Genomic Mating V2.0

Deniz Akdemir

2018-07-01

GenomicMating

Genomic selection (GS) is being used increasingly in plant breeding to accelerate genetic gain (Crossa et al. 2010; Roorkiwal et al. 2016; Zhang et al. 2015; Edriss et al. 2017). Genomic selection focuses on best performance of parents before mating, while genomic mating (GM) (Akdemir 2017) includes information on complementation of parents to be mated and thereby is more sustainable in the longer term.

What is this package for?

This package implements the genomic mating (GM) approach. The cross-variance term can be calculated with 3 different methods: Method 1 uses the calculations in ''Efficient Breeding by Genomic Mating'', Method 2 uses the calculations in''Genetic gain increases by applying the usefulness criterion with improved variance prediction in selection of crosses''without the estimation variance terms. Method 2 comes with two types (DH (type=0) or riself (type=1)) and each of these types can be applied for progeny at a specified''generation''. Method 3 is for polyploid organisms, where the marker data is recorded as proportions ofalleles at genomewide loci.

As opposed to the continuous parentage contribution proportions solutions in the GS method, the genomic mating (GM) method gives the list of parent mates of the progeny. Multi-objective optimization problem (assuming maximization is sought for the trait) of the GM problem involves minimization of $-Gain(P)$, $-Usefulness(P)$ and $Inbreeding(P)$ with respect to mating plan P.

The expected gain for a mating plan can be calculated from the mid parent genetic values. There are several alternative measures of inbreeding based on mating plans Leutenegger et al. [2003], Wang [2011]. In Akdemir and Sánchez [2016], we have used a measure derived under the infinitesimal genetic effects assumption proposed by Quaas [1988] and Legarra et al. [2009]. Measures of expected cross-variance usefulness) can be obtained using the results in Akdemir and Sánchez [2016] under the assumption of unlinked markers.

There are three options for the calculation of usefulness. Method 1 uses the calculations in ''Efficient Breeding by Genomic Mating'', Method 2 uses the calculations in''Genetic gain increases by applying the usefulness criterion with improved variance prediction in selection of crosses''without the estimation variance terms. Method 2 comes with two types (DH (type=0) or riself (type=1)) and each of these types can be applied for progeny at a specified''generation''. Method 3 is for polyploid organisms, where the marker data is recorded as proportions ofalleles at genomewide loci.

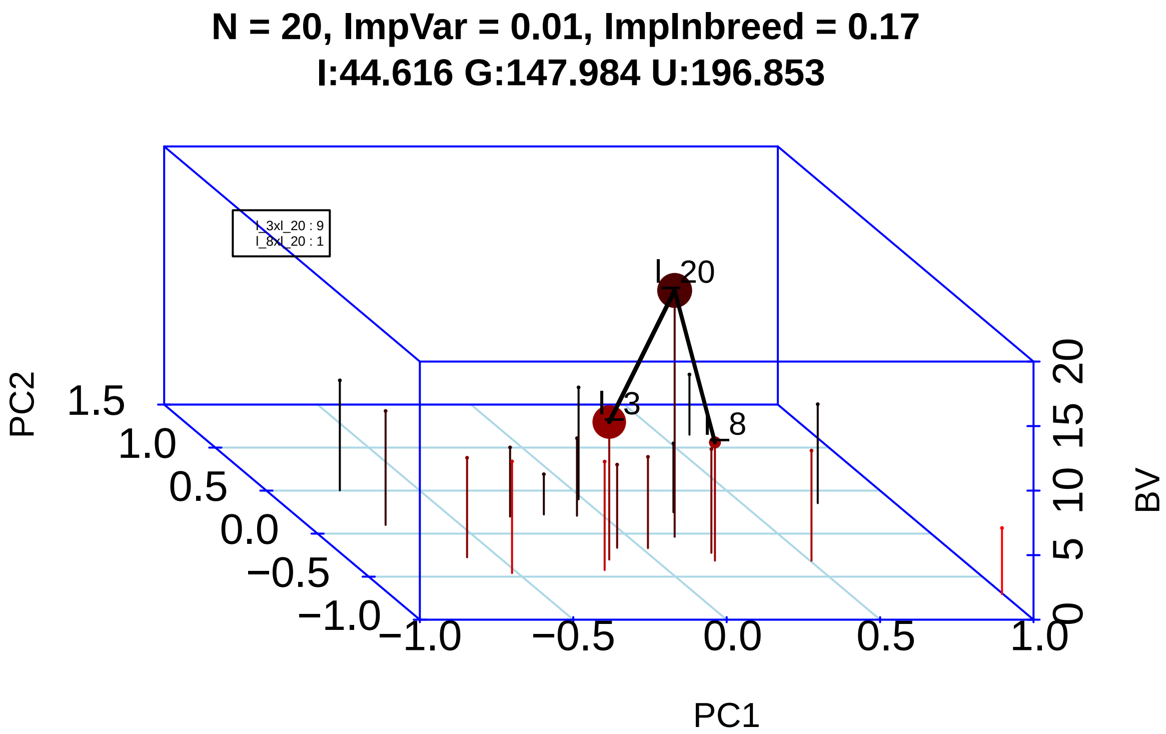
Use of Package

Example 1

One set of markers with Method 1.

```
library(GenomicMating)

set.seed(1234)
N=20
nmarkers=100
```

```r
Markers<-c()
for (i in 1:N){
  Markers<-rbind(Markers,rbinom(nmarkers, 2,.1)-1)
}

markereffects<-rep(0,nmarkers)
markereffects[sample(1:nmarkers,nmarkers/2)]<-rnorm(nmarkers/2)
Markers[1:5,1:5]
#>      [,1] [,2] [,3] [,4] [,5]
#> [1,]   -1   -1   -1   -1    0
#> [2,]   -1   -1   -1   -1   -1
#> [3,]   -1   -1   -1   -1   -1
#> [4,]   -1    0   -1   -1    0
#> [5,]   -1   -1   -1   -1   -1
#library(parallel)
K=Amat.pieces(rbind(Markers), pieces=5)

K[1:5,1:5]
#>             [,1]        [,2]        [,3]        [,4]        [,5]
#> [1,]  0.82526782 -0.10860731  0.05984485 -0.04950129 -0.19726635
#> [2,] -0.10860731  1.49907647 -0.04654599 -0.15589213 -0.06723310
#> [3,]  0.05984485 -0.04654599  0.59475434 -0.04654599 -0.01699298
#> [4,] -0.04950129 -0.15589213 -0.04654599  1.32175840 -0.18544514
#> [5,] -0.19726635 -0.06723310 -0.01699298 -0.18544514  0.90801625
rownames(Markers)<-paste("l", 1:nrow(Markers),sep="_")

rownames(K)<-colnames(K)<-c(rownames(Markers))

which.max(Markers%*%markereffects)
#> [1] 20
markermap=as.matrix(data.frame(chr=rep(1,nmarkers),
                               pos=seq(0,1,length=nmarkers)))

colnames(Markers)<-1:nmarkers
gasols<-getGaSolutions(Markers=Markers,Markers2=NULL,
                       K=K, markereffects=markereffects,
                       markermap=markermap,
                       nmates=10,
                       minparents=3,
                       impinbreedstepsize=.02, impvar=.01,
                       impforinbreed=.01,
                       npopGA=100, nitGA=300, miniters=300,
                       minitbefstop=50,plotiters=F,
                       mc.cores=1,nelite=20, mutprob=0.8,
                       noself=T,
                       method=1, type=0L, generation=0L)
```




```
gasols
#> $Mates
#>       [,1]  [,2]
#>  [1,] "l_3" "l_20"
#>  [2,] "l_3" "l_20"
#>  [3,] "l_3" "l_20"
#>  [4,] "l_3" "l_20"
#>  [5,] "l_3" "l_20"
#>  [6,] "l_3" "l_20"
#>  [7,] "l_3" "l_20"
#>  [8,] "l_3" "l_20"
#>  [9,] "l_3" "l_20"
#> [10,] "l_8" "l_20"
#>
#> $StatVal_Iter
#>   StatVal_Iter1   StatVal_Iter2   StatVal_Iter3   StatVal_Iter4
#>       -91.20075      -100.36515      -103.81517      -114.65212
#>   StatVal_Iter5   StatVal_Iter6   StatVal_Iter7   StatVal_Iter8
#>      -114.65212      -115.40536      -115.40536      -115.65873
#>   StatVal_Iter9  StatVal_Iter10  StatVal_Iter11  StatVal_Iter12
#>      -115.65873      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter13  StatVal_Iter14  StatVal_Iter15  StatVal_Iter16
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter17  StatVal_Iter18  StatVal_Iter19  StatVal_Iter20
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter21  StatVal_Iter22  StatVal_Iter23  StatVal_Iter24
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter25  StatVal_Iter26  StatVal_Iter27  StatVal_Iter28
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter29  StatVal_Iter30  StatVal_Iter31  StatVal_Iter32
```

```
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter33  StatVal_Iter34  StatVal_Iter35  StatVal_Iter36
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter37  StatVal_Iter38  StatVal_Iter39  StatVal_Iter40
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter41  StatVal_Iter42  StatVal_Iter43  StatVal_Iter44
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter45  StatVal_Iter46  StatVal_Iter47  StatVal_Iter48
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter49  StatVal_Iter50  StatVal_Iter51  StatVal_Iter52
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter53  StatVal_Iter54  StatVal_Iter55  StatVal_Iter56
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter57  StatVal_Iter58  StatVal_Iter59  StatVal_Iter60
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter61  StatVal_Iter62  StatVal_Iter63  StatVal_Iter64
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter65  StatVal_Iter66  StatVal_Iter67  StatVal_Iter68
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter69  StatVal_Iter70  StatVal_Iter71  StatVal_Iter72
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter73  StatVal_Iter74  StatVal_Iter75  StatVal_Iter76
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter77  StatVal_Iter78  StatVal_Iter79  StatVal_Iter80
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter81  StatVal_Iter82  StatVal_Iter83  StatVal_Iter84
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter85  StatVal_Iter86  StatVal_Iter87  StatVal_Iter88
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter89  StatVal_Iter90  StatVal_Iter91  StatVal_Iter92
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter93  StatVal_Iter94  StatVal_Iter95  StatVal_Iter96
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>  StatVal_Iter97  StatVal_Iter98  StatVal_Iter99 StatVal_Iter100
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter101 StatVal_Iter102 StatVal_Iter103 StatVal_Iter104
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter105 StatVal_Iter106 StatVal_Iter107 StatVal_Iter108
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter109 StatVal_Iter110 StatVal_Iter111 StatVal_Iter112
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter113 StatVal_Iter114 StatVal_Iter115 StatVal_Iter116
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter117 StatVal_Iter118 StatVal_Iter119 StatVal_Iter120
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter121 StatVal_Iter122 StatVal_Iter123 StatVal_Iter124
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter125 StatVal_Iter126 StatVal_Iter127 StatVal_Iter128
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter129 StatVal_Iter130 StatVal_Iter131 StatVal_Iter132
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter133 StatVal_Iter134 StatVal_Iter135 StatVal_Iter136
#>      -115.73098      -115.73098      -115.73098      -115.73098
```

