**Power and sample size analysis for two-way admixture mapping**

The descriptions are based on the PAMAM R functions for the power and sample size in three-way admixture mapping studies. The source codes can be downloaded from the github page <https://github.com/MershaLab/PAMAM/blob/master/AMPower_2way.R>.

* Discrete phenotype with genotype risk ratio (GRR)

PowerDiscreteGRR(admix.prop, grr, px, py, case.n, control.n, type1.error, side, study.design)

SampleDiscreteGRR(admix.prop, grr, px, py, type1.error, type2.error, side, study.design)

The arguments in the functions are:

* admix.prop = Proportion of genome from the population X
* grr = genotype risk ratio under multiplicative mode
* px = allele frequency of risk allele in the population X
* py = allele frequency of risk allele in the population Y
* case.n = number of case sample
* control.n = number of control samples (= 0, in case only study)
* type1.error = Type 1 error rate after the adjustment for multiple testing, such as after Bonferroni correction
* type2.error = Type 2 error rate
* side = 1 for one-sided test, = 2 for two-sided test
* study.design = 1 for case-only, = 2 for case-control
* Discrete phenotype with parental risk ratio (PRR)

PowerDiscretePRR(generation, recom.rate, admix.prop, prr, case.n, control.n, type1.error, side, study.design, mode, process)

SampleDiscretePRR(generation, recom.rate, admix.prop, prr, type1.error, type2.error, side, study.design, mode, process)

The arguments in the functions are:

* + generation = # of generation since admixture
  + recom.rate = recombination rate between the disease locus and marker locus
  + admix.prop = Proportion of genome from the Population X
  + prr = parental risk ratio
  + case.n = # of case samples
  + control.n = # of control samples ( =0 for case-only studies)
  + type1.error = Type I error rate after adjustment for multiple testing
  + type2.error = Type II error rate
  + side = 1 for one-sided test, = 2 for two-sided test
  + study.design = 1 for case-only design, = 2 for case-control design
  + mode = ‘multi’ for multiplicative mode,

= ‘add’ for additive mode,

= ‘rec’ for recessive mode, and

= ‘dom’ for dominant mode

* + process = ‘HI’ for hybrid-isolation process,

= ‘CGF’ for Continuous Gene Flow Admixture process

* Discrete phenotype with ancestral odds ratio (AOR)

PowerDiscreteAOR(admix.prop, aor, case.n, control.n, type1.error, side)

SampleDiscreteAOR(admix.prop, aor, type1.error, type2.error, side)

The arguments in the functions are:

* + admix.prop = Proportion of genome from the Population X
  + aor = ancestry odds-ratio per 1 parental chromosome from the population X
  + case.n = number of case sample
  + control.n = number of control samples
  + type1.error = Type 1 error rate after the adjustment for multiple testing, eg after Bonferroni correction
  + type2.error = Type 2 error rate
  + side = 1 for one-sided test, = 2 for two-sided test
* Quantitative Trait with effect as the regression coefficient

PowerQTraitCoeff(sigma.u, coeff.u, sample.size, type1.error, r2.covariates, sigma.error, side)

SampleQTraitCoeff(sigma.u, coeff.u, type1.error, type2.error, r2.covariates, sigma.error, side)

The arguments in the functions are:

* Sigma.u = standard deviation of the ancestry variable u
* Coeff.u = Coefficient of the variable u under alternate hypothesis
* sample.size = Sample size
* r2.covatiates = Multiple R-square from a linear fit of the variable u against all other covariates in the model(aka the inflation factor)
* sigma.error = Standard deviation of the error term
* type1.error = Type I error after adjustment of multiple testing
* type2.error = Type II error rate
* side = 1 if one-sided test, = 2 if two-sided test
* Quantitative Trait with effect as R-square

PowerQTraitRSquare(r.square, sample.size, type1.error, side)

SampleQTraitRSquare(r.square, type1.error, type2.error, side)

The arguments in the functions are:

* r.square = r-square value (in the percentage form)
* sample.size = Sample size
* type1.error = Type I error rate after adjustment for multiple testing
* type2.error = Type II error rate
* side = 1 for one-sided test, = 2 for two-sided test