Examples

All scripts in this tutorial are designed for the Bash Unix shell. You may need to change some of the commands if you use other shells. In Mac OS, zcat may need to change to gzcat. Lines starting with ### are manual operations.

Examples 1: large-scale case-control study of a complex disease

[Back to top]

This example guides you to perform gene prioritization or genetic association test for a complex disease. The hypothesis is that the missing heritability of the disease lies in low-frequency to rare variants conferring an intermediate to high risk.

1.1 Analysis without phasing

First, prepare the input files including a Pedigrees File (pedigrees.txt), a Sample File (samples.txt), and a Seed File (seeds.txt). If you have allele frequency data specific to the study population, please also prepare an Annotation File with allele frequency (af.txt). If your samples were target-enriched by different reagents or sequenced by different platforms or depths, then you need to generate a Cohort File (cohort.txt). Here a cohort is a sample set that is target-enriched by the same reagent and sequenced by the same platform with the same depth. Please read VICTOR's manual for the file formats. Quality controls of .bam files should have been done, i.e., contaminated or low-yield samples should have been removed from the samples.txt and pedigrees.txt. Suppose you have a raw VCF file that contains genotypes for all samples and all chromosomes. There is no need to decompose, normalize, or annotate the VCF file.

Second, create a data file directory (e.g., /path/to/GRCh37/PI_name/project_ID/files/) to store the input files such as the Pedigree File, Sample File, etc. The programs will not modify the input files; so you don't need to have write permissions to that directory. Then, create a working directory. I like to let the full path indicates the genome assembly, such as /path/to/GRCh37/PI_name/project_ID/analysis_1/. The programs will identify the assembly from the full path and automatically choose the right database to use. Copy and paste the slurm.all_steps and par.txt to this directory and modify parameters in it following the instructions inside the script. For a large-scale study, you may want to change the rare variant association test statistics to regression. This can be done by setting the AAA parameter in the slurm.all_steps. For example, if you want to do penalized logistic regression, then AAA="--logistic"; in addition, if you want to use individual weights to control for familial correlations, then AAA="--logistic --weight=../files/Individual_Weight_File" (replace the Individual_Weight_File with the real file name). If you have a Cohort File, then you also need to set QC1="--cohort=../files/cohort.txt" and QC2="--cohort=../files/cohort.txt".

Finally, do the following at the working directory:

If the parameters are correct, the samples.txt and pedigree.txt have no errors, and the PROVEAN program is already installed, then the procedure is quite straightforward:

```
# 1. Calculate the biological relevance of each gene to the disease of interest by a gene network analysis.
gnGBA -s seeds.txt > DiseaseOfInterest.GBA_scores

# 2. Run analyses.
# Copy /path/to/VICTOR/script_template/slurm.all_steps to a working directory. Set parameters in it and run the script by the command:
victor.sbatch --array=1 slurm.all_steps
```

If you are not sure about the QC filters, the samples.txt and pedigree.txt, or PROVEAN is not installed, then:

```
# Calculate the biological relevance of each gene to the disease of interest by a gene network analysis.
gnGBA -s seeds.txt > DiseaseOfInterest.GBA_scores
```

```
# Step 1: variant-wise quality control and basic annotations (MaxAF, BayesDel, functional consequence).
### 1) Copy /path/to/VICTOR/script_template/slurm.all_steps to a working directory. Set parameters following the instructions in the script.
### 2) Set "yes" to STEP_0 to SPL_QC, "no" to anything after, then run the script by the command below.
victor.sbatch --array=1 slurm.all_steps

# Step 1.x: sample-wise quality control again after changing some parameters.
# Step 2: More annotations (BayesDel, functional consequence).
# Step 3: PERCH analysis.
### 1) If you did not install PROVEAN and $OUT.for_PROV is not empty, submit it to http://provean.jcvi.org/ and get the condensed result,
###    name it xxx.result.one.tsv (here xxx can be anything), then set PVN=xxx.result.one.tsv in slurm.all_steps.
### 2) Review the files $OUT.sample_qc and $OUT.sample_rm. If necessary, change the QC cutoff in slurm.all_steps based on
###    study population, target region, sequencing depth, and value distributions in $OUT.sample_qc.
### 3) Review $OUT.king.kin0 and $OUT.king.kin to find cryptic relatedness and pedigree errors. Correct samples.txt and pedigree.txt.
### 4) Review $OUT.data_c.eigenvec to find population outliers. If necessary, remove them from samples.txt and pedigrees.txt.
### 5) Set "yes" to SPL_QC and beyond, "no" to STEP_0 and STEP_1 in slurm.all_steps. Run the script by the command below.
victor.sbatch --array=2 slurm.all_steps
```

It's important to check errors and warnings in the stderr log file after each analysis.