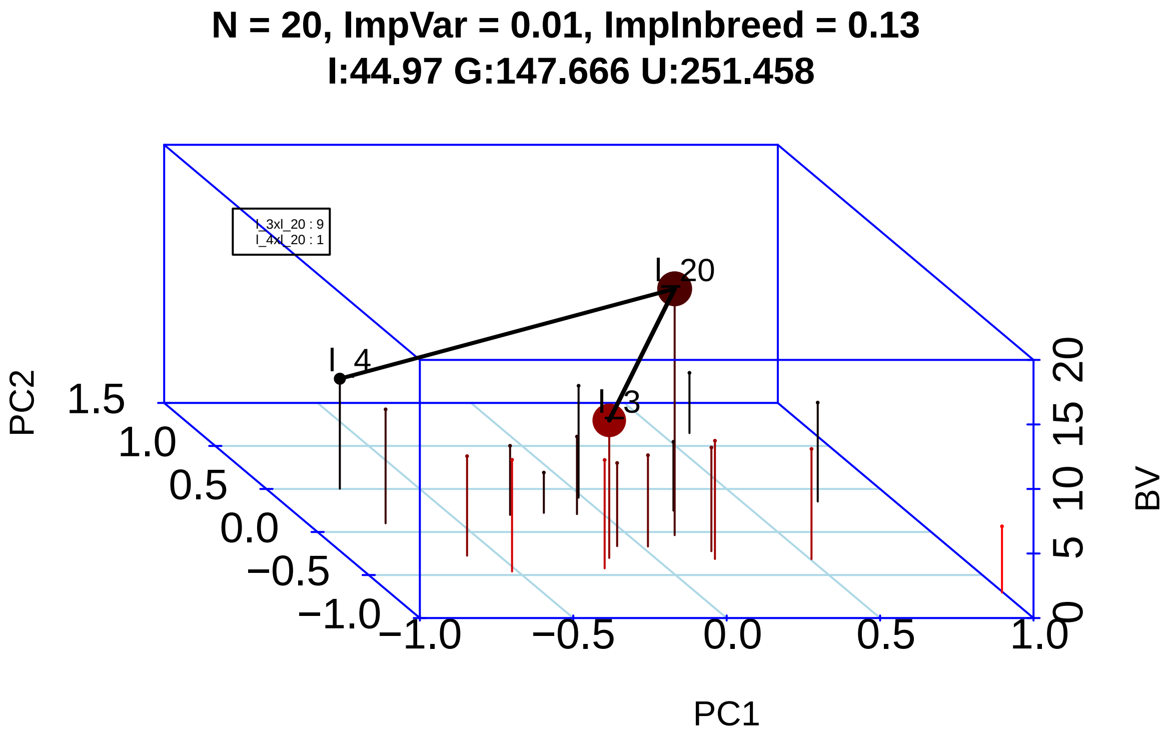
```
#> StatVal_Iter137 StatVal_Iter138 StatVal_Iter139 StatVal_Iter140
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter141 StatVal_Iter142 StatVal_Iter143 StatVal_Iter144
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter145 StatVal_Iter146 StatVal_Iter147 StatVal_Iter148
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter149 StatVal_Iter150 StatVal_Iter151 StatVal_Iter152
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter153 StatVal_Iter154 StatVal_Iter155 StatVal_Iter156
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter157 StatVal_Iter158 StatVal_Iter159 StatVal_Iter160
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter161 StatVal_Iter162 StatVal_Iter163 StatVal_Iter164
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter165 StatVal_Iter166 StatVal_Iter167 StatVal_Iter168
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter169 StatVal_Iter170 StatVal_Iter171 StatVal_Iter172
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter173 StatVal_Iter174 StatVal_Iter175 StatVal_Iter176
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter177 StatVal_Iter178 StatVal_Iter179 StatVal_Iter180
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter181 StatVal_Iter182 StatVal_Iter183 StatVal_Iter184
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter185 StatVal_Iter186 StatVal_Iter187 StatVal_Iter188
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter189 StatVal_Iter190 StatVal_Iter191 StatVal_Iter192
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter193 StatVal_Iter194 StatVal_Iter195 StatVal_Iter196
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter197 StatVal_Iter198 StatVal_Iter199 StatVal_Iter200
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter201 StatVal_Iter202 StatVal_Iter203 StatVal_Iter204
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter205 StatVal_Iter206 StatVal_Iter207 StatVal_Iter208
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter209 StatVal_Iter210 StatVal_Iter211 StatVal_Iter212
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter213 StatVal_Iter214 StatVal_Iter215 StatVal_Iter216
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter217 StatVal_Iter218 StatVal_Iter219 StatVal_Iter220
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter221 StatVal_Iter222 StatVal_Iter223 StatVal_Iter224
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter225 StatVal_Iter226 StatVal_Iter227 StatVal_Iter228
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter229 StatVal_Iter230 StatVal_Iter231 StatVal_Iter232
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter233 StatVal_Iter234 StatVal_Iter235 StatVal_Iter236
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter237 StatVal_Iter238 StatVal_Iter239 StatVal_Iter240
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter241 StatVal_Iter242 StatVal_Iter243 StatVal_Iter244
```

```
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter245 StatVal_Iter246 StatVal_Iter247 StatVal_Iter248
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter249 StatVal_Iter250 StatVal_Iter251 StatVal_Iter252
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter253 StatVal_Iter254 StatVal_Iter255 StatVal_Iter256
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter257 StatVal_Iter258 StatVal_Iter259 StatVal_Iter260
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter261 StatVal_Iter262 StatVal_Iter263 StatVal_Iter264
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter265 StatVal_Iter266 StatVal_Iter267 StatVal_Iter268
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter269 StatVal_Iter270 StatVal_Iter271 StatVal_Iter272
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter273 StatVal_Iter274 StatVal_Iter275 StatVal_Iter276
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter277 StatVal_Iter278 StatVal_Iter279 StatVal_Iter280
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter281 StatVal_Iter282 StatVal_Iter283 StatVal_Iter284
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter285 StatVal_Iter286 StatVal_Iter287 StatVal_Iter288
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter289 StatVal_Iter290 StatVal_Iter291 StatVal_Iter292
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter293 StatVal_Iter294 StatVal_Iter295 StatVal_Iter296
#>      -115.73098      -115.73098      -115.73098      -115.73098
#> StatVal_Iter297 StatVal_Iter298 StatVal_Iter299 StatVal_Iter300
#>      -115.73098      -115.73098      -115.73098      -115.73098
#>
#> $Stats
#> $Stats$I
#> [1] 44.61581
#>
#> $Stats$G
#> [1] 147.9843
#>
#> $Stats$U
#> [1] 196.8535
```

