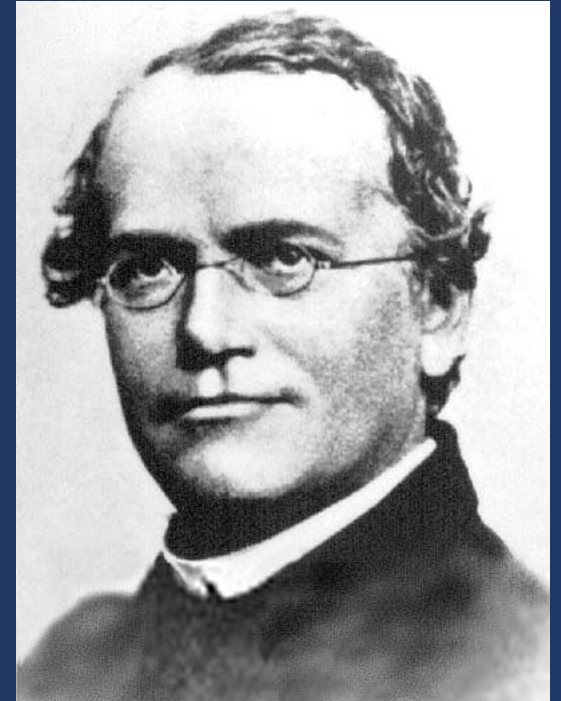


VarianceComponent Models.jl



OpenMendel Workshop
ASHG Annual Meeting 2020
Juhyun Kim

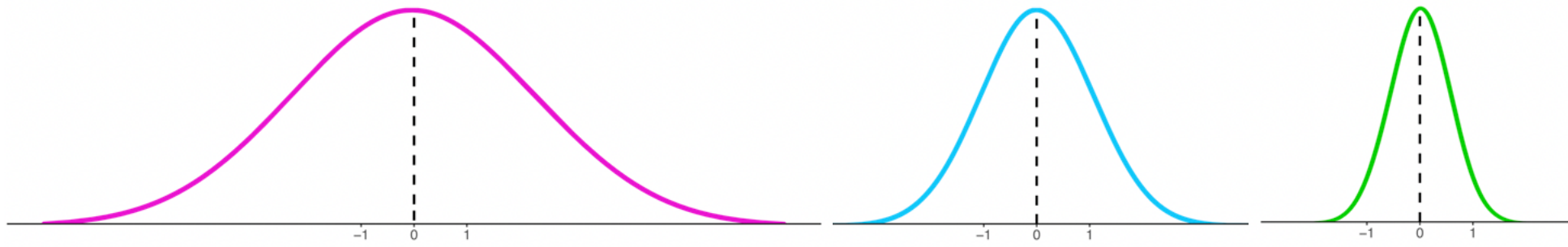
Variance component methods in genetics

Variance component methods are used

- for genetic analysis of quantitative traits (e.g. BMI, cholesterol)
- to assess the strength of genetic effects on a trait
- to characterize the genetic effects on a trait through analyses of gene-gene and gene-environment interaction
- to localize genes influencing a trait through either linkage/association methods
- to explore whether related traits have shared genetic influences in multivariate analyses

Idea behind variance component methods

Idea: decompose overall variance in a phenotype into particular sources



$$\begin{array}{ccccccc} \text{Phenotypic variance} & = & \text{Genetic variance} & + & \text{Environmental variance} \\ \sigma_p^2 & = & \sigma_g^2 & + & \sigma_e^2 \end{array}$$



VarianceComponentModels.jl

Utilities for fitting and testing variance component models

$$Y \sim \text{Normal}(XB, \sigma_1^2 V_1 + \dots + \sigma_m^2 V_m)$$

In this model, **data** is represented by

- Y : continuously varying quantitative trait(s)
- X : covariates (e.g. sex, age, principal components)
- V_i : structuring matrix corresponding to i -th variance component ($i = 1, \dots, m$)

and **parameters** are

- B : mean fixed effects coefficient
- σ_i^2 : i -th variance component ($i = 1, \dots, m$)

Feature 1: heritability analysis

Heritability: proportion of the phenotypic variance in a trait that is attributable to genetic effects

1. Additive genetic effect (σ_a^2), unique/unshared environmental effect (σ_e^2)

$$\Omega = \sigma_a^2(2\Phi) + \sigma_e^2 I, \quad h^2 = \frac{\sigma_a^2}{\sigma_a^2 + \sigma_e^2}$$

```
data1 = VarianceComponentVariate(pheno, X, (2Φgrm, Matrix(1.0I, nobs, nobs)))
```

```
# heritability and its standard error  
h, hse = heritability(trait_model.Σ, Σcov)  
[h[1], hse[1]]
```

```
2-element Array{Float64,1}:  
0.5735231845909365 ← heritability estimate  
0.2564683288674781 ← standard error
```

Feature 1: heritability analysis

2. Additive genetic effect (σ_a^2), unique/unshared environmental effect (σ_e^2), shared household effect (σ_h^2)

$$\Omega = \sigma_a^2(2\Phi) + \sigma_h^2\Psi_h + \sigma_e^2I$$

```
data2 = VarianceComponentVariate(pheno, X, (2Φgrm, Ψhousehold, Matrix(1.0I, nobs, nobs)))
```

