# RVsharing R package

Prepared by Alexandre Bureau

## Input to the main function RVsharing

* **Pedigree description:** the function should accept either three vectors of ID, father ID and mother ID, or a pedigree object, or the object returned by a previous call to RVsharing.
* **Information on relatedness between founders**: 1) relfounders, a vector specifying the ids of the founders who are related. If it is missing, then all founders are assumed to be related. 2) phi, either a scalar giving the constant kinship coefficient between the founders in relfounders, or a square symmetric matrix of kinship coefficients for every pair of founders in relfounders. Default is phi = 0.

### Additional input in the future

* **Sequenced affected subjects**: for now RVsharing assumes that all final descendants are sequenced affected subjects, and the pedigree needs to be trimmed beforehand. In the future the function should handle sequenced affected subjects who are not final descendants.
* **Genotype of the sequenced subjects**: for now RVsharing assumes only final descendants have been sequenced and their genotype is heterozygote for the variant. In the future the genotype of all sequenced subjects (which could include unaffected subjects) should be passed to the function, coded such that say, allele 2 is the rare allele and allele 1 the common allele.

### Future higher level functions for data processing and gene level analysis

A function to extract variant data in the Variant Call Format (VCF) and other variant genotype format, and recode the rare and common allele for input to RVsharing.

The genomic position of the variants in the sequence will be used to retrieve the gene or genes to which they belong using the Gencode (gencodegenes.org) reference annotation and their frequency in appropriate reference populations from the most comprehensive public databases of sequence data. For exonic variant frequencies, it is currently the NHLBI GO Exome Sequencing Project database (evs.gs.washington.edu/EVS/). For the frequency of genomic variants outside exons, the 1000 Genomes project (www.1000genomes.org) data will be used.

## Algorithm to decide which approximation method to use to account for unknown relationships

If phi = 0, then apply the standard approach with not approximation

If phi is a square symmetric matrix of kinship coefficients, or phi is a constant but relfounders does not contain all founders, then apply approximation method 1.

If phi is a positive constant and relfounders contains all founders or is missing, then apply approximation method 2.

## Implementation of approximation methods 1 and 2

In common to the two methods are the sharing probabilities conditional on the introduction of the RV by two of the founders.

The implementation needs to be recursive to be applicable to general non-inbred pedigree structures. The recursion is more complex than for the computation with introduction of the variant by a single founder currently implemented because we need to keep track of pairs of founders and not only individual founders above each sequenced subject. Within a subpedigree where all sequenced subjects are related to founders through independent lines of descents, the formulas in sections A.1 to A.3 apply to different pairs of founders. Section A.4 of the manuscript gives an outline of how to proceed to adapt the recursive formulas to process successive subpedigrees defined by branching individuals. The best way to implement it requires more thinking.

For method 1, the terms corresponding to the introduction of the variant by a single founder (event FU) in the numerator and denominator of equation 10 are computed by the function RVsharing.weighted, where the P[FUj] are computed taking the kinship of specific founder pairs into account using equation 12. Once the terms corresponding to the introduction of the variant by a pair of founders are implemented, the result of equation 10 will need to be computed within RVsharing.

For method 2, the terms corresponding to the introduction of the variant by a single founder (event FU) in the numerator and denominator of equation 22 are computed by the function RVsharing.approx2. P[FUj] is a constant P[FU] for all founders. It is computed by sequentially calling the functions compute.phi.vec, one of the functions infer.theta\* and the corresponding function PFU.direct\* depending on the order of the polynomial approximation to the distribution of the number of distinct alleles. The computation of PFU needs to be integrated into a single function. Once the terms corresponding to the introduction of the variant by a pair of founders are implemented, the result of equation 22 will need to be computed within RVsharing.

## Output

The elements currently returned by RVsharing are described in RVsharing.Rd. The elements iancestors and desfounders serve as input to RVsharing.weighted and RVsharing.approx2 and could be removed once the calls to these functions are made internal to RVsharing. The essential output is p.share: the probability that all final descendants in the pedigree share a rare variant given that a rare variant has been detected in any one of these final descendants.