Method 2

```
gasols<-getGaSolutions(Markers=Markers,Markers2=NULL,
                       K=K, markereffects=markereffects,
                       markermap=markermap,
                       nmates=10,
                       minparents=3,
                       impinbreedstepsize=.02, impvar=.01,
                       impforinbreed=.01,
                       npopGA=100, nitGA=300, miniters=300,
                       minitbefstop=50,plotiters=F,
                       mc.cores=1,nelite=20, mutprob=0.8,
```

```
                       noself=T,
                       method=2, type=0L, generation=0L)
```




```
gasols
#> $Mates
#>       [,1]  [,2]
#>  [1,] "l_3" "l_20"
#>  [2,] "l_3" "l_20"
#>  [3,] "l_3" "l_20"
#>  [4,] "l_3" "l_20"
#>  [5,] "l_3" "l_20"
#>  [6,] "l_3" "l_20"
#>  [7,] "l_3" "l_20"
#>  [8,] "l_3" "l_20"
#>  [9,] "l_3" "l_20"
#> [10,] "l_4" "l_20"
#>
#> $StatVal_Iter
#>   StatVal_Iter1   StatVal_Iter2   StatVal_Iter3   StatVal_Iter4
#>       -92.78621       -96.96453      -103.91771      -115.48006
#>   StatVal_Iter5   StatVal_Iter6   StatVal_Iter7   StatVal_Iter8
#>      -115.81627      -119.35778      -121.16153      -121.65117
#>   StatVal_Iter9  StatVal_Iter10  StatVal_Iter11  StatVal_Iter12
#>      -121.65117      -123.08326      -123.40751      -123.40751
#>  StatVal_Iter13  StatVal_Iter14  StatVal_Iter15  StatVal_Iter16
#>      -123.40751      -123.40751      -123.62034      -123.66084
#>  StatVal_Iter17  StatVal_Iter18  StatVal_Iter19  StatVal_Iter20
#>      -123.66084      -123.66084      -123.66084      -123.66084
#>  StatVal_Iter21  StatVal_Iter22  StatVal_Iter23  StatVal_Iter24
#>      -123.66084      -123.66084      -123.66084      -123.66084
```

```
#>  StatVal_Iter25  StatVal_Iter26  StatVal_Iter27  StatVal_Iter28
#>      -123.66084      -123.66084      -123.66084      -123.66084
#>  StatVal_Iter29  StatVal_Iter30  StatVal_Iter31  StatVal_Iter32
#>      -123.66084      -123.66084      -123.66084      -123.66084
#>  StatVal_Iter33  StatVal_Iter34  StatVal_Iter35  StatVal_Iter36
#>      -123.66084      -123.66084      -123.66084      -123.66084
#>  StatVal_Iter37  StatVal_Iter38  StatVal_Iter39  StatVal_Iter40
#>      -123.66084      -123.66084      -123.66084      -123.66084
#>  StatVal_Iter41  StatVal_Iter42  StatVal_Iter43  StatVal_Iter44
#>      -123.66084      -123.66084      -123.66084      -123.66084
#>  StatVal_Iter45  StatVal_Iter46  StatVal_Iter47  StatVal_Iter48
#>      -123.66084      -123.66084      -123.66084      -123.66084
#>  StatVal_Iter49  StatVal_Iter50  StatVal_Iter51  StatVal_Iter52
#>      -123.66084      -123.66084      -123.66084      -123.66084
#>  StatVal_Iter53  StatVal_Iter54  StatVal_Iter55  StatVal_Iter56
#>      -123.66084      -123.66084      -123.66084      -123.66084
#>  StatVal_Iter57  StatVal_Iter58  StatVal_Iter59  StatVal_Iter60
#>      -123.66084      -123.66084      -123.66084      -123.66084
#>  StatVal_Iter61  StatVal_Iter62  StatVal_Iter63  StatVal_Iter64
#>      -123.66084      -123.66084      -123.66084      -123.66084
#>  StatVal_Iter65  StatVal_Iter66  StatVal_Iter67  StatVal_Iter68
#>      -123.66084      -123.66084      -123.66084      -123.66084
#>  StatVal_Iter69  StatVal_Iter70  StatVal_Iter71  StatVal_Iter72
#>      -123.66084      -123.66084      -123.66084      -123.66084
#>  StatVal_Iter73  StatVal_Iter74  StatVal_Iter75  StatVal_Iter76
#>      -123.66084      -123.66084      -123.66084      -123.66084
#>  StatVal_Iter77  StatVal_Iter78  StatVal_Iter79  StatVal_Iter80
#>      -123.66084      -123.66084      -123.66084      -123.66084
#>  StatVal_Iter81  StatVal_Iter82  StatVal_Iter83  StatVal_Iter84
#>      -123.66084      -123.66084      -123.66084      -123.66084
#>  StatVal_Iter85  StatVal_Iter86  StatVal_Iter87  StatVal_Iter88
#>      -123.66084      -123.66084      -123.66084      -123.66084
#>  StatVal_Iter89  StatVal_Iter90  StatVal_Iter91  StatVal_Iter92
#>      -123.66084      -123.66084      -123.66084      -123.66084
#>  StatVal_Iter93  StatVal_Iter94  StatVal_Iter95  StatVal_Iter96
#>      -123.66084      -123.66084      -123.66084      -123.66084
#>  StatVal_Iter97  StatVal_Iter98  StatVal_Iter99 StatVal_Iter100
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter101 StatVal_Iter102 StatVal_Iter103 StatVal_Iter104
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter105 StatVal_Iter106 StatVal_Iter107 StatVal_Iter108
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter109 StatVal_Iter110 StatVal_Iter111 StatVal_Iter112
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter113 StatVal_Iter114 StatVal_Iter115 StatVal_Iter116
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter117 StatVal_Iter118 StatVal_Iter119 StatVal_Iter120
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter121 StatVal_Iter122 StatVal_Iter123 StatVal_Iter124
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter125 StatVal_Iter126 StatVal_Iter127 StatVal_Iter128
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter129 StatVal_Iter130 StatVal_Iter131 StatVal_Iter132
```

```
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter133 StatVal_Iter134 StatVal_Iter135 StatVal_Iter136
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter137 StatVal_Iter138 StatVal_Iter139 StatVal_Iter140
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter141 StatVal_Iter142 StatVal_Iter143 StatVal_Iter144
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter145 StatVal_Iter146 StatVal_Iter147 StatVal_Iter148
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter149 StatVal_Iter150 StatVal_Iter151 StatVal_Iter152
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter153 StatVal_Iter154 StatVal_Iter155 StatVal_Iter156
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter157 StatVal_Iter158 StatVal_Iter159 StatVal_Iter160
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter161 StatVal_Iter162 StatVal_Iter163 StatVal_Iter164
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter165 StatVal_Iter166 StatVal_Iter167 StatVal_Iter168
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter169 StatVal_Iter170 StatVal_Iter171 StatVal_Iter172
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter173 StatVal_Iter174 StatVal_Iter175 StatVal_Iter176
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter177 StatVal_Iter178 StatVal_Iter179 StatVal_Iter180
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter181 StatVal_Iter182 StatVal_Iter183 StatVal_Iter184
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter185 StatVal_Iter186 StatVal_Iter187 StatVal_Iter188
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter189 StatVal_Iter190 StatVal_Iter191 StatVal_Iter192
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter193 StatVal_Iter194 StatVal_Iter195 StatVal_Iter196
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter197 StatVal_Iter198 StatVal_Iter199 StatVal_Iter200
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter201 StatVal_Iter202 StatVal_Iter203 StatVal_Iter204
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter205 StatVal_Iter206 StatVal_Iter207 StatVal_Iter208
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter209 StatVal_Iter210 StatVal_Iter211 StatVal_Iter212
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter213 StatVal_Iter214 StatVal_Iter215 StatVal_Iter216
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter217 StatVal_Iter218 StatVal_Iter219 StatVal_Iter220
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter221 StatVal_Iter222 StatVal_Iter223 StatVal_Iter224
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter225 StatVal_Iter226 StatVal_Iter227 StatVal_Iter228
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter229 StatVal_Iter230 StatVal_Iter231 StatVal_Iter232
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter233 StatVal_Iter234 StatVal_Iter235 StatVal_Iter236
#>      -123.66084      -123.66084      -123.66084      -123.66084
```

```
#> StatVal_Iter237 StatVal_Iter238 StatVal_Iter239 StatVal_Iter240
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter241 StatVal_Iter242 StatVal_Iter243 StatVal_Iter244
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter245 StatVal_Iter246 StatVal_Iter247 StatVal_Iter248
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter249 StatVal_Iter250 StatVal_Iter251 StatVal_Iter252
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter253 StatVal_Iter254 StatVal_Iter255 StatVal_Iter256
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter257 StatVal_Iter258 StatVal_Iter259 StatVal_Iter260
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter261 StatVal_Iter262 StatVal_Iter263 StatVal_Iter264
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter265 StatVal_Iter266 StatVal_Iter267 StatVal_Iter268
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter269 StatVal_Iter270 StatVal_Iter271 StatVal_Iter272
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter273 StatVal_Iter274 StatVal_Iter275 StatVal_Iter276
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter277 StatVal_Iter278 StatVal_Iter279 StatVal_Iter280
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter281 StatVal_Iter282 StatVal_Iter283 StatVal_Iter284
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter285 StatVal_Iter286 StatVal_Iter287 StatVal_Iter288
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter289 StatVal_Iter290 StatVal_Iter291 StatVal_Iter292
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter293 StatVal_Iter294 StatVal_Iter295 StatVal_Iter296
#>      -123.66084      -123.66084      -123.66084      -123.66084
#> StatVal_Iter297 StatVal_Iter298 StatVal_Iter299 StatVal_Iter300
#>      -123.66084      -123.66084      -123.66084      -123.66084
#>
#> $Stats
#> $Stats$I
#> [1] 44.97045
#>
#> $Stats$G
#> [1] 147.6656
#>
#> $Stats$U
#> [1] 251.4577
```

