Tutorial: optimizing crop collections

M. Humberto Reyes-Valdés

October 8, 2019

Optimization of the composition of crop collections.

This tutorial shows the use of an R code for optimization of crop collections, based on the Kullback-Leibler divergence. The data is a table of numbers of Triticum accessions from the CIMMYT wheat collection for class and type, along with a set of ideal or optimal proportions.

Read the table

```
tab<-read.csv("data/TriticumClassType_Paper.csv",head=T)
tab
```

```
##       class  type total ideal
## 1  Breeders Bread 30934    25
## 2  Breeders Durum  9289     4
## 3  Breeders Other   555     1
## 4  Cultivar Bread 36772    20
## 5  Cultivar Durum  4242     2
## 6  Cultivar Other   452     1
## 7  Landrace Bread 38283    32
## 8  Landrace Durum  7442     4
## 9  Landrace Other  1607     1
## 10     Wild   All  5660    10
```

This table contains class, type, actual accession numbers and ideal proportions in this order.

Relative balance

To make the basic calculations, we need the actual and optimal proportions from the table. The actual proportions can be calculated as follows:

```
a<-tab$total/sum(tab$total)
```

Since the optimal proportions are expressed as a %, we need to convert them to fractions:

```
o<-tab$ideal/100
```

Load the R script

```
source("RelativeBalance.R")
```

Calculate the Kullback-Leibler divergence between actual and optimal proportions

```
divergence(o,a) #Optimal and actual argument values must be in that order
```

```
## [1] 0.08017841
```

Calculate the relative balance of the collection composition

```
r.balance(o,a)
```

```
## [1] 0.9879319
```

Coverage

To calculate the coverage of the collection, assume a target of 150,000 accessions

```
#Actual size of the Triticum collection
sum(tab$total)
```

```
## [1] 135236
```

```
#Coverage
#---The arguments are target size, actual size, actual proportions and optimal proportions
coverage(150000, sum(tab$total), a, o)
```

```
## [1] 0.8917897
```

Optimization examples

Assume that we want to allocate 5000 accessions, and we need to decide which groups must be enriched so the relative balance RB has a maximum growth.

```
#The first argument is table with end group names, actual numbers and optimal proportions
#The second argument is the number of accessions to be added
#Creating the right table
endGroup<-paste(tab$class,tab$type,sep="-")
table<-data.frame("endGroup" = endGroup, actual = tab$total, ideal = tab$ideal)
table
```

```
##           endGroup actual ideal
## 1   Breeders-Bread  30934    25
## 2   Breeders-Durum   9289     4
## 3   Breeders-Other    555     1
## 4   Cultivar-Bread  36772    20
## 5   Cultivar-Durum   4242     2
## 6   Cultivar-Other    452     1
## 7   Landrace-Bread  38283    32
## 8   Landrace-Durum   7442     4
## 9   Landrace-Other   1607     1
## 10        Wild-All   5660    10
```

```
#Names do not have to be the same as in the example
#Allocate accessions
allocateAccessions(table,5000)
```

```
## $allocation
##         end.group allocate
## 1 Breeders-Other      417
## 2 Cultivar-Other      520
## 3       Wild-All     4063
##
## $RB
## [1] 0.9930412
```

```
#The first object in the output list is a table with recommended allocations
#The second object is the RB to be obtained after allocation
```

Now, assume that we want to remove 5000 accessions

```
removeAccessions(table,5000)
```

```
## $allocation
##         end.group remove
## 1 Breeders-Durum   2256
## 2 Cultivar-Bread   1609
## 3 Cultivar-Durum    726
## 4 Landrace-Durum    409
##
## $RB
## [1] 0.990557
```

```
#The first object in the output list is a table with recommended removals
#The second object is the RB to be obtained after accession removals
```

Now, assume we want a collection size of 150,000, and want to remove and add accessions

```
makeEven(table,150000)
```

```
##                end action
## 1  Breeders-Bread   6566
## 2  Breeders-Durum  -3289
## 3  Breeders-Other    945
## 4  Cultivar-Bread  -6772
## 5  Cultivar-Durum  -1242
## 6  Cultivar-Other   1048
## 7  Landrace-Bread   9717
## 8  Landrace-Durum  -1442
## 9  Landrace-Other   -107
## 10       Wild-All   9340
```

This is a rather arithmetical calculation. The actions column indicates how many accessions to add (positive numbers), and how many accessions to remove (negative numbers).