Leveraging External Allele Frequencies (LEAF) User Manual

Written by

Jiun-Sheng Chen and Chad Huff

Huff Lab

Department of Epidemiology

The University of Texas MD Anderson Cancer Center

Houston, Texas

Table of Contents

[1.Introduction: 3](#_Toc73517634)

[2. Software and data requirement: 4](#_Toc73517635)

[2.1. Required software tools 4](#_Toc73517636)

[2.2. Recommended software for preparation of genotype files 4](#_Toc73517637)

[2.3. External data 4](#_Toc73517638)

[3. LEAF workflow: 5](#_Toc73517639)

[3.1. External control allele frequency file 5](#_Toc73517640)

[3.2 Site frequency spectrum file 6](#_Toc73517641)

[3.3 Dynamic filtering thresholds 6](#_Toc73517642)

[3.4 Calculating variant effect size 7](#_Toc73517643)

[3.5 export a bed file to filter variants 8](#_Toc73517644)

# 1.Introduction:

Leveraging External Allele Frequencies (LEAF) is designed to enrich for rare disease-causing variation using the allele frequency of external controls to improve the power of association tests. Under the null model with LEAF, the expected allele frequency for a variant which is not associated with the disease of interest is the same among internal cases, internal controls, and external controls. Under the alternative model, disease-causing variants tend to be enriched among internal cases because combined case-control internal allele frequency distributions shift to higher allele frequencies relative to external controls.

LEAF detects and removes variants which are inconsistent with the alternative model by comparing allele frequencies in the internal dataset to the external controls in the absence of the disease phenotype information, enriching for rare variants that increase the risk of developing the disease. LEAF statistics are therefore orthogonal to statistics that are a function of internal case and control allele frequency data and thus can be utilized in a standard case-control study design without influencing type I error.

In single-marker tests, this filtering approach decreases multiple testing burden, and in gene-based tests, the approach increases the proportion of variants associated with the disease of interest. Both scenarios lead to an increase in statistical power. Because LEAF is applied prior to evaluation of the phenotype, it can be used with any existing case-control gene-based association test. This software is developed to support the article entitled: Leveraging external variant frequency information to increase the statistical power of rare variant association studies. Please refer to the paper submitted to ***Bioinformatics*** for more details.

# 2. Software and data requirement:

## 2.1. Required software tools

LEAF requires python 3.5 or greater. LEAF is available through git hub by git clone at the link below.

<https://github.com/roger894351/LEAF>

LEAF also requires two additional packages, bedtools to restrict variants in a vcf file and annovar to annotate the variant effect such as synonymous and nonsynonymous variants. These tools can be download from the following links:

Bedtools: <https://bedtools.readthedocs.io/en/latest/>

Annovar: <https://annovar.openbioinformatics.org/en/latest/>

## 2.2. Recommended software for preparation of genotype files

To prepare the input files for LEAF, we recommended using XPAT and VAAST to manipulate the file format transformation. For example, converting the genotype file from VCF format to CDR format; combining the internal case CDR and internal control CDR. VAAST and XPAT are available from the following links.

VAAST: https://hufflab.org/software/vaast/

XPAT: https://hufflab.org/software/xpat/

## 2.3. External data

LEAF supports allele frequency of external controls from summary data in a vcf file format. LEAF provides support to extract this information from the publicly available vcf file from gnomAD (https://gnomad.broadinstitute.org/downloads). (Please refer to 3.1 for the format and software usage). Users can also prepare the external allele frequency file as the format defined in 3.1.

# 3. LEAF workflow:

There are five steps in the LEAF workflow. First, LEAF exports allele frequencies from an external control dataset of the specified ancestry for the following steps. Second, LEAF calculates the allele frequency spectrum from the external controls to calibrate the joint allele frequency spectrum of the internal and external data. Third, LEAF sets dynamic allele frequency thresholds based on the user-selected option: 1) sensitivity equal to the specified threshold or 2) maximize accuracy while achieving a minimum sensitivity of at least the specified threshold. Fourth, LEAF calculates effect size (OR) between internal and external data. Fifth, LEAF exports a bed file for the subsequent association test based on the effect size and the dynamic thresholds level calculated in the fourth step. More details of each step were described below.

## 3.1. External control allele frequency file

**Description:**

LEAF takes the external allele counts to calculate the allele frequency of a variant for the LEAF framework. Currently, LEAF converts the allele frequency file of external controls (vcf format) from gnomAD or other sources with the same file format.

**Command:**

python extract\_count\_controls.py <vcf file> <out file name> <ancestry group> <subgroup>

**Input:**

<vcf file>: A file includes the allele frequency information of external controls. Such as, the vcf file from gnomAD.

<out file name>: The file name of the output file

<ancestry group> The defined ancestry group. In gnomAD, this includes the following: asj,asj\_female,oth\_female,female,nfe\_onf,male,sas\_female,amr\_male,fin\_male,nfe\_seu,eas,amr\_female,nfe\_female,asj, as\_jpn, afr\_male, afr, amr, nfe\_male, oth,oth\_male, afr\_female, nfe\_nwe, nfe\_bgr, eas\_female, asj\_male, nfe\_est, nfe, nfe\_swe, fin\_female,fin, eas\_oea, sas\_male, raw, eas\_male, sas, eas\_kor, popmax.

<subgroup>: The subgroups defined by gnomAD. such as non-cancers, controls

**Output:**

<out file>: allele frequency information from external controls of the desired population.