Optimal Parental Contributions

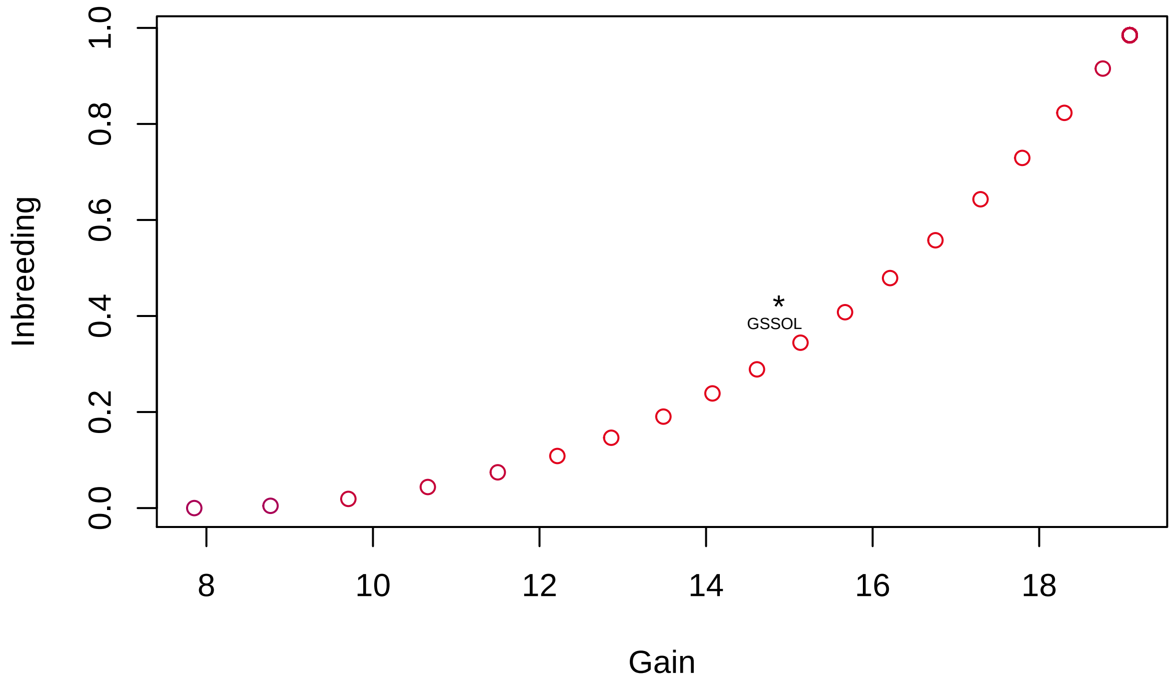
getOptParentalProportions function can be used to calculate optimal parental contributions given a relationship matrix and GEBVs.

```
oprop<-getOptParentalProportions(Amat=K,
      gebvs=Markers%*%markereffects, lambda=.8, ul=1)
```

Compare GS solution to the frontier curve

plotOPFrontier function can be used to the frontier curve for optimal parental contributions given a relationship matrix and GEBVs. We compare GS with truncation selection and equal weights for selected lines to this curve. GS with truncation selection and equal weights is not necessarily optimal.

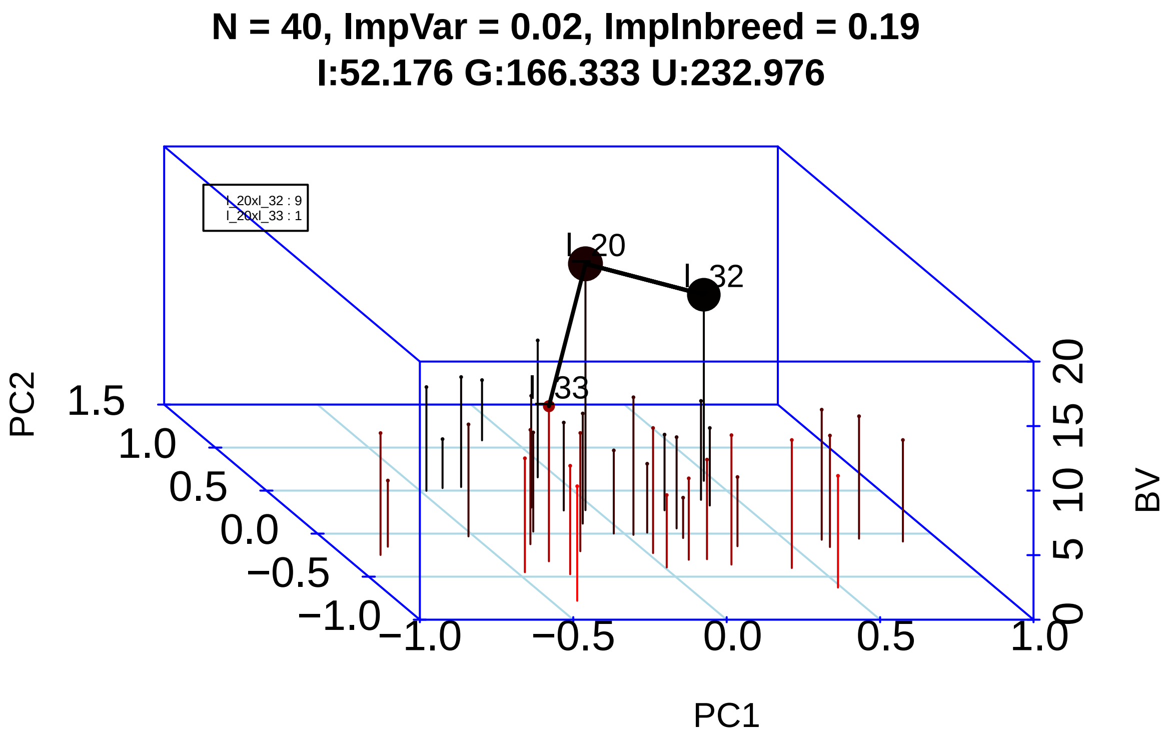
```
pout<-plotOPFrontier(Amat=K,
      gebvs=Markers%*%markereffects, ul=1, identify=FALSE)
round(oprop,3)
#>        l_1        l_2        l_3        l_4        l_5        l_6
#>      0.000      0.000      0.000      0.000      0.000      0.000
#>        l_7        l_8        l_9       l_10       l_11       l_12
#>      0.000      0.000      0.000      0.000      0.000      0.000
#>       l_13       l_14       l_15       l_16       l_17       l_18
#>      0.000      0.000      0.000      0.000      0.000      0.000
#>       l_19       l_20     lambda       Gain Inbreeding  G/I ratio
#>      0.000      1.000      0.800     19.088      0.985     19.381
uhat<-Markers%*%markereffects

gsselected<-which(uhat>quantile(uhat,.9))
gsgain<-mean(uhat[gsselected])

onesvec<-matrix(1,nrow=length(uhat),ncol=1)

onesvec[-gsselected]<-0
onesvec<-onesvec/sum(onesvec)
gsinbreed<-t(onesvec)%*%K%*%onesvec
gsgain
#> [1] 14.87335
gsinbreed
#>           [,1]
#> [1,] 0.4351681
round(oprop,3)
#>        l_1        l_2        l_3        l_4        l_5        l_6
#>      0.000      0.000      0.000      0.000      0.000      0.000
#>        l_7        l_8        l_9       l_10       l_11       l_12
#>      0.000      0.000      0.000      0.000      0.000      0.000
#>       l_13       l_14       l_15       l_16       l_17       l_18
#>      0.000      0.000      0.000      0.000      0.000      0.000
#>       l_19       l_20     lambda       Gain Inbreeding  G/I ratio
#>      0.000      1.000      0.800     19.088      0.985     19.381
t(oprop[1:(length(oprop)-4)])%*%K%*%oprop[1:(length(oprop)-4)]
#>           [,1]
#> [1,] 0.9848541
points(gsgain,gsinbreed, pch="*")
text(x=gsgain-.05,y=gsinbreed-.05, "GSSOL", cex=.5)
```



Other methods

Two sets of markers

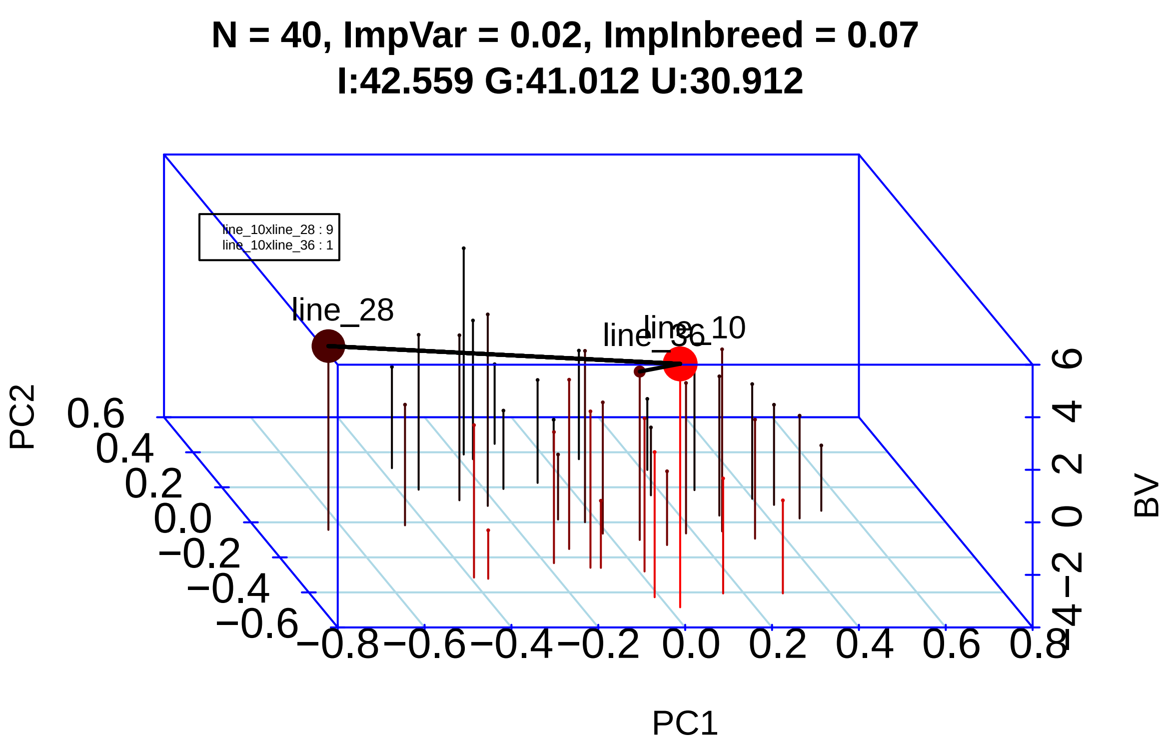
```
Markers2<-c()
for (i in 1:N){
  Markers2<-rbind(Markers2,rbinom(nmarkers, 2,.1)-1)
}


K2=Amat.pieces(rbind(Markers,Markers2), pieces=5)
rownames(Markers2)<-paste("l", (nrow(Markers)+1):(nrow(Markers)+
                                                  nrow(Markers2)),sep="_")
rownames(K2)<-colnames(K2)<-c(rownames(Markers),rownames(Markers2))


gasols1<-getGaSolutions(Markers=Markers,Markers2=Markers2, K=K2, markereffects,markermap=markermap,nmate
                       minparents=3,
                       impinbreedstepsize=.02, impvar=.02,
                       impforinbreed=.07,
                       npopGA=100, nitGA=300, miniters=300,minitbefstop=50,
                       plotiters=F,
                       mc.cores=2,nelite=20, mutprob=0.8, noself=F,
                       method=1,
                       type=0L, generation=0L)
```