1.2 Analysis with phasing

If you want to do phasing, then you need to submit jobs to multiple computer nodes in a cluster for STEP_2 because this step is time-consuming.

```
# Calculate the biological relevance of each gene to the disease of interest by a gene network analysis.
gnGBA -s seeds.txt > DiseaseOfInterest.GBA_scores

# Step 1: variant-wise quality control and basic annotations (MaxAF, BayesDel, functional consequence).
### 1) Copy /path/to/VICTOR/script_template/slurm.all_steps to the working directory, save as slurm.step1.
### 2) Set "no" to STEP_2, STEP_3 and beyond. Set other parameters following the instructions in the script.
### 3) Run the script by the command below.
victor.sbatch --array=1 slurm.step1

# Step 1.x: sample-wise quality control again after changing some parameters. No need to do this step if nothing is changed.
### 1) Review the files $OUT.sample_qc and $OUT.sample_rm. If necessary, change the QC cutoff in slurm.step1 based on
###    study population, target region, sequencing depth, and value distributions in $OUT.sample_qc.
### 2) Review $OUT.king.kin0 and $OUT.king.kin to find cryptic relatedness and pedigree structure errors, respectively. Correct them.
### 3) Review $OUT.data_c.eigenvec to find population outliers. If necessary, remove them from samples.txt and pedigrees.txt.
### 4) Set "no" to STEP_1 and "yes" to Step 1.x in slurm.step1. Run the script by the command below.
victor.sbatch --array=2 slurm.step1

# Step 2: Phasing and more annotations (BayesDel, functional consequence).
### 1) Copy /path/to/VICTOR/script_template/slurm.step2 to the working directory. Set parameters following the instructions in the script.
### 2) If $OUT.for_PROV is not empty, submit it to http://provean.jcvi.org/ to calculate PROVEAN scores, get the condensed result file,
###    name it xxx.result.one.tsv (here xxx can be anything), then set PVN=xxx.result.one.tsv in slurm.step2.
victor.sbatch slurm.step2

# Step 3: PERCH analysis.
### 1) Copy slurm.step1 to slurm.step3.
### 2) Set "yes" to STEP_3 and beyond, "no" to all previous steps.
### 3) Run the script by the command below.
victor.sbatch --array=1 slurm.step3
```

1.3 Details of the top genes

After the analysis, you may want to see what variants contributed to the result of a gene:

```
GENE=CRNN
grep $'\t'$GENE$'\t' $OUT.results_detail
```

1.4 Using another gene database

The above example uses the default gene database refGeneLite. In an exploratory investigation, you may want to use a larger gene database such as Ensembl. Modify the par.txt with the following command and then run all analysis again.

```
echo "--gdb=ensGeneLite" >> par.txt
```

Examples 2: Small-scale pedigree-based study of a Mendelian disease

[Back to top]

