD is a hierarchical Bayesian model for identifying rare variants from short-read sequence data.

Positional arguments:
(gen, gibbs, one_sample_test, germline_test, paired_difference_test, somatic_test)

sub-command help

gen Demo: generate simulation sample data from the RVD model
       -h, --help show this help message and exit
       -N Number of replicates in computer simulation data
       -J Number of positions in computer simulation data
       -s SEEDINT random process seed.

gibbs fit the RVD model using Gibbs sampling

positional arguments:
    dcf file depth chart file name

optional arguments:
    -h, --help show this help message and exit
    -o OUTPUTFILE output HDF5 file name, default (output)
    -p, --pool POOL number of workers in multithread pool, default None
    -g, --ngibbs NGIBBS sampling size, default 4000
    -m, --nmh NMH Metropolis-Hastings sampling size, default 10
    -b, --burnin BURNIN , default 0.2
    -t, --thin THIN thin, default 2
    -s SEEDINT random process seed.

one_sample_test One side Bayesian posterior density test of one single sample

positional arguments:
    HDF5Name HDF5 sample file

optional arguments:
    -h, --help show this help message and exit
    -i, --intvl INTVL interval of interest in in posterior distribution.
    -a, --alpha ALPHA hypothesis test credible level
    -o OUTPUTFILE output HDF5 file name, default (output)

germline_test Germline test on a single sample, which includes a one side Bayesian density test and an optional chi square test.

positional arguments:
    HDF5Name HDF5 sample file
optional arguments:
-h, --help       show this help message and exit
-i, --intvl     INTVL interval of interest in in posterior distribution.
-a, --alpha     ALPHA hypothesis test credible level
-o, --chi2      Whether to include chi square test in the germline test, default True (Include)

paired_difference_test
One sided posterior density difference test on control-case paired sample, with
an optional chi square test.

positional arguments:
controlHDF5Name HDF5 control sample file
caseHDF5Name     HDF5 case sample file

optional arguments:
-h, --help       show this help message and exit
-i, --intvl     INTVL interval of interest in in posterior distribution.
-a, --alpha     ALPHA hypothesis test credible level
-o, --chi2      Whether to include chi square test in the paired difference test, default True
-s, --seed      random process seed.
-n, --n          N Posterior difference distribution sampling size.

somatic_test
Somatic test, which includes a two sided posterior density difference test and
chi square test on the control-case paired sample.

positional arguments:
controlHDF5Name HDF5 control sample file
caseHDF5Name     HDF5 case sample file

optional arguments:
-h, --help       show this help message and exit
-i, --intvl     INTVL interval of interest in in posterior distribution.
-a, --alpha     ALPHA hypothesis test credible level
-o, --chi2      Whether to include chi square test in the paired difference test, default True
-s, --seed      random process seed.
-n, --n          N Posterior difference distribution sampling size.

Optional arguments:
-h, --help       show this help message and exit
--version       show program's version number and exit
-v, --verbose   increase verbosity (specify multiple times for more)
IV. RVD2 CLI demo

![Diagram](image)

Figure 4 The flowchart of a demo using computer generated simulation data

A demo written in bash is provided to test if RVD2 can be successfully run in the computer. In the demo, simulation data will be generated by the computer and variants will be called in the simulation data. A success message will be displayed upon finish. Three hdf5 files and a vcf file will be created in the directory provided.

The content inside of the vcf file is provided in below. The positions in the vcf file are variants called by RVD2.

```vcf
##fileformat=VCFv4.1
##fileDate=20140514
##source=rvd2
##Posterior test in cancer-normal-paired sample.
##contig=<ID=0,len=10>
##INFO=<ID=COAF,Number=1,Type=Float,Description="Control Allele Frequency">
##INFO=<ID=CAAF,Number=1,Type=Float,Description="Case Allele Frequency">
##FORMAT=<ID=AU,Number=1,Type=Integer,Description="Number of 'A' alleles">
##FORMAT=<ID=CU,Number=1,Type=Integer,Description="Number of 'C' alleles">
##FORMAT=<ID=GU,Number=1,Type=Integer,Description="Number of 'G' alleles">
##FORMAT=<ID=TU,Number=1,Type=Integer,Description="Number of 'T' alleles">

#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT  Normal  Case
chr0 5  .  A  C  .  PASS  COAF=0.921;CAAF=13.120  AU:CU:GU:TU  1988:4:4  1711:289:0:0
chr0 7  .  A  C  .  PASS  COAF=0.825;CAAF=10.421  AU:CU:GU:TU  1985:5:5  1779:221:0:0
```