Method 3 for polyploids, markers coded as probabilities.

```
N=20
nmarkers=100
Markers<-c()
for (i in 1:N){
  Markers<-rbind(Markers,runif(nmarkers))
}


Markers2<-c()
for (i in 1:N){
  Markers2<-rbind(Markers2,runif(nmarkers))
}

markereffects<-rep(0,nmarkers)
markereffects[sample(1:nmarkers,nmarkers/2)]<-rnorm(nmarkers/2)
Markers[1:5,1:5]
#>           [,1]      [,2]       [,3]      [,4]      [,5]
#> [1,] 0.2897657 0.2268388 0.56845472 0.8622676 0.5901730
#> [2,] 0.9023577 0.6428505 0.41603166 0.3976138 0.2422303
#> [3,] 0.0587096 0.8385445 0.74680579 0.9430178 0.8885681
#> [4,] 0.6067942 0.3662235 0.61196930 0.6403971 0.8314868
#> [5,] 0.7017032 0.1255202 0.03437093 0.4150999 0.8826229
#library(parallel)
K=Amat.pieces(rbind(Markers)*2-1, pieces=5)

K2=Amat.pieces(rbind(Markers,Markers2)*2-1, pieces=5)
K[1:5,1:5]
```

```r
#>              [,1]         [,2]        [,3]        [,4]        [,5]
#> [1,]  0.603988304 -0.005422122 -0.06792327  0.03579216 -0.03365847
#> [2,] -0.005422122  0.646140752 -0.17141561  0.04374552 -0.04893127
#> [3,] -0.067923267 -0.171415606  0.70028407 -0.06765105 -0.13966793
#> [4,]  0.035792159  0.043745517 -0.06765105  0.62840233 -0.09318536
#> [5,] -0.033658470 -0.048931270 -0.13966793 -0.09318536  0.70773486
rownames(Markers)<-paste("line", 1:nrow(Markers),sep="_")
rownames(Markers2)<-paste("line", (nrow(Markers)+1):(nrow(Markers)+nrow(Markers2)),sep="_")
rownames(K2)<-colnames(K2)<-c(rownames(Markers),rownames(Markers2))
rownames(K)<-colnames(K)<-c(rownames(Markers))

which.max(Markers%*%markereffects)
#> [1] 10
markermap=as.matrix(data.frame(chr=rep(1,nmarkers),
                               pos=seq(0,1,length=nmarkers)))

colnames(Markers)<-1:nmarkers

gasols2<-getGaSolutions(Markers=Markers,Markers2=Markers2,
                        K=K2, markereffects,markermap=markermap,
                        nmates=10,
                       minparents=1,
                       impinbreedstepsize=.02, impvar=.02,
                       impforinbreed=.07,
                       npopGA=100, nitGA=300, miniters=300,
                       minitbefstop=50,plotiters=F,
                       mc.cores=1,nelite=20, mutprob=0.8,
                       noself=F, method=3,
                       type=0L, generation=0L)
```



```
gasols2
#> $Mates
#>       [,1]      [,2]
#>  [1,] "line_10" "line_28"
#>  [2,] "line_10" "line_28"
#>  [3,] "line_10" "line_28"
#>  [4,] "line_10" "line_28"
#>  [5,] "line_10" "line_28"
#>  [6,] "line_10" "line_28"
#>  [7,] "line_10" "line_28"
#>  [8,] "line_10" "line_28"
#>  [9,] "line_10" "line_28"
#> [10,] "line_10" "line_36"
#>
#> $StatVal_Iter
#>   StatVal_Iter1   StatVal_Iter2   StatVal_Iter3   StatVal_Iter4
#>       -20.60283       -23.67248       -24.65878       -28.25755
#>   StatVal_Iter5   StatVal_Iter6   StatVal_Iter7   StatVal_Iter8
#>       -28.36247       -30.37453       -31.73627       -33.47872
#>   StatVal_Iter9  StatVal_Iter10  StatVal_Iter11  StatVal_Iter12
#>       -33.47872       -34.57977       -34.87022       -34.93006
#>  StatVal_Iter13  StatVal_Iter14  StatVal_Iter15  StatVal_Iter16
#>       -34.95972       -34.95972       -34.95972       -34.95972
#>  StatVal_Iter17  StatVal_Iter18  StatVal_Iter19  StatVal_Iter20
#>       -34.95972       -34.95972       -34.95972       -34.95972
#>  StatVal_Iter21  StatVal_Iter22  StatVal_Iter23  StatVal_Iter24
#>       -34.95972       -34.95972       -34.95972       -34.95972
#>  StatVal_Iter25  StatVal_Iter26  StatVal_Iter27  StatVal_Iter28
#>       -34.95972       -34.95972       -34.95972       -34.95972
#>  StatVal_Iter29  StatVal_Iter30  StatVal_Iter31  StatVal_Iter32
#>       -34.95972       -34.95972       -34.95972       -34.95972
#>  StatVal_Iter33  StatVal_Iter34  StatVal_Iter35  StatVal_Iter36
#>       -34.95972       -34.95972       -34.95972       -34.95972
#>  StatVal_Iter37  StatVal_Iter38  StatVal_Iter39  StatVal_Iter40
#>       -34.95972       -34.95972       -34.95972       -34.95972
#>  StatVal_Iter41  StatVal_Iter42  StatVal_Iter43  StatVal_Iter44
#>       -34.95972       -34.95972       -34.95972       -34.95972
#>  StatVal_Iter45  StatVal_Iter46  StatVal_Iter47  StatVal_Iter48
#>       -34.95972       -34.95972       -34.95972       -34.95972
#>  StatVal_Iter49  StatVal_Iter50  StatVal_Iter51  StatVal_Iter52
#>       -34.95972       -34.95972       -34.95972       -34.95972
#>  StatVal_Iter53  StatVal_Iter54  StatVal_Iter55  StatVal_Iter56
#>       -34.95972       -34.95972       -34.95972       -34.95972
#>  StatVal_Iter57  StatVal_Iter58  StatVal_Iter59  StatVal_Iter60
#>       -34.95972       -34.95972       -34.95972       -34.95972
#>  StatVal_Iter61  StatVal_Iter62  StatVal_Iter63  StatVal_Iter64
#>       -34.95972       -34.95972       -34.95972       -34.95972
#>  StatVal_Iter65  StatVal_Iter66  StatVal_Iter67  StatVal_Iter68
#>       -34.95972       -34.95972       -34.95972       -34.95972
#>  StatVal_Iter69  StatVal_Iter70  StatVal_Iter71  StatVal_Iter72
#>       -34.95972       -34.95972       -34.95972       -34.95972
#>  StatVal_Iter73  StatVal_Iter74  StatVal_Iter75  StatVal_Iter76
```

```
#>        -34.95972       -34.95972       -34.95972       -34.95972
#>  StatVal_Iter77  StatVal_Iter78  StatVal_Iter79  StatVal_Iter80
#>        -34.95972       -34.95972       -34.95972       -34.95972
#>  StatVal_Iter81  StatVal_Iter82  StatVal_Iter83  StatVal_Iter84
#>        -34.95972       -34.95972       -34.95972       -34.95972
#>  StatVal_Iter85  StatVal_Iter86  StatVal_Iter87  StatVal_Iter88
#>        -34.95972       -34.95972       -34.95972       -34.95972
#>  StatVal_Iter89  StatVal_Iter90  StatVal_Iter91  StatVal_Iter92
#>        -34.95972       -34.95972       -34.95972       -34.95972
#>  StatVal_Iter93  StatVal_Iter94  StatVal_Iter95  StatVal_Iter96
#>        -34.95972       -34.95972       -34.95972       -34.95972
#>  StatVal_Iter97  StatVal_Iter98  StatVal_Iter99 StatVal_Iter100
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter101 StatVal_Iter102 StatVal_Iter103 StatVal_Iter104
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter105 StatVal_Iter106 StatVal_Iter107 StatVal_Iter108
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter109 StatVal_Iter110 StatVal_Iter111 StatVal_Iter112
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter113 StatVal_Iter114 StatVal_Iter115 StatVal_Iter116
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter117 StatVal_Iter118 StatVal_Iter119 StatVal_Iter120
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter121 StatVal_Iter122 StatVal_Iter123 StatVal_Iter124
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter125 StatVal_Iter126 StatVal_Iter127 StatVal_Iter128
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter129 StatVal_Iter130 StatVal_Iter131 StatVal_Iter132
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter133 StatVal_Iter134 StatVal_Iter135 StatVal_Iter136
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter137 StatVal_Iter138 StatVal_Iter139 StatVal_Iter140
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter141 StatVal_Iter142 StatVal_Iter143 StatVal_Iter144
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter145 StatVal_Iter146 StatVal_Iter147 StatVal_Iter148
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter149 StatVal_Iter150 StatVal_Iter151 StatVal_Iter152
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter153 StatVal_Iter154 StatVal_Iter155 StatVal_Iter156
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter157 StatVal_Iter158 StatVal_Iter159 StatVal_Iter160
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter161 StatVal_Iter162 StatVal_Iter163 StatVal_Iter164
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter165 StatVal_Iter166 StatVal_Iter167 StatVal_Iter168
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter169 StatVal_Iter170 StatVal_Iter171 StatVal_Iter172
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter173 StatVal_Iter174 StatVal_Iter175 StatVal_Iter176
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter177 StatVal_Iter178 StatVal_Iter179 StatVal_Iter180
#>        -34.95972       -34.95972       -34.95972       -34.95972
```

```
#> StatVal_Iter181 StatVal_Iter182 StatVal_Iter183 StatVal_Iter184
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter185 StatVal_Iter186 StatVal_Iter187 StatVal_Iter188
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter189 StatVal_Iter190 StatVal_Iter191 StatVal_Iter192
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter193 StatVal_Iter194 StatVal_Iter195 StatVal_Iter196
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter197 StatVal_Iter198 StatVal_Iter199 StatVal_Iter200
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter201 StatVal_Iter202 StatVal_Iter203 StatVal_Iter204
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter205 StatVal_Iter206 StatVal_Iter207 StatVal_Iter208
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter209 StatVal_Iter210 StatVal_Iter211 StatVal_Iter212
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter213 StatVal_Iter214 StatVal_Iter215 StatVal_Iter216
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter217 StatVal_Iter218 StatVal_Iter219 StatVal_Iter220
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter221 StatVal_Iter222 StatVal_Iter223 StatVal_Iter224
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter225 StatVal_Iter226 StatVal_Iter227 StatVal_Iter228
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter229 StatVal_Iter230 StatVal_Iter231 StatVal_Iter232
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter233 StatVal_Iter234 StatVal_Iter235 StatVal_Iter236
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter237 StatVal_Iter238 StatVal_Iter239 StatVal_Iter240
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter241 StatVal_Iter242 StatVal_Iter243 StatVal_Iter244
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter245 StatVal_Iter246 StatVal_Iter247 StatVal_Iter248
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter249 StatVal_Iter250 StatVal_Iter251 StatVal_Iter252
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter253 StatVal_Iter254 StatVal_Iter255 StatVal_Iter256
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter257 StatVal_Iter258 StatVal_Iter259 StatVal_Iter260
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter261 StatVal_Iter262 StatVal_Iter263 StatVal_Iter264
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter265 StatVal_Iter266 StatVal_Iter267 StatVal_Iter268
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter269 StatVal_Iter270 StatVal_Iter271 StatVal_Iter272
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter273 StatVal_Iter274 StatVal_Iter275 StatVal_Iter276
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter277 StatVal_Iter278 StatVal_Iter279 StatVal_Iter280
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter281 StatVal_Iter282 StatVal_Iter283 StatVal_Iter284
#>       -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter285 StatVal_Iter286 StatVal_Iter287 StatVal_Iter288
```

```
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter289 StatVal_Iter290 StatVal_Iter291 StatVal_Iter292
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter293 StatVal_Iter294 StatVal_Iter295 StatVal_Iter296
#>        -34.95972       -34.95972       -34.95972       -34.95972
#> StatVal_Iter297 StatVal_Iter298 StatVal_Iter299 StatVal_Iter300
#>        -34.95972       -34.95972       -34.95972       -34.95972
#>
#> $Stats
#> $Stats$I
#> [1] 42.5594
#>
#> $Stats$G
#> [1] 41.01169
#>
#> $Stats$U
#> [1] 30.91179
```