The strategy of analysis in this situation is a little different from large-scale studies. Since the sample size is small, you may want to relax the quality control filter and BayesDel filter to retain as many variant as possible, and filter out the false positives by filters. Suppose you sequenced these families, and then performed a joint variant calling with samples from other studies. In this case you don't want to remove a variant just because it is not genotyped in the other studies, so you need a special sample file that contains the studied samples only and use this sample file for QC. Below is an example of parameters in slurm.all_steps:

```
STEP_0=yes # Step 0  : Split a VCF by chromosome.
STEP_1=yes # Step 1  : Variant-wise quality control and basic annotations (MaxAF, BayesDel, func consequence)
```

```
SPL_QC=yes # Step 1.a: Sample-wise quality control
PED_QC=yes # Step 1.c: Calculate sample relatedness (for pedigree structure quality control)
REL_QC=yes # Step 1.d: Calculate sample relatedness (for cryptic relatedness quality control)
SPL_PC=no  # Step 1.e: Principle component analysis (for population stratification quality control)
IMPUTE=no  # Step 1.f: Prepare a file for imputation, do imputation yourself, then merge VCFs
STEP_2=yes # Step 2  : More annotations (BayesDel, functional consequence)
STEP2B=no  # Step 2.b: PERCH report incidental findings
STEP_3=yes # Step 3  : PERCH analysis (gene prioritization, variant prioritization, VUS classification, etc.)
STEP4A=yes # Step 4.a: PERCH pathway/gene-set analysis
STEP4B=no  # Step 4.b: PERCH analysis after removing samples carrying variants in known high-penetrance genes
STEP4C=no  # Step 4.c: PERCH analysis with ExAC nonTCGA. Not recommended

export VCF=../files/MyStudy.vcf.gz
export SPL=../files/samples.txt
export PED=../files/pedigrees.txt
export GBA=../files/disease.gba
export QC1="--spl=../files/samples_in_this_study.txt"         # do missing rate QC with this study samples only
export QC2="--spl=../files/samples_in_this_study.txt"         # do missing rate QC with this study samples only
export VAR_MISS=0.2                                           # increase variant-wise missing rate cutoff
export SPL_MISS=0.2                                           # increase sample-wise missing rate cutoff because the sample size is small
export AAA="--filt-MaxAF=0.01 --filt-FdrAF=0 --filt-del=-inf" # do not filter variant by BayesDel or allele frequency in samples
export SEG="--filt-MaxAF=0.01 --filt-FdrAF=0 --filt-del=-inf" # do not filter variant by BayesDel or allele frequency in samples
```

After the analysis, you can remove false positives by filtering out variants with a negative co-segregation score, or genes that have a similar or more severe variant (in terms of BayesDel) observed in controls or in public databases such as gnomAD.

```
vFILT $OUT.results_varRank --detail $OUT.results_detail --filt-obs=1 --filt-del=MaxAF_20170801.ann.del.gz --filt-seg=0
```

Examples 3: de *novo* mutations and structural variations

[Back to top]

You are working on a rare disease (incidence 1/50000) that has a very early age of onset and a fatal outcome. Because of these, we hypothesize that the disease is caused by *de novo* mutations. Recessive inheritance or compound heterozygosity is possible, but in this analysis we are only interested in *de novo* mutations with a dominant model. You performed whole genome sequencing on a number of trios (an affected child with two unaffected parents). All three persons in a trio are sequenced. The reason to choose whole genome sequencing is to detect structural variations, and the price should not be too high since the sample size is not large.

Suppose you got two files from the sequencing lab: one VCF with SNVs and InDels for all trios and independent controls, and one VCF with structural variations. The analysis is similar to the above example. Below I will show only the differences.

First you need to do some basic QC on the structural variation genotypes. You don't need to run the QC methods that are based on allele frequencies, because they will be taken care of by vAAA later. Then run vSVA to record damaged genes. If you have multiple VCF files with structural variations for different samples, do them separately, then in slurm.all_steps set $AAA with --sv=vSVA.output.detail.file(s). However, it is highly recommended to call structural variations for all samples jointly.

```
# QC on structural variations.
vQC MyStudySV.vcf.gz --ped=pedigrees.txt --spl=samples.txt --filt-obs-pv=0 --filt-de-novo=0 >MyStudySV.qc.vcf 2>MyStudySV.qc.log
vSVA MyStudySV.qc.vcf --spl=samples.txt --detail=MyStudySV.detail
```

Then, in slurm.all_steps set the following

```
SPL=samples.txt
PED=pedigrees.txt
VAG="--par= --pas= --pav="
AAA="--filt-miss-rate=0.05 --ped=$PED --sv=MyStudySV.detail --de-novo=1"
SEG="--do-nothing"
FIN="--biol=$OUT.bio --no-ncRNA"
```

Explanations: Because the sample size is small, you may want to increase the missing rate threshold, hence the "--filt-miss-rate=0.05". It's important to set SEG="--do-nothing", so that the script will not do linkage analysis. Since the hypothetical causal variant have a severe functional consequence, you may want to reduce the searching space by excluding transcription factor binding sites, miRNA binding sites, and non-coding RNA genes. Hence, the setting "--par= --pas= --pav=" for $VAG and "--no-ncRNA" for $FIN. The --sv option reads the outputs of vSVA, so that vAAA will jointly analyze structural variations and other variants. the "--de-novo=1" option for $AAA tells the vAAA program to include only the genes with at least one *de novo* mutation among the trios. Lastly, to identify *de novo* variants, vAAA needs the "--ped=$PED" argument.

