RAREsim Vignette

Megan Null

7/18/2020

This vigette describes how to use the RAREsim R package to simulate rare variant genetic data.

Here we will walk through an example using RAREsim to simulate one cM block on chromosome 19, to match the data from the African ancestry group from gnomAD v2.1 (Karczewski, et al., 2020).

## Install the package

library(RAREsim)

The source code for all functions within the RAREsim package can be found at <https://github.com/meganmichelle/RAREsim_package>. The package currently must be downloaded through github using devtools.

RAREsim has three main steps: (1) simulate genetic data with an abundance of rare variants using [HAPGEN2](https://mathgen.stats.ox.ac.uk/genetics_software/hapgen/hapgen2.html) (Su, 2011), (2) estimate the expected number of variants in MAC bins, and (3) probabilistically prune the rare variants to match the estimated number of variants in each MAC bin.

An example simulation with HAPGEN2 can be found on the [RAREsim github page](https://github.com/meganmichelle/RAREsim/blob/master/HAPGEN2_simulation_example). By simulating with default parameters and input haplotypes with information at all sequencing bases, including monomorphic sites, HAPGEN2 simulates an abundance of rare variants.

In order to emulate real sequencing data, RAREsim prunes the simulated variants by returning all or a subset of alternate alleles back to reference. In order to prune, RAREsim first estimates the expected number of variants within MAC bins. The number of variants in each MAC bin can either estimated using default parameters, modifying default parameters, or fitting target data. Additionally, if the exact sample size of observed sequencing data is to be simulated, the observed data can be matched directly.

Here we will demonstrate fitting target data as well as using default parameters.

## Fitting the Variants per Kb function

For a given region, the *Variants per Kb* function estimates the number of variants per Kb, , for a sample size . This is done by estimating and to optimize the function to fit the target data.

The Variants per Kb target data consists of various sample sizes () and the observed number of variants per Kb in the region of interest. Ancestry specific data is advised. Data should be formatted with the first column as the number of individuals () and the second column as the observed number of variants per Kb in the region of interest ().

Here we will fit the example data for the African ancestry population. Example data is available in the R package for each of the four ancestries: African (AFR), East Asian (EAS), Non-Finnish European (NFE), and South Asian (SAS).

# load the data  
data("var\_per\_kb\_afr")  
print(var\_per\_kb\_afr, row.names = FALSE)

## n per\_kb  
## 10 0.2627568  
## 20 0.6831678  
## 50 1.5239897  
## 100 2.7326712  
## 200 4.3092123  
## 500 7.6199485  
## 1000 12.1919176  
## 2000 19.3914551  
## 3070 25.2246571  
## 5000 33.4226707  
## 5040 33.7905302  
## 8128 45.1941773

The target data is used to estimate and within a least squares loss function, optimizing using sequential quadratic programming (SQP). This optimization is implemented via the *Fit\_fvar* function.

nvar <- Fit\_fvar(var\_per\_kb\_afr)  
nvar

## $phi  
## [1] 0.1638108  
##   
## $omega  
## [1] 0.6248848

The output of the *Fit\_fvar* function are the parameters phi () and omega (), respectively. The estimated parameters can then be used to determine the expected number of variants per Kb within the region of interest, given the number of individuals to be simulated, .

For example, to simulate the sample size observed in the target data, (), we calculate . This can be done with the *Variants\_per\_Kb* function. Parameter values for phi (), omega (), and the sample size (n) are required.

Variants\_per\_Kb(phi = nvar$phi, omega = nvar$omega, n = 8128)

## [1] 45.46027

Above, the number of variants per Kb was determined using parameters estimated from target data. However, RAREsim also provides ancestry specific default parameters that can be used instead. To use the default parameters, the ancestry must be specified: African (AFR), East Asian (EAS), Non-Finnish European (NFE), or South Asian (SAS).

Variants\_per\_Kb(n=8128, pop = 'AFR')

## [1] 43.66395

The example data here is a cM block with 19,029 bp. Thus, to calculate the total expected number of variants in the region, we multiple the expected number of variants per Kb (*Variants\_per\_Kb*) by 19.029.