Multitrait GM

Most breeding programs are concerned with simultaneous improvement of several traits. For example, although yield is usually the primary trait of interest for most crops; maturity, standability, grain quality, stalk quality, abiotic and biotic stress tolerance, etc. are also economically important traits. Simultaneous selection for several traits is necessary if recurrent selection methods are used. Selection that emphasizes only one trait can be detrimental to the overall agronomic performance of the germplasm [Hallauer and Carena Filho].

Extention to multi-trait genomic mating for a k trait problem (assuming maximization is sought for traits) is defined by the optimization problem which seeks minimization of $-Gain(P)_i$, $-Usefulness(P)_i$ for $i = 1 :, 2, \ldots, k$ and $Inbreeding(P)$ with respect to mating plan P.

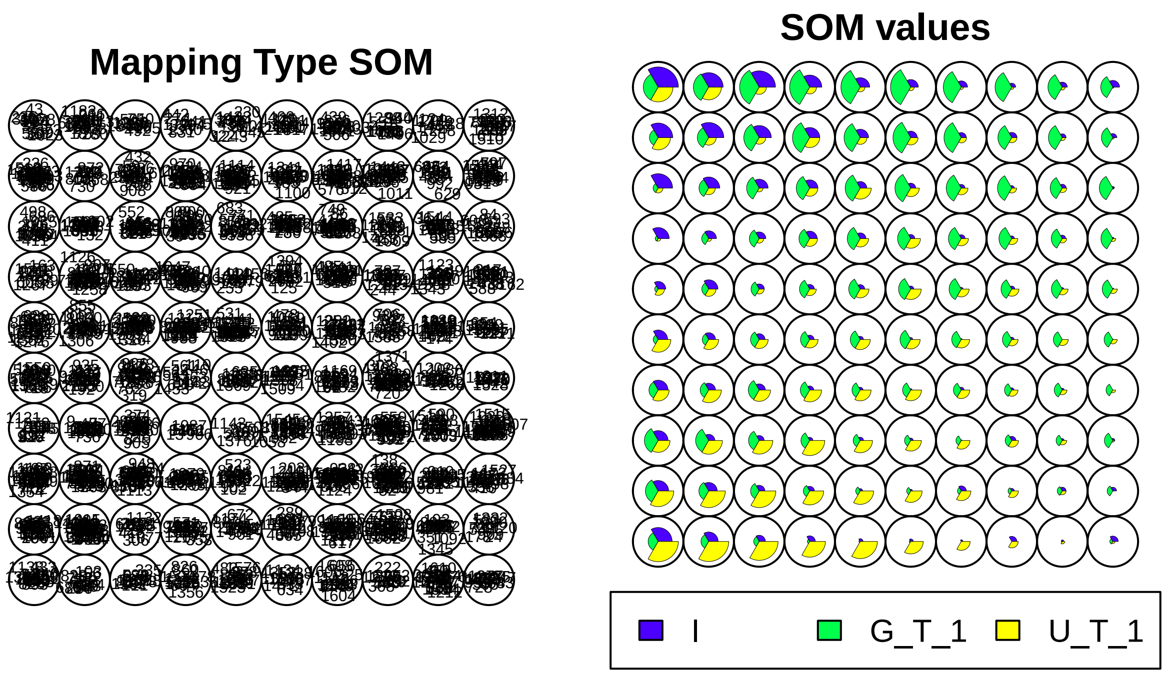
```
N=10
nmarkers=200
Markers<-c()
for (i in 1:N){
  Markers<-rbind(Markers,rbinom(nmarkers, 2,.1)-1)
}


Markers2<-c()
for (i in 1:N){
  Markers2<-rbind(Markers2,rbinom(nmarkers, 2,.1)-1)
}

Markers[1:5,1:5]
#>      [,1] [,2] [,3] [,4] [,5]
#> [1,]   -1    0   -1   -1    0
#> [2,]   -1    0   -1   -1   -1
#> [3,]   -1   -1   -1   -1   -1
#> [4,]    0   -1   -1   -1   -1
#> [5,]   -1   -1   -1   -1   -1
```

```r
K=Amat.pieces(rbind(Markers), pieces=5)

K2=Amat.pieces(rbind(Markers,Markers2), pieces=5)
K[1:5,1:5]
#>             [,1]        [,2]        [,3]        [,4]        [,5]
#> [1,]  0.93566898 -0.20295983 -0.18483842 -0.01872546 -0.11537300
#> [2,] -0.20295983  1.01419511 -0.05496829 -0.10027182  0.01449713
#> [3,] -0.18483842 -0.05496829  1.08064029 -0.17275748 -0.08819088
#> [4,] -0.01872546 -0.10027182 -0.17275748  1.02023558 -0.01268499
#> [5,] -0.11537300  0.01449713 -0.08819088 -0.01268499  0.76653579
rownames(Markers)<-paste("l", 1:nrow(Markers),sep="_")
rownames(Markers2)<-paste("l", (nrow(Markers)+1):(nrow(Markers)+
                                                    nrow(Markers2)),sep="_")
rownames(K2)<-colnames(K2)<-c(rownames(Markers),rownames(Markers2))
rownames(K)<-colnames(K)<-c(rownames(Markers))


markereffects<-rep(0,nmarkers)
markereffects[sample(1:nmarkers,nmarkers/2)]<-rnorm(nmarkers/2)
markereffects2<-rep(0,nmarkers)
markereffects2[sample(1:nmarkers,nmarkers/2)]<-rnorm(nmarkers/2)

markermap=as.matrix(data.frame(chr=rep(1,nmarkers),
                               pos=seq(0,1,length=nmarkers)))

which.max(Markers%*%markereffects)
#> [1] 8

gasols4<-getGaSolutionsFrontierMultiTrait(Markers=Markers,
             Markers2=Markers2,K=K2,
             markereffectslist=list(markereffects,markereffects2),
            markermap=markermap,nmates=20,npopGA=100, nitGA=300,
            plotiters=F, mc.cores=1, mutprob=0.999,method=2,
                               type=0L, generation=3L)


gasols4[[1]][1:5,]
#>             I   -G_T_1    -U_T_1    -G_T_2     -U_T_2
#> [1,] 43.54325 85.70087 -720.3798  8.354192  -911.5120
#> [2,] 48.26093 58.46630 -591.3124 17.650471 -1206.4278
#> [3,] 34.74117 62.55688 -421.1700 16.358305  -943.0785
#> [4,] 50.11100 51.16071 -477.6654 12.946827 -1069.3394
#> [5,] 28.40823 55.22396 -490.2665 31.735891  -727.5563
```