Examples 4: de *novo* mutations and compound heterozygosity

[Back to top]

Continue with the above scenario but consider a recessive model. Now it is important to phase the genotypes. Below are the parameters for

slurm.step1:

```
SPL=samples.txt
PED=pedigrees.txt
QC1="--out-dn=$OUT.chr$JOB.dn_log"
```

slurm.step2:

```
SPL=samples.txt
PED=pedigrees.txt
PHA=BEAGLE
BGL="$HOME/local/app/beagle.r1399.jar ped=pedigrees.txt"
VAG="--par= --pas= --pav="
QC2="--filt-miss-rate=0.05 --insert-dn=$OUT.chr$JOB.dn_log"
```

slurm.step3:

```
SPL=samples.txt
PED=pedigrees.txt
VAG="--par= --pas= --pav="
AAA="--ped=$PED --filt-miss-rate=0.05 --de-novo=1 --sv=MyStudySV.detail"
SEG="--do-nothing"
FIN="--biol=$OUT.bio_long_list --no-ncRNA"
```

Explanations: Both slurm.step1 and slurm.step3 are modified from VICTOR/script_template/slurm.all_steps; while slurm.step2 is modified from VICTOR/script_template/slurm.step2. See the above "Analysis with phasing" section for more details. Because the study involves trios, it is better to use BEAGLE than IMPUTE2, so PHA=BEAGLE and BGL="$HOME/local/app/beagle.r1399.jar ped=pedigrees.txt". However, *de novo* mutations will be removed by BEAGLE, so they need to be recorded in the first vQC and then inserted back by the second vQC run. Therefore, QC1="--out-dn=$OUT.chr$JOB.dn_log" and QC2="--insert-dn=$OUT.chr$JOB.dn_log". Lastly, the "--de-novo=1" option tells that only the genes with at least one *de novo* mutation will be considered.

Examples 5: compare cases to ExAC

[Back to top]

Suppose you have sequenced some cases of European origin, and would like to compare them with ExAC. First, prepare the coverage files according to the Manual. Then, in slurm.all_stpes, set the following:

```
# What to do.
ExACnT=yes

# Main Parameters.
export COV=VICTOR           # prefix of the coverage files
export EFP=ExAC1/noTCGA     # use the ExAC provided by VICTOR, which is nonTCGA subset annotated with refGeneLite
export AWX="--xct-pop=NFE"  # compare to the NFE population only
```

Examples 6: variant classification

[Back to top]

You are a genetic counselor. A cancer patient came in and asked what caused her disease, and whether the disease will be transmitted to her son. You then sequenced a panel of high-penetrance genes known to be associated with the disease in the patient and her family members, some of whom were also affected with the same disease. You did find a rare variant in one of the known disease genes. This variant was not observed in any large cohorts, such as the UK10K, 1000 Genome and ExAC. It was a missense variant predicted to be damaging by most deleteriousness algorithms. By searching a variant database, you found that this variant was already observed in another family that was affected with the same disease. However, this variant was still a VUS (variant of unknown significance) according to the database. So you still had no answers to the patient's questions. If you do the variant classification analysis by pooling this new pedigree with existing data, you may have sufficient power to classify this VUS as either pathogenic or neutral.

Below are the commands to do the variant classification. Because this example does not contain a large-scale case-control data, the program vAAA is not used. By this analysis, the posterior probability of pathogenicity is 0.997, hence the variant is classified as "Pathogenic" following the guidelines of the International Agency for Research on Cancer (IARC).

```
# parameters in slurm.3_perch
AAA=--do-nothing
SEG=--vc
FIN=--vc
```

