ゲノミックセレクション支援プログラム - SL96　説明書

（以降、SL96 と表記）

2016 年 3 月 3 日

国立研究開発法人　農業・食品産業技術総合研究機構
野菜茶業研究所
山本英司

著作権、免責事項等

1. このプログラムおよびマニュアルを利用することで生じるすべての損害、損失について、国立研究開発法人　農業・食品産業技術総合研究機構はその責を負いません。
2. このプログラムおよびマニュアルの内容の全部または一部について、無断での引用・転載はしないでください。また、内容の全部または一部について、無断で改変を行うことはできません。
3. このプログラムおよびマニュアルは、予告なしに内容の変更、削除をする場合がありますが、あらかじめご了承下さい。

2016 年 3 月 3 日
国立研究開発法人　農業・食品産業技術総合研究機構
山本英司

Copyright and disclaimer

1. NARO Institute does not take responsibility for any loss or damage caused by the use of this program and manual.
2. All the contents of this program and manual are provided on an "as is" basis, without warranty of any kind, including without limitation the warranties of merchantability, fitness for a particular purpose and non-infringement.
3. Please note that the contents of this program and manual may be changed or deleted without notice.

3 Mar 2016
NARO Institute of Vegetable and Tea Science
Eiji Yamamoto

.

目次

0. このファイルについて

＜はじめに＞

農研機構の 2015 年度普及成果情報「トマトの高品質多収育種のためのゲノム情報に基づく高精度形質予測」をお読みください。このプログラムを用いて得られた成果がわかりやすく説明されています。

http://www.naro.affrc.go.jp/project/results/laboratory/vegetea/2015/*****.html

（＊公開準備中）

＜これは何か？＞

ゲノミックセレクション（GS）を利用した育種設計において、

1. GS モデルの構築
2. 交差検証による GS モデルの予測精度評価
3. シミュレーションによる育種効果の推定

を行う。

なお 1. については、既存のプログラムを多数引用している（P2 参照）。

＜特徴＞

- 統計解析ソフト「**R**」上で操作する。

＜フォルダの中身について＞

1. **SL96 説明書.docx**：このワード文書。
2. **scripts**：解析用プログラムを格納しているフォルダ。

   「scripts」の中のファイル一覧（＊2016 年 3 月 3 日現在）

   funcsCV.R

   funcsData.R

   funcsSim.R

   funcsWGP.R
3. **SL96.R**：「scripts」内のプログラムを実行するためのファイル。
4. **testG.txt**：デモ用遺伝子型ファイル、その１。
5. **trainG.txt**：デモ用遺伝子型ファイル、その２。
6. **trainP.txt**：デモ用形質データファイル。

1. 準備

<1-1. R のインストール>

既存のソフトを利用しているので、先にそれらをインストールする必要がある。

1. 「**R**」をインストールする。

   http://www.r-project.org/ に行けばダウンロードできる。

2. 「**R**」を起動する。するとコンソール画面というもの（図 1-1）が現れる。

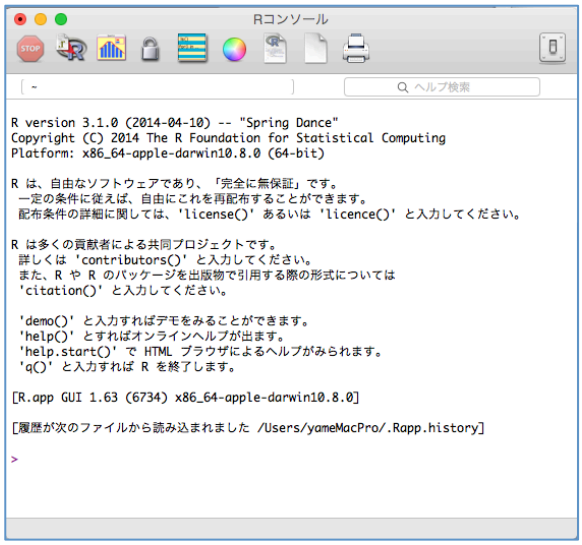


図 1–1. MAC 版「**R**」のコンソール画面。

3. コンソール画面の「 > 」の所に、以下のテキストを入力し、リターンキーを押す。

```
install.packages("rrBLUP")
install.packages("randomForest")
```

4. 以上で必要なプログラムのインストールは終了。

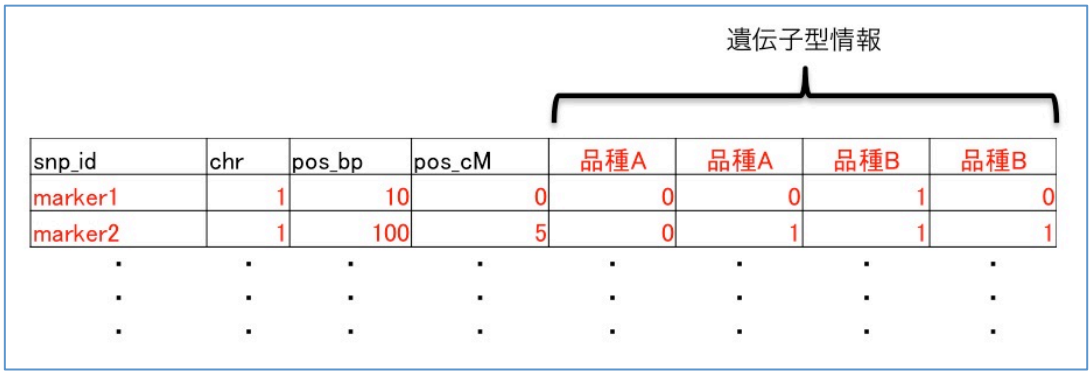
＜1-2. データのフォーマット＞

入力情報となるファイルは、'遺伝子型ファイル' と '形質値ファイル' の２種類。

「**SL96**」では、以下に示すフォーマットに厳密に即している必要がある。

1.　遺伝子型ファイルのフォーマット

'タブ区切り' もしくは 'スペース区切り' のテキストファイル。行には DNA マーカー情報が、列には各個体の遺伝子型情報が整理される。マーカー数や個体数に制限はない。



snp_id	chr	pos_bp	pos_cM	品種A	品種A	品種B	品種B
marker1	1	10	0	0	0	1	0
marker2	1	100	5	0	1	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