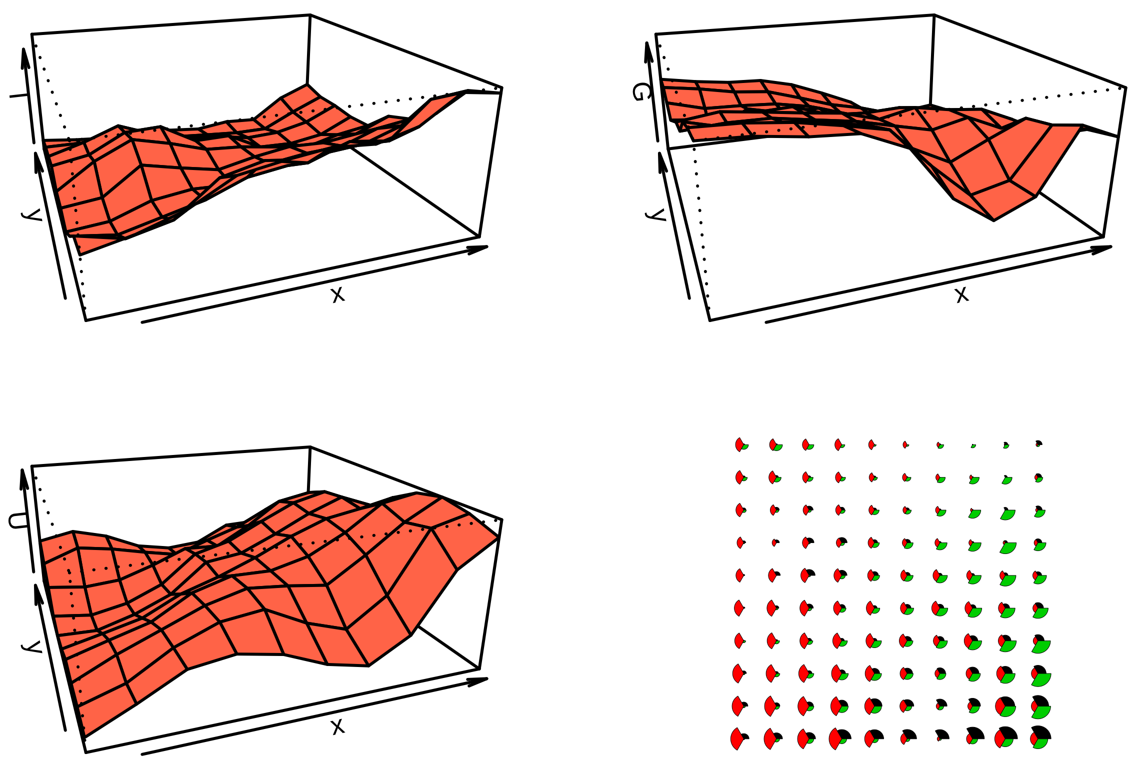
Plotting the results.

Self-Organizing Maps (SOMs) [Kohonen [1981, 1998]] have been recommended for visualizing the pareto optimal solutions for high dimensional multi-objective problems [Obayashi and Sasaki [2003]]. Neural networks are used in learning tasks that are too complex for human brain to comprehend and SOM is a unsupervised neural networks technique for organizing complex or vast amounts of data by providing lower dimensional representations of data in manner that is most easily understood. Specifically, SOMs are a type of artificial neural network (ANN) that provides a topology preserving mapping from the high dimensional space to map units. The property of topology preserving means that the mapping preserves the relative distance between the points; points that are near each other in the input space are mapped to nearby map units in the SOM. The SOM can thus serve as a cluster analyzing tool of high-dimensional data and be used as a visual aid in determining a 'good' solution on the frontier surface.

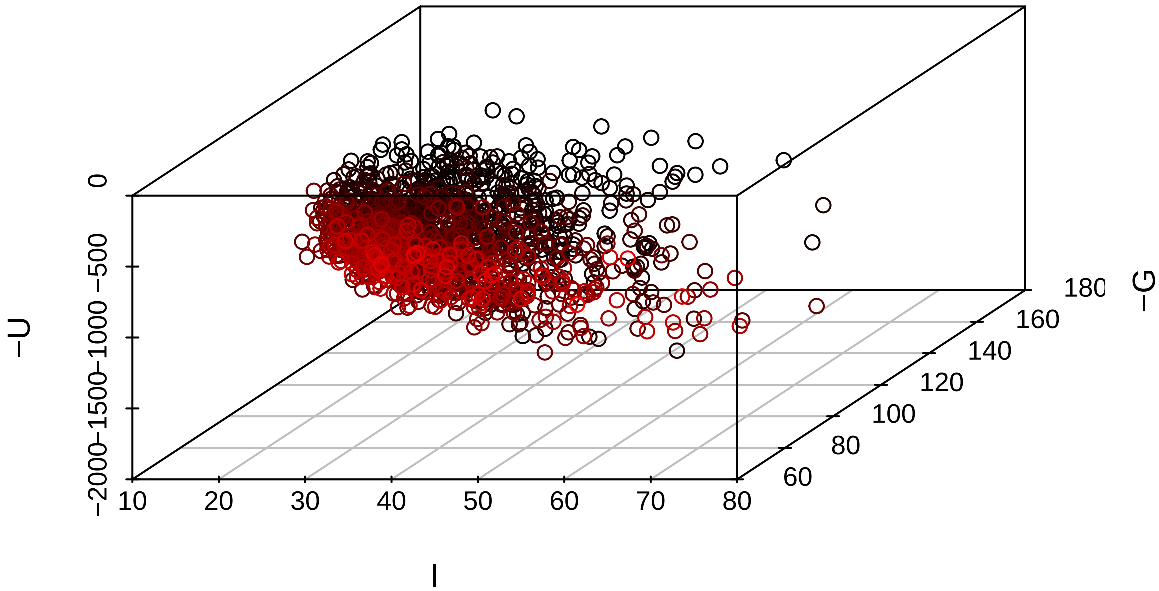
```
GenomicMating::plotGM(GMsols=gasols4, type="SOM", traitnum=1)
```




```
GenomicMating::plotGM(GMsols=gasols4, type="SOM2", traitnum=1)
```



```r
GenomicMating::plotGM(GMsols=gasols4, type="3D",
                      idealsol=c(0,-100,-100,-100,-100))
```