Examples 7: exome array

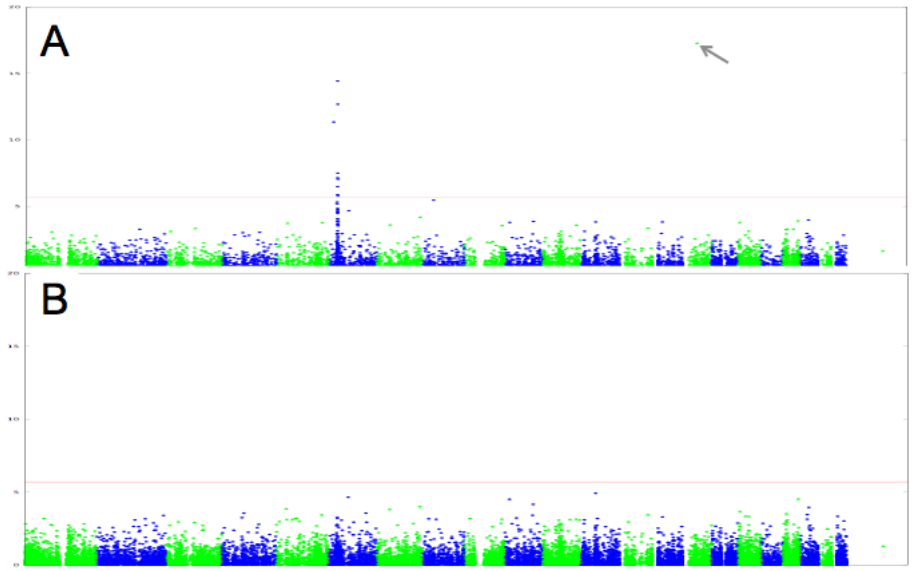
[Back to top]

The design of a customized exome array could include variants discovered from previous next-generation sequencing studies of the disease, tagging variants associated with similar diseases, functional variants in biological relevant genes, etc. The aim is to find causal variants with a low minor allele frequency (<0.05), but not too low (>0.005). The analysis is similar to a gene confirmation study.

```
# slurm.3_perch parameters for unadjustd analysis
SPL=ex3_raw.spl
AAA="--linear --out-mlp"
FIN="--plot-to=ex3_raw.pdf"

# slurm.3_perch parameters for adjusted analysis
SPL=ex3_cov.spl
AAA="--linear --out-mlp"
FIN="--plot-to=ex3_adj.pdf"
```

The first analysis conducts linear regression without covariates, while the second is adjusted for HLA-C*0602 (Cw6), an allele known to be associated with psoriasis, the disease of interest. For this purpose, ex3_raw.spl does not contain a covariate, while ex3_cov.spl has the Cw6 genotypes as a covariate. Below is the output Manhattan plots:



The upper panel is the Manhattan plot of raw test results, while the bottom one is adjusted for Cw6. As expected, the MHC region on chromosome 6 had quite a few significant results by raw test due to linkage disequilibrium (LD) with Cw6, which disappeared when the analysis was adjusted for Cw6. Interestingly, a gene named USP8 on chromosome 15 (pointed by an arrow) showed the same pattern. The p-value for this gene was 0.9 when adjusted for Cw6. Since USP8 and HLA-C are on different chromosomes, they should not be in LD, but the associated variant showed otherwise even in controls. Something's wrong here. Looking at the genome, there is a USP8 pseudogene, USP8P1, which has >94% sequence similarity with USP8, just 3.4kb upstream of HLA-C! So it is very likely that the exome array was designed to target a variant on chromosome 15, but mistakenly typed another variant on chromosome 6. This may be an example to show that, it is sometimes helpful to adjust for a known risk allele that is on a different chromosome.

Examples 8: Gene prioritization by network analysis only

[Back to top]

Suppose you have a number of seed genes and a number of candidate genes. Now you want to prioritize the candidate genes by gene-gene association network analysis. Below is the code. Beware, the seeds.txt follows the format of a Seed File (see PERCH's manual page for details), i.e., it has 2 columns: disease_name and gene_symbol. The disease_name will not be used by the gnGBA program, but they should be there and be the same for all lines.

```
gnGBA --gs -s seeds.txt | transpose | sort -k 1,1 > gba_scores.txt
join -a 1 <(sort candidates.txt) gba_scores.txt | sort -gr -k 2 > candidates_prioritized.txt
```

The output candidates_prioritized.txt contains all candidate genes sorted by GBA score (high to low). The gene symbols that were not found in the gene network database are listed at the bottom of the file, whose GBA scores are empty.

Examples 9: Annotation of functional consequence, MaxAF, and BayesDel

[Back to top]

Below are the procedures to annotate a VCF file for functional consequence, MaxAF and BayesDel scores.

Your input file My.vcf.gz must be compressed by bgzip and indexed by tabix. Copy slurm.all_steps to your work directory and set the parameter "VCF" to your input file.

```
export VCF=My.vcf.gz
```

If your VCF file is small (<1000 variants), you can set `export USE_TABIX=yes` to speed up. Submit a job to run this script by `victor.sbatch --array=1 slurm.all_steps`, or follow the instructions inside the script to run it directly. The final result file is VICTOR.ann.del.gz.