The script will export a tab delimitated file with allele frequencies of the desired control group with the following columns (for example, ancestry=nfe, subgroup=non-cancers)

The top ten lines from the example output file

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| #CHROM | POS | START\_POS | REF | ALT | non\_cancer\_AC\_ref | non\_cancer\_AC\_alt | non\_cancer\_AN\_nfe | non\_cancer\_AF\_nfe |
| 10 | 92946 | 92946 | C | T | 32957 | 1 | 32958 | 3.03E-05 |
| 10 | 92947 | 92947 | A | G | 33074 | 16 | 33090 | 4.84E-04 |
| 10 | 92952 | 92952 | A | G | 34841 | 1 | 34842 | 2.87E-05 |
| 10 | 92953 | 92955 | AAG | - | 34842 | 0 | 34842 | 0.00E+00 |
| 10 | 92955 | 92955 | G | C | 35503 | 1 | 35504 | 2.82E-05 |

## 3.2 Site frequency spectrum file

**Description:**

LEAF calculates the allele frequency spectrum from the external controls to calibrating the joint allele frequency spectrum of the internal and external data. LEAF calculates the allele frequency spectrum by including only nonsynonymous variants to account for the impact of purifying selection.

**Command:**

python site\_frequency\_spectrum.py <external allele frequency information> <annotated external allele frequency information >

**Inputs:**

<external allele frequency information >: The file includes the allele frequency information of external controls (output from 3.1) or user prepared file.

< annotated external allele frequency information > The file includes annotated variant categories by annovar or other annotation tools, including synonymous, nonsynonymous information.

**Output:**

LEAF will generate a site spectrum file from the external controls in the file name, <external allele frequency>\_out.txt.

## 3.3 Dynamic filtering thresholds

**Description**:

LEAF sets dynamic allele frequency thresholds based on the user-selected option: 1) sensitivity equal to the specified threshold or 2) maximize accuracy while achieving a minimum sensitivity of at least the specified threshold. Users can define multiple threshold levels in the parameter file, such as a sensitivity of 0.8 to detect an alternative model. In the manuscript, we use RR as 1.5 as the minimal detectable threshold as the sample size around 2,000 individuals each for a case-control study.

**Command:**

python set\_thresholds.py <site frequency spectrum file> <parameter file> <output file name>

**Inputs:**

<site frequency spectrum file>: the output from 3.2

<parameter file>: the file including the internal sample size(haploid), external sample size, and user-defined sensitivity thresholds, internal to external relative risk (default 1.5), maximum internal allele frequency.

Example of the parameter file:

external\_haploid\_sample\_size=102754

internal\_haploid\_sample\_size=3998

sensitivity\_thresholds=0.7,0.75,0.8,0.85,0.9,0.95,0.99

internal\_to\_external\_relative\_risk=1.5

max\_internal\_allele\_frequency=0.05

<output file name> The name of the output file includes the allele frequency threshold based on the parameter files. This file will be used in the last step.

**Output:**

<output file>: The output file includes the allele frequency threshold with multiple threshold levels from the parameter files. This file will be used in the last step for filtering variants based on the selected threshold.

## 3.4 Calculating variant effect size

**Description:**

LEAF takes the combined genotype information from the internal cases and controls in a CONDENSER (CDR) file format. Please find more information regarding the file format and tools converting vcf files to CDR format as following (https://hufflab.org/software/pvaast/, <https://hufflab.org/software/xpat/>)

LEAF also provides a weighting approach to calculate weights for every variant. The input weight file is a linear scale of weights for every variant. This feature is currently experimental.

**Command:**

Python Calculate\_af\_weight.py <genotype cdr> <external allele frequency information> <weighting file> <output file> <subgroup sample size[exome\_non\_cancer|exome\_control]> <allele frequency thresholds > <allele frequency level> <external allele frequency information>

**Input:**

<genotype cdr>: a genotype file in CDR format for internal cases and controls information

<external allele frequency information >: the output of 3.1

<weighting file>: user defined weight in linear scale. User can set all variants with value 0 for equal weighting for each variant. This weighting approach does not affect the allele frequency filtering approach. The weighting format is bed file format following variant types(SNV, INDEL, SPDA) and weighting.

Example of Casm score weight file:

chr1 876606 876606 SNV 0

<output file>: the name of the output file

<subgroup sample size[exome\_non\_cancer|exome\_control]>: The average detectable individuals of all variants in the sub-group of controls for non-observed variants. Currently, LEAF only support “exome\_non\_cancer” or “exome\_control” sub-groups. Please provide average individuals of all variants other than these two sub-group.

<dynamic allele frequency thresholds >: the output file of 3.3

<allele frequency level> : a desired allele frequency threshold such as 0.9, low\_prior\_0.8

<external allele frequency information> : the output file of 3.1

**Output:**

<output file>

LEAF generates a file with the effect size and weights of all variants between internal and external data.

## 3.5 export a bed file to filter variants

**Description:**

This script creates a bed file base on the effect size from step 3.4 based on the selected threshold level.

**Command:**

python create\_bed\_threshold.py <quality control bed> <allele frequency ORfile> <dynamic allele frequency thresholds <selected threshold> <nonsense variant bed> -a

**Input:**

<quality control bed>: a bed file listing variant sites that failed quality control (QC)

<allele frequency OR file>: the output from step 3.4

<dynamic allele frequency thresholds >: the output file of step 3.3

<selected threshold>: selected allele frequency threshold in a sensitivity level, which is the same as the parameter defined at step 3.3, e.g., 0.9 or the combination of max accuracy and sensitivity level defined with the prefix ”low\_prior" e.g., low\_prior\_0.8, low\_prior\_0.9

<nonsense variant bed>: a bed file including nonsense variants and other variants to be excluded from LEAF filtering.

**Output:**

The output file lists variants filtered by LEAF, which are not likely disease-associated. Variants that failed QC are also included. This bed file can be directly applied in association testing using the internal case-control data.