3. Additive genetic effect (σ_a^2), unique/unshared environmental effect (σ_e^2), shared household effect (σ_h^2), sibling effect (σ_{SP}^2), maternal effect (σ_M^2)


$$\Omega = \sigma_a^2(2\Phi) + \sigma_h^2\Psi_h + \sigma_{SP}^2\Delta + \sigma_M^2\Lambda + \sigma_e^2I$$


```
data3 = VarianceComponentVariate(pheno, X,  
    (2Φgrm, Ψhousehold, Δsp, Λm, Matrix(1.0I, nobs, nobs)))
```


Feature 2: multivariate analysis

When multiple traits are known to be correlated, we want to know whether they are influenced by the same genes

- e.g. hypertension, abdominal obesity, high triglyceride levels, low HDL cholesterol levels

 Heritability

 Search

 GitHub
6 Stars · 5 Forks

[VarianceComponentModels.jl](#)
[Home](#)
[Manual ^](#)
 [MLE/REML](#)
 [Heritability](#)
[API](#)

3-trait analysis

Researchers want to jointly analyze traits 5-7. Our strategy is to try both Fisher scoring and MM algorithm with different starting point, and choose the best local optimum. We first form the data set and run Fisher scoring, which yields a final objective value -1.4700991+04.

```
traitidx = 5:7
# form data set
trait57_data = TwoVarCompVariateRotate(cg10kdata_rotated.Yrot[:, traitidx], cg10kdata_rotated.eigval, cg10kdata_rotated.eigvec, cg10kdata_rotated.logdet)
# initialize model parameters
trait57_model = VarianceComponentModel(trait57_data)
# estimate variance components
@time mle_fs!(trait57_model, trait57_data; solver=:Ipopt, verbose=true)
trait57_model
```

This is Ipopt version 3.12.4, running with linear solver mumps

Table of contents

- [Read in binary SNP data](#)
- [Summary statistics of SNP data](#)
- [Empirical kinship matrix](#)
- [Phenotypes](#)
- [Pre-processing data for heritability analysis](#)
- [Save intermediate results](#)
- [Heritability of single traits](#)
- [Pairwise traits](#)
- [3-trait analysis](#)
- [13-trait joint analysis](#)
- [Save analysis results](#)

Feature 3: association analysis

- Test whether the mean trait values differ by genotype
 - Likelihood ratio test
- Set up the null model without SNPs

```
X = [ones(nobs) age sex PC1 PC2 PC3 PC4]
nulldata = VarianceComponentVariate(pheno, X,
                                     (2Φgrm, Matrix(1.0I, nobs, nobs)))
nullmodel = VarianceComponentModel(nulldata)
@time nulllogl, nullmodel, = fit_mle!(nullmodel, nulldata; algo=:FS)
```

- Set up the full model with SNPs

```
snp_mat = convert(Vector{Float64}, EUR_subset[:, 10:20])
Xalt = [ones(nobs) age sex PC1 PC2 PC3 PC4 snp_mat]
altdata = VarianceComponentVariate(pheno, Xalt,
                                     (2Φgrm, Matrix(1.0I, nobs, nobs)))
altmodel = VarianceComponentModel(altdata)
@time altlogl, altmodel, = fit_mle!(altmodel, altdata; algo=:FS)
```


Feature 3: association analysis

- Likelihood ratio test (LRT) to test the goodness-of-fit between two models

```
using Distributions  
LRT = 2*(altlogl - nulllogl)
```

```
11.639045364202502
```

- The associated p -value

```
pval = ccdf(Chisq(11), LRT)
```


```
0.3913771524949879
```


Other features


- Choice of algorithms
 - Fisher-scoring algorithm
 - Minorization-maximization (MM) algorithm
- Constrained estimation of the mean parameters

```
fit_mle!(model, data; algo=:FS)
```

```
fit_mle!(model, data; algo=:MM)
```

 MLE/REML

 Search

 GitHub
6 Stars · 5 Forks

VarianceComponentModels.jl

Home

Manual ^

MLE/REML

Heritability

API

Constrained estimation of B

Many applications invoke constraints on the mean parameters B . For demonstration, we enforce $B[1,1]=B[1,2]$ and all entries of B are within $[0, 2]$.

```
# set up constraints on B
vcmodel_constr = deepcopy(vcmodel)
vcmodel_constr.A = [1.0 0.0 -1.0 0.0]
vcmodel_constr.sense = '='
vcmodel_constr.b = 0.0
vcmodel_constr.lb = 0.0
vcmodel_constr_ub = 2.0
```

Table of contents

Demo data

Maximum likelihood estimation (MLE)

Restricted maximum likelihood estimation (REML)

Optimization algorithms

Starting point

Constrained estimation of B

For more
information

- VarianceComponentModels.jl documentation:
<https://openmendel.github.io/VarianceComponentModels.jl/latest/>
- Jupyter notebook tutorial:
<https://github.com/OpenMendel/ASHG-OpenMendelWorkshop-2020-Oct/tree/master/05-VarianceComponent-Kim>