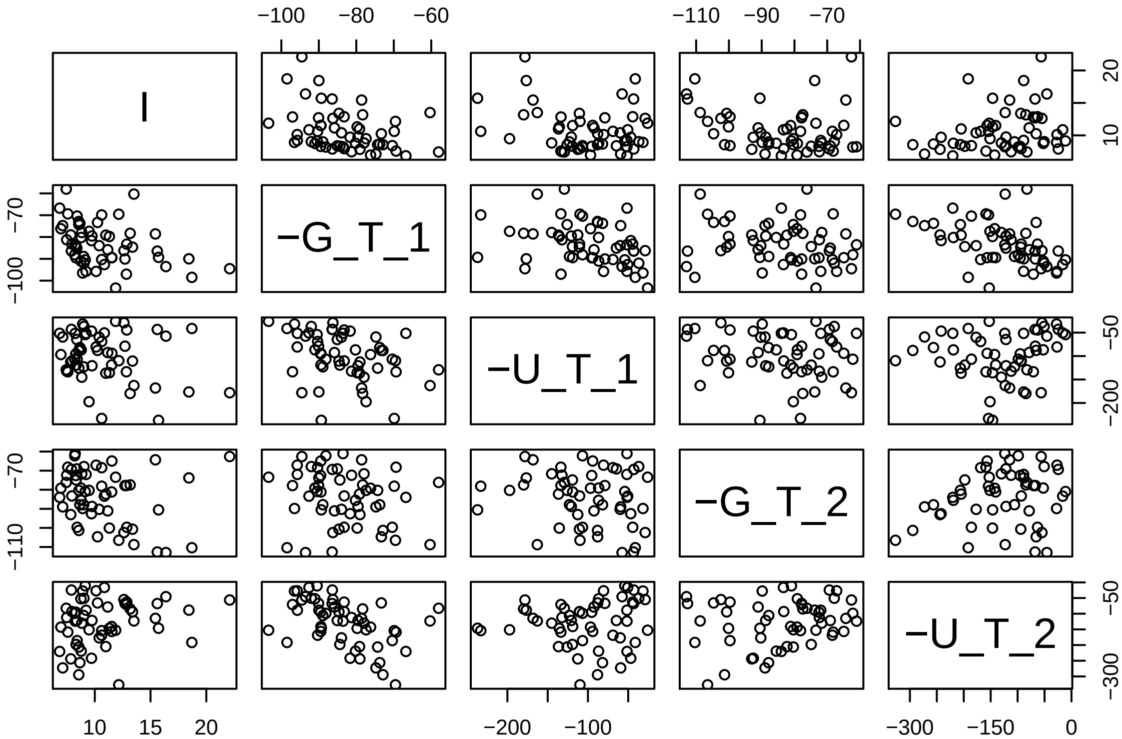


Example with simcross

```r
N=10
nmarkers=200
Markers<-c()
for (i in 1:N){
  Markers<-rbind(Markers,rbinom(nmarkers, 2,.1)-1)
}

Markers2<-c()
for (i in 1:N){
  Markers2<-rbind(Markers2,rbinom(nmarkers, 2,.1)-1)
}

markereffects<-rep(0,nmarkers)
markereffects[sample(1:nmarkers,nmarkers/2)]<-rnorm(nmarkers/2)
Markers[1:5,1:5]
#>      [,1] [,2] [,3] [,4] [,5]
#> [1,]   -1   -1   -1   -1   -1
#> [2,]   -1   -1   -1    0   -1
#> [3,]   -1   -1   -1   -1   -1
#> [4,]   -1   -1   -1   -1    0
#> [5,]   -1   -1    0    0   -1
library(parallel)
K=Amat.pieces(rbind(Markers), pieces=5)

K2=Amat.pieces(rbind(Markers,Markers2), pieces=5)
K[1:5,1:5]
#>             [,1]        [,2]        [,3]        [,4]        [,5]
```

```
#> [1,]  0.93108298 -0.07313643 -0.07032349 -0.16596343 -0.10407876
#> [2,] -0.07313643  0.94796062 -0.09001406 -0.15752461 -0.03938115
#> [3,] -0.07032349 -0.09001406  0.86919831 -0.04219409 -0.14908579
#> [4,] -0.16596343 -0.15752461 -0.04219409  0.98734177 -0.10407876
#> [5,] -0.10407876 -0.03938115 -0.14908579 -0.10407876  0.88607595
rownames(Markers)<-paste("l", 1:nrow(Markers),sep="_")
rownames(Markers2)<-paste("l", (nrow(Markers)+1):(nrow(Markers)+nrow(Markers2)),sep="_")
rownames(K2)<-colnames(K2)<-c(rownames(Markers),rownames(Markers2))
rownames(K)<-colnames(K)<-c(rownames(Markers))

markereffects<-rep(0,nmarkers)
markereffects[sample(1:nmarkers,nmarkers/2)]<-rnorm(nmarkers/2)
markereffects2<-rep(0,nmarkers)
markereffects2[sample(1:nmarkers,nmarkers/2)]<-rnorm(nmarkers/2)

markermap=as.matrix(data.frame(chr=rep(1,nmarkers),pos=seq(0,1,length=nmarkers)))

map<-cbind(1:nmarkers,1,seq(0,1e+2, length=nmarkers))
map<-qtl::sim.map(len=c(.5), n.mar=nmarkers, anchor.tel=TRUE,
  include.x=FALSE, sex.sp=FALSE, eq.spacing=FALSE)
map<-cbind(1:nmarkers,1,map[[1]])
dim(map)
#> [1] 200   3

rownames(K)<-colnames(K)<-rownames(Markers)<-1:nrow(Markers)
rownames(map)<-1:ncol(Markers)
sum(is.na(map))
#> [1] 0

gasols5<-getGaSolutionsFrontierMultiTraitSimcross(Markers=Markers,
K=K,map=map, markereffectslist=list(markereffects, markereffects2),
nmates=10,npopGA=10, nitGA=10,mc.cores=1,mutprob=0.999,
nSim = 10,simtype="riself")

gasols5[[1]]
#>               I     -G_T_1     -U_T_1     -G_T_2     -U_T_2
#>  [1,] 11.511955  -89.42812 -118.95484  -81.15413 -140.76810
#>  [2,] 16.378340  -93.54979  -57.55765 -112.93250  -45.63020
#>  [3,]  9.219409  -95.72727  -51.60071  -71.97526  -88.86222
#>  [4,] 10.246132  -73.32129  -87.92383 -104.66198  -65.55716
#>  [5,] 11.533052  -89.46690  -94.11350  -64.94147 -156.55879
#>  [6,] 10.618847  -90.32322  -68.93850  -68.43789 -168.61256
#>  [7,]  8.917018  -96.49061  -31.67968  -89.83537  -27.65235
#>  [8,]  8.755274  -91.03699  -86.64706  -79.10579  -52.28640
#>  [9,] 18.706048  -98.48279  -41.22884 -110.38293 -191.68027
#> [10,]  9.704641  -81.64623  -46.80845  -92.56239 -241.99913
#> [11,] 15.717300  -89.33515 -237.05251  -90.54179 -146.37165
#> [12,]  9.746835  -79.53187 -120.67474  -88.76405 -121.48727
#> [13,] 13.509142  -60.31960 -162.88309 -108.77262 -123.05626
#> [14,] 11.308017  -79.76206 -135.58519 -100.11207 -146.60603
#> [15,] 10.991561  -79.13624 -136.54118  -82.29167 -204.86968
#> [16,]  8.783404  -80.22201  -86.77896  -85.55013 -219.26736
#> [17,] 11.160338  -85.82594  -92.49124  -91.07850  -78.55086
```

```
#> [18,] 12.869198  -83.24485  -44.39146  -99.66911  -62.99928
#> [19,]  8.241913  -83.54077  -51.54805  -61.05230 -123.80314
#> [20,] 11.870605 -103.40045  -25.92094  -73.43735 -152.69103
#> [21,]  9.029536  -92.00829  -36.86997  -67.85598  -50.71573
#> [22,] 15.436006  -78.57365 -168.17441  -64.31284 -114.79293
#> [23,] 12.707454  -90.07347  -78.95943  -77.80096  -67.07833
#> [24,] 12.616034  -86.24583  -28.93304 -102.44062  -54.81468
#> [25,]  8.417722  -70.39910 -106.72386  -99.67342 -185.78431
#> [26,]  8.572433  -74.30036 -125.77366  -80.34260 -206.21418
#> [27,]  9.142053  -90.45552  -54.23178  -80.93002  -11.07080
#> [28,]  8.199719  -88.12791 -106.75458  -62.18646  -98.76636
#> [29,]  8.565401  -72.88078  -88.09245 -101.32400 -294.60746
#> [30,]  8.389592  -85.07105  -64.64386  -69.06817 -121.73370
#> [31,] 13.171589  -78.30240 -179.82443  -77.45976  -84.64497
#> [32,] 13.382560  -84.62681 -110.63783 -100.67709  -94.12912
#> [33,] 10.119550  -95.76153  -80.58076  -67.12203  -26.80564
#> [34,] 12.834037  -97.02341 -133.46260  -77.80646  -70.76737
#> [35,]  7.123769  -74.76489  -59.31558  -88.97567 -272.76426
#> [36,]  6.962025  -76.19119  -96.78514  -79.20631 -142.63898
#> [37,]  7.904360  -86.40421  -43.04969  -69.36178  -24.49443
#> [38,]  9.486639  -77.39384 -197.29618  -80.34131 -151.25801
#> [39,]  8.663854  -73.74095  -81.89922  -87.88847 -256.17409
#> [40,]  8.832630  -77.93222 -144.99184  -71.66150 -130.10766
#> [41,] 15.604782  -86.44793  -43.28207 -112.62724  -67.64235
#> [42,]  7.482419  -81.25863 -132.28964  -72.40066 -112.08418
#> [43,] 12.151899  -69.54778 -109.87299 -106.49724 -326.75400
#> [44,]  8.326301  -84.38616 -119.27033  -74.87594 -197.99576
#> [45,] 18.438819  -89.98131 -176.54856  -73.80950  -88.65351
#> [46,] 22.095640  -94.52426 -178.20132  -62.60244  -56.26761
#> [47,]  8.305204  -89.51894  -95.27824  -72.52471  -95.18444
#> [48,]  7.440225  -58.02665 -129.34075  -76.17446  -82.62927
#> [49,]  7.946554  -83.22341 -110.33674  -83.25334  -93.17724
#> [50,] 10.393812  -83.95842  -59.93762  -90.27061 -176.98554
#> [51,]  7.848101  -79.03384 -112.63243  -93.01941 -244.02179
#> [52,]  6.849508  -66.76018  -51.44348  -83.99706 -220.37723
#> [53,] 10.857947  -92.63044  -50.62180  -83.36310  -16.27624
#> [54,] 10.618847  -69.86919 -233.07526  -78.14872 -153.98588
#> [55,]  9.001406  -88.92734 -122.94658  -87.84064 -105.71094
#> [56,]  7.580872  -69.38772 -133.59321  -68.18391 -158.62278
pairs(gasols5[[1]])
```



References

Deniz Akdemir and Julio I Sánchez. Efficient breeding by genomic mating. *Frontiers in genetics*, 7:210, 2016.

AR Hallauer and MJ Carena Filho. Jbm (2010) quantitative genetics in maize breeding, handbook of plant breeding.

Teuvo Kohonen. Automatic formation of topologicalmaps of patterns in a self-organizing system. *Scand. Conf. on Image Analysis*, pages 214–220, 1981.

Teuvo Kohonen. The self-organizing map. *Neurocomputing*, 21(1):1–6, 1998.

Andres Legarra, I Aguilar, and I Misztal. A relationship matrix including full pedigree and genomic information. *Journal of dairy science*, 92(9):4656–4663, 2009.

Anne-Louise Leutenegger, Bernard Prum, Emmanuelle Génin, Christophe Verny, Arnaud Lemainque, Françoise Clerget-Darpoux, and Elizabeth A Thompson. Estimation of the inbreeding coefficient through use of genomic data. *The American Journal of Human Genetics*, 73(3):516–523, 2003.

Shigeru Obayashi and Daisuke Sasaki. Visualization and data mining of pareto solutions using self-organizing map. In *EMO*, pages 796–809. Springer, 2003.

RL Quaas. Additive genetic model with groups and relationships. *Journal of Dairy Science*, 71(5):1338–1345, 1988.

Jinliang Wang. Coancestry: a program for simulating, estimating and analysing relatedness and inbreeding coefficients. *Molecular ecology resources*, 11(1):141–145, 2011.