Trait simulation

Huwenbo Shi

August 15, 2016

1 Model (GLM / GLMM)

$$\boldsymbol{Y} \sim \text{Dist}\left(\text{Link}^{-1}\left(\boldsymbol{\mu} = \boldsymbol{X\beta} + \sum_{i=1}^{k} \boldsymbol{Z}_i \boldsymbol{u}_i\right), \text{ other distribution-specific parameters}\right)$$

- $\boldsymbol{X\beta}$: Fixed effect component
- $\sum_{i=1}^{k} \boldsymbol{Z}_i \boldsymbol{u}_i$: k random effect component, k can be 0
- If k is non-zero, then draw $\boldsymbol{\mu}$ from $N(\boldsymbol{X\beta}, \sum_{i=1}^{k} \boldsymbol{C}_i \otimes \boldsymbol{A}_i)$, where $\boldsymbol{C}_i$ are the cross covariance matrices, and $\boldsymbol{A}_i$ are the covariance matrices.
- Other distribution-specific parameters can be σ^2 for Gaussian distribution, N for binomial distribution, etc.

2 Implementation

```
"""
A type to store simulation parameters.
"""
type Model
  """
  Specify which distribution of the response:
    1) Binomial 2) Gamma 3) Normal 4) Poisson 5) Exponential
    6) Inverse Gaussian 7) Bernoulli etc.
  """
  distribution::AbstractString

  """
  Additional parameters for the distribution, e.g. variance for normal,
  N for binomial, etc.
  """
  parameters::Vector{Float64}

  """
```

```
  Specify the link function, GLM.jl currently supports:
  1) CauchitLink 2) CloglogLink 3) IdentityLink 4) InverseLink
  5) LogitLink 6) LogLink 7) ProbitLink 8) SqrtLink
  """
  link::AbstractString

  """
  Specify the formula of the simulation, e.g. TC ~ AGE + SNP1*SNP2 + HDL
  Using Formula of DataFrame.jl?
  """
  formula::Formula

  """
  Coefficient for each term in the formula
  """
  coefficients::Vector{Float64}

  """
  Intercept term
  """
  intercept::Float64

  """
  Variables whose effects are random
  """
  random_effect_variables::Array{Symbol}

  """
  Variance components
  """
  variance_components::Vector{Float64}

  """
  Cross covariances for each of the covariance matrix
  """
  cross_covariances::Array{Matrix{Float64}}
end

"""
Simulate traits based on model specified in 'model' using data
stored in 'data_frame'.
"""
function simulate(model::Model, data_frame::DataFrame)
  # TODO: implement this function
end
```