19.029\*Variants\_per\_Kb(phi = 0.1638108, omega = 0.6248848, n = 8128)

## [1] 865.0633

At this point, we have estimated the total number of variants within the region. We now need to estimate parameters for the *Allele Frequency Spectrum (AFS)* function to estimate the proportion of variants within MAC bins.

## Fitting the Allele Frequency Spectrum function

The *AFS* function inputs a MAC (*z*) and outputs the proportion of variants at MAC = z, (). This is done by estimating and to optimize the function . Here *b* ensures that the sum of the individual rare allele count proportions equals the total proportion of rare variants, .

The *AFS* function inputs a data frame with the upper and lower boundaries for each bin and proportion of variants within each respective bin. The default bins used here and within the evaluation of RAREsim are:

MAC = 1  
MAC = 2  
MAC = 3 - 5  
MAC = 6 - 10  
MAC = 11 - 20  
MAC = 21 - MAF = 0.5%  
MAC = 0.5% - MAF = 1%

Below is an example of the AFS target data for the African ancestry group. The first two columns identify the lower and upper boundaries of each MAC bin. The third column specifies the observed proportion of variants within each MAC bin in the target data.

# load the data  
data("afs\_afr")  
print(afs\_afr)

## Lower Upper prop  
## 1 1 1 0.50257998  
## 2 2 2 0.16305470  
## 3 3 5 0.08255934  
## 4 6 10 0.05882353  
## 5 11 20 0.03715170  
## 6 21 81 0.05675955  
## 7 82 162 0.01754386

To fit the *AFS* function (*Fit\_AFS*), RAREsim requires the data frame with MAC bins and proportion of variants (shown above), the number of subjects to simulate , and the total proportion of rare variants, . Here, we will simulate the sample size observed in gnomAD, with 97% of variants assumed to be rare, .The function estimates the parameters alpha (), beta (), and , and includes the estimated proportion of variants based on calculations from the fitted parameters, as shown below.

af <- Fit\_AFS(prop\_df = afs\_afr, N = 8128, p\_rv = 0.97)  
print(af)

## $alpha  
## [1] 1.531338  
##   
## $beta  
## [1] -0.3090162  
##   
## $b  
## [1] 0.2926182  
##   
## $Fitted\_results  
## Lower Upper Fitted\_prop  
## 1 1 1 0.515383183430996  
## 2 2 2 0.130900952615951  
## 3 3 5 0.131313223208255  
## 4 6 10 0.0689151687234886  
## 5 11 20 0.0488202660410061  
## 6 21 81 0.0582841702824958  
## 7 82 162 0.0163830356978067

As with the *Variants per Kb* function, default parameters can be used to estimate the parameters for the *AFS* function with the *AFS\_calc* function. As the default parameters are ancestry specific, the ancestry needs to be specified as pop = AFR, EAS, NFE, or SAS when default parameters are used. The parameters alpha (), beta (), and b can be specified, or default parameters can be used. Both implementations of the function require a MAC bin dataframe, with the bins specified.

This is the first two columns of the AFS target data.

mac <- afs\_afr[,c(1:2)]  
mac

## Lower Upper  
## 1 1 1  
## 2 2 2  
## 3 3 5  
## 4 6 10  
## 5 11 20  
## 6 21 81  
## 7 82 162

Using the MAC bins as input and specifying an African ancestry, the default parameters are used below to estimate the proportion of variants within each bin.

