

Simulation of data using Julia

Rohan L. Fernando

May 2015

Julia Packages

- [List of registered Julia packages](http://docs.julialang.org/en/release-0.1/packages/packagelist/#available-packages) (<http://docs.julialang.org/en/release-0.1/packages/packagelist/#available-packages>)
- Will use [Distributions Package](http://distributionsjl.readthedocs.org/en) (<http://distributionsjl.readthedocs.org/en>) to simulate data.
- It can be added to your system with the command:

```
In [1]: Pkg.add("Distributions")
```

```
INFO: Nothing to be done
```

```
INFO: METADATA is out-of-date — you may not have the latest version of Distributions
```

```
INFO: Use `Pkg.update()` to get the latest versions of your packages
```

- This needs to be done only once.
- But, to access the functions in the Distributions package the "using" command has to be invoked as:

```
In [2]: using Distributions
```

Simulate matrix of ``genotype'' covariates

```
In [3]: nRows = 10  
nCols = 5  
X = sample([0,1,2],(nRows,nCols))
```

```
Out[3]: 10x5 Array{Int64,2}:  
 0  0  2  1  2  
 1  1  1  1  0  
 0  2  1  1  0  
 1  2  1  0  2  
 0  1  1  0  2  
 2  1  0  0  2  
 2  2  2  0  2  
 1  2  2  2  1  
 0  0  2  2  1  
 2  2  0  2  0
```

Each element in **X** is sampled from the array [0,1,2].

Other methods of the function ``sample``

```
In [4]: methods(sample)
```

```
Out[4]: 7 methods for generic function sample:
```

- `sample(a::AbstractArray{T,N})` at [/Users/rohan/.julia/v0.3/StatsBase/src/sampling.jl:277](#)
(<https://github.com/JuliaStats/StatsBase.jl/tree/23b36af460cfd6c147efef9074a76a4e8cf13dae/src/>)
- `sample{T}(a::AbstractArray{T,N},n::Integer)` at [/Users/rohan/.julia/v0.3/StatsBase/src/sampling.jl](#)
(<https://github.com/JuliaStats/StatsBase.jl/tree/23b36af460cfd6c147efef9074a76a4e8cf13dae/src/>)
- `sample{T}(a::AbstractArray{T,N},dims::(Int64...,))` at [/Users/rohan/.julia/v0.3/StatsBase/src/sampling.jl](#)
(<https://github.com/JuliaStats/StatsBase.jl/tree/23b36af460cfd6c147efef9074a76a4e8cf13dae/src/>)
- `sample(wv::WeightVec{W,Vec<:AbstractArray{T<:Real,1}})` at [/Users/rohan/.julia/v0.3/StatsBase/src/sampling.jl](#)
(<https://github.com/JuliaStats/StatsBase.jl/tree/23b36af460cfd6c147efef9074a76a4e8cf13dae/src/>)
- `sample(a::AbstractArray{T,N},wv::WeightVec{W,Vec<:AbstractArray{T<:Real,1}})` at [/Users/rohan/.julia/v0.3/StatsBase/src/sampling.jl:347](#)
(<https://github.com/JuliaStats/StatsBase.jl/tree/23b36af460cfd6c147efef9074a76a4e8cf13dae/src/>)
- `sample{T}(a::AbstractArray{T,N},wv::WeightVec{W,Vec<:AbstractArray{T<:Real,1}},n::Integer)` at [/Users/rohan/.julia/v0.3/StatsBase/src/sampling.jl:529](#)
(<https://github.com/JuliaStats/StatsBase.jl/tree/23b36af460cfd6c147efef9074a76a4e8cf13dae/src/>)
- `sample{T}(a::AbstractArray{T,N},wv::WeightVec{W,Vec<:AbstractArray{T<:Real,1}},dims::(Int64...,))` at [/Users/rohan/.julia/v0.3/StatsBase/src/sampling.jl:532](#)
(<https://github.com/JuliaStats/StatsBase.jl/tree/23b36af460cfd6c147efef9074a76a4e8cf13dae/src/>)