AFS\_calc(mac = mac, pop = 'AFR')

## Warning in if (colnames(mac == c("Lower", "Upper")) == FALSE) {: the condition  
## has length > 1 and only the first element will be used

## Lower Upper Fitted\_prop  
## 1 1 1 0.515735497258123  
## 2 2 2 0.124612409098275  
## 3 3 5 0.120339057680533  
## 4 6 10 0.0604721589411936  
## 5 11 20 0.0412275867360237  
## 6 21 81 0.0465828339399243  
## 7 82 162 0.0122941803088416

print(mac)

## Lower Upper  
## 1 1 1  
## 2 2 2  
## 3 3 5  
## 4 6 10  
## 5 11 20  
## 6 21 81  
## 7 82 162

## Expected Number of Variants per MAC bin

mac$Lower <- as.numeric(as.character(mac$Lower))  
mac$Upper <- as.numeric(as.character(mac$Upper))  
mac

## Lower Upper  
## 1 1 1  
## 2 2 2  
## 3 3 5  
## 4 6 10  
## 5 11 20  
## 6 21 81  
## 7 82 162

Using the parameter estimates from the *Variants per Kb* and *AFS* functions, the expected number of variants in each MAC bin can be estimated. Direction using the estimated parameters is demonstrated below.

bin\_estimates <- Expected\_variants(alpha = af$alpha , beta = af$beta, b = af$b, phi = nvar$phi , omega = nvar$omega , Ntar = 8128, Size = 19.029, mac = mac)  
print(bin\_estimates)

## Lower Upper Expected\_var  
## 1 1 1 445.83914  
## 2 2 2 113.23763  
## 3 3 5 113.59427  
## 4 6 10 59.61599  
## 5 11 20 42.23263  
## 6 21 81 50.41950  
## 7 82 162 14.17236

We can also estimate the number of variants per MAC bin using the default parameters, again specifying the ancestry population as AFR, EAS, NFE, or SAS. Here the default parameters for both the *AFS* and *Variants per Kb* functions were used.

bin\_estimates <- Expected\_variants(pop='AFR' , Ntar = 8128, Size = 19.029, mac=mac)  
print(bin\_estimates)

## Lower Upper Expected\_var  
## 1 1 1 428.53078  
## 2 2 2 103.53268  
## 3 3 5 99.97549  
## 4 6 10 50.23531  
## 5 11 20 34.24618  
## 6 21 81 38.69085  
## 7 82 162 10.21018

The output of the *Expected\_variants* function is the exected number of variants in each MAC bin within the simulation region. This output (shown above) is input for the pruning function.

## Pruning Variants

In order to use RAREsim to prune simulated data, genetic data must be simulated with HAPGEN2 with all sequencing bases, including monomorphic variants, added to the input haplotypes. HAPGEN2 will simulate an abudance of rare variants to allow for variant pruning. Additionally, a MAC file (count of the number of alternate alleles at each bp) enables an efficient and fast pruning process. It is recommended to create the MAC file within the process of simulating data with HAPGEN2, as shown in the example code that is available on the [RAREsim github page](https://github.com/meganmichelle/RAREsim/blob/master/HAPGEN2_simulation_example).

Below is an example MAC file created from the haplotypes simulated for the African ancestry group and the region of interest. Each row represents one bp in the haplotype file.

data("MAC\_afr")  
dim(MAC\_afr)

## [1] 19029 1

head(MAC\_afr)

## V1  
## 1 0  
## 2 2  
## 3 0  
## 4 0  
## 5 0  
## 6 0

Pruning variants requires a MAC file from the simulated data, the expected number of variants within each MAC bin (product of the *Expected\_variants* function), and the name and location of the gzipped haplotype file. Additionally, here it is specified that the computer running the pruning algorithm is a mac.

When running the pruning function, the haplotypes files are rewritten. Thus, if the original files are needed, a copy needs to be made prior to pruning.

# Pruning\_function(hap\_file\_name = '/Users/megansorenson/Documents/RAREsim/Example/Block37\_rep1.controls.haps.gz', MAC = MAC\_afr, expected = bin\_estimates, computer\_type = 'mac')

The output within R states the number of variants that were pruned. The haplotype files have been pruned and now have the expected number of variants per MAC bin.

Once the pruning process is complete, RAREsim has produced a haplotype file with simulated data that on average matches what is expected based on the *Expected\_variants* input. The haplotypes emulate real data with respect to the total number of variants, AFS, and haplotype structure. Variant annotation of any type can be easily added to the simulated data.