Column of ones for intercept

```
In [5]: X = [ones(nRows,1) X]
```

```
Out[5]: 10x6 Array{Float64,2}:  
 1.0  0.0  0.0  2.0  1.0  2.0  
 1.0  1.0  1.0  1.0  1.0  0.0  
 1.0  0.0  2.0  1.0  1.0  0.0  
 1.0  1.0  2.0  1.0  0.0  2.0  
 1.0  0.0  1.0  1.0  0.0  2.0  
 1.0  2.0  1.0  0.0  0.0  2.0  
 1.0  2.0  2.0  2.0  0.0  2.0  
 1.0  1.0  2.0  2.0  2.0  1.0  
 1.0  0.0  0.0  2.0  2.0  1.0  
 1.0  2.0  2.0  0.0  2.0  0.0
```

Simulate effects from normal distribution

```
In [6]: nRowsX, nColsX = size(X)
        mean = 0.0
        std  = 0.5
        b = rand(Normal(mean,std),nColsX)
```

```
Out[6]: 6-element Array{Float64,1}:
        -0.34724
         0.0406174
        -0.316707
         0.233593
         0.0933254
         0.277288
```

Simulate phenotypic values

```
In [7]: resStd = 1.0  
y = X*b + rand(Normal(0,resStd),nRowsX)
```

```
Out[7]: 10-element Array{Float64,1}:  
 -0.0880872  
 -1.17895  
 -2.80082  
  2.08141  
  0.371737  
 -0.358808  
  0.0203133  
 -1.26218  
  0.317851  
 -1.2807
```

Function to simulate data

```
In [8]: using Distributions
function simDat(nObs,nLoci,bMean,bStd,resStd)
    X = [ones(nObs,1) sample([0,1,2],(nObs,nLoci))]
    b = rand(Normal(bMean,bStd),size(X,2))
    y = X*b + rand(Normal(0.0, resStd),nObs)
    return (y,X)
end
nObs      = 10
nLoci     = 5
bMean     = 0.0
bStd      = 0.5
resStd    = 1.0
res = simDat(nObs,nLoci,bMean,bStd,resStd)
y = res[1]
X = res[2]
```

```
Out[8]: 10x6 Array{Float64,2}:
 1.0  1.0  0.0  2.0  2.0  1.0
 1.0  2.0  0.0  2.0  0.0  2.0
 1.0  2.0  1.0  0.0  0.0  0.0
 1.0  0.0  1.0  1.0  1.0  1.0
 1.0  2.0  1.0  1.0  2.0  1.0
 1.0  2.0  1.0  1.0  0.0  0.0
 1.0  1.0  1.0  0.0  0.0  0.0
 1.0  1.0  2.0  2.0  0.0  0.0
 1.0  0.0  0.0  2.0  0.0  0.0
 1.0  1.0  2.0  0.0  0.0  2.0
```


XSim: Genome sampler

- Simulate SNPs on chromosomes
- Random mating in finite population to generate LD
- Efficient algorithm for sampling sequence data

Install XSim

```
In [9]: # installing package  
# only needs to be done once  
Pkg.clone("https://github.com/reworkhow/XSim.jl.git")
```

```
INFO: Cloning XSim from https://github.com/reworkhow/XSim.jl.git
```

```
XSim already exists
```

```
while loading In[9], in expression starting on line 3
```

```
in error at error.jl:21  
in clone at pkg/entry.jl:148  
in clone at pkg/entry.jl:175  
in anonymous at pkg/dir.jl:28  
in cd at /Applications/Julia-0.3.7.app/Contents/Resources/julia/lib/julia/sys.dylib  
in __cd#228__ at /Applications/Julia-0.3.7.app/Contents/Resources/julia/lib/julia/sys.dylib  
in clone at pkg.jl:30
```

Initialize sampler

```
In [10]: using XSim
chrLength = 1.0
numChr     = 1
numLoci    = 2000
mutRate    = 0.0
locusInt   = chrLength/numLoci
mapPos     = [0:locusInt:(chrLength-0.0001)]
geneFreq   = fill(0.5,numLoci)
XSim.init(numChr,numLoci,chrLength,geneFreq,mapPos,muRate)
```

Simulate random mating in finite population

```
In [14]: pop      = startPop()  
         nGen     = 10  
         popSize  = 500  
         pop.popSample(nGen, popSize)
```

```
Sampling 500 animals into base population.  
Sampling 500 animals into generation: 1  
Sampling 500 animals into generation: 2  
Sampling 500 animals into generation: 3  
Sampling 500 animals into generation: 4  
Sampling 500 animals into generation: 5  
Sampling 500 animals into generation: 6  
Sampling 500 animals into generation: 7  
Sampling 500 animals into generation: 8  
Sampling 500 animals into generation: 9
```