図 1-2-1. 遺伝子型ファイルのフォーマット。赤字の部分はデータによって異なる。

snp_id：DNA マーカーの名前。適当でも良い。

chr：各マーカーの染色体番号。

pos_bp：各マーカーの物理地図上の位置。

pos_cM：各マーカーの連鎖地図上の位置。

遺伝子型情報：各個体の名称は、アルファベットと数字の組み合わせで示す（例. hinshu、cv1 など）。ただし、先頭の文字はアルファベットでなければならない。各個体につき２列で構成される（つまり、各列は半数体に該当する）。マーカー遺伝子型は ‘0’ もしくは ‘1’ で示す。欠測値は含まない。任意のプラットフォームで得られた遺伝子型情報について遺伝解析ソフトウェア Beagle* を使って処理すると、このようなフォーマットになる。

* https://faculty.washington.edu/browning/beagle/beagle.html

2.　形質値ファイルのフォーマット

'タブ区切り' もしくは 'スペース区切り' のテキストファイル。行には個体、列には形質のデータがまとめられる。形質の数に制限はない。

gid	形質１	形質２	・・・
品種A	5.8	1250	・・・
品種B	3.7	1777	・・・
品種C	2.6	986	・・・
・	・	・	
・	・	・	
・	・	・	

図 1-2-2. 形質値ファイルのフォーマット。赤字の部分はデータによって異なる。

＜1-3. SL96 を使用するための準備＞

1. 「**SL96**」のフォルダをデスクトップに置く。

2. 「**R**」を起動する。

3. コンソール画面の「>」の後ろに以下のテキストを入力し、リターンキーを押す。

```
＊mac の場合は、setwd("~/desktop/SL96")
＊windows の場合は、setwd("c:/任意/desktop/SL96")
source("SL96.R")
```

- 1 行目は作業ディレクトリの指定。「これからデスクトップ上の SL96 というフォルダの中で作業します」という意味。
- 2 行目は今回の解析で用いる一連のプログラムを使用するために必要。

4. 以上＜1-3.＞で示した作業は、新たに解析を始めるたびに実施する必要がある。これ以降はやりたい解析によって操作が違ってくる。

2. ゲノミックセレクション

Genomic Selection：GS

ゲノムワイドな DNA マーカー情報から形質値を予測する計算式（GS モデル）を構築し、その GS モデルが予測する形質値（ゲノム育種価）を用いて育種選抜を行うこと。

GS モデルを構築するためには、様々な統計手法が提唱されている。ここでは、

- GBLUP
- Reproducing kernel Hilbert space regression (RKHS)
- Random forest (RF)

の３手法が利用可能である。

<2-1. 統計的手法の選択と評価　～交差検証～>

GS において最も重要なのは、GS モデルが未知の形質値をうまく予測できるかどうかである。その点を評価するために、GS モデル構築のための集団の情報（トレーニング集団）を 'ひとまずの GS モデル構築用' と '答え合わせ用' に分けて、GS モデルの精度を評価する**"交差検証"**を行う。

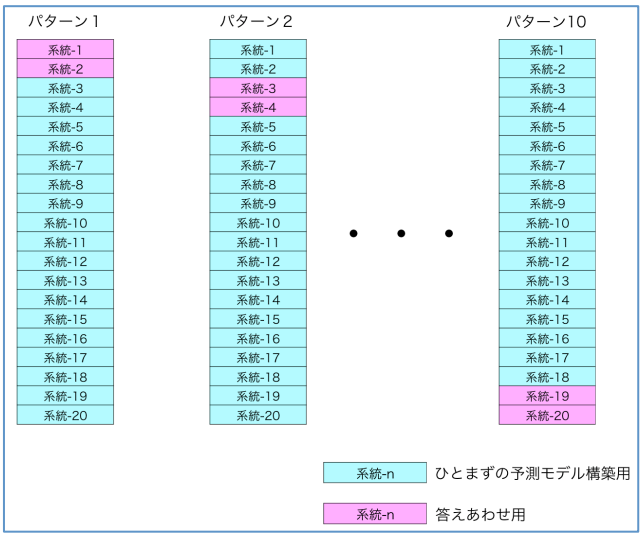


図 2-1. 交差検証の概念図。
20 系統のトレーニング集団について 10 分割交差検証を行う場合。最後に各パターンで計算された、各系統の'ひとまずの予測値'と実際の形質値との相関係数を計算し、予測精度の指標とする。

＊ とりあえずやってみる

1. コンソール画面に以下のようなテキストを入力

   CV("trainG.txt", "trainP.txt", "GBLUP")

   ＜解説＞ CV("*遺伝子型ファイル名*", "*形質値ファイル名*", "*手法*")

   遺伝子型ファイル名：テキストファイルの名前（拡張子付き）

   形質値ファイル名：テキストファイルの名前（拡張子付き）

   手法："GBLUP", "RKHS", "RF"の３種類から選択

2. フォルダ内に、”trainG_trainP_10CV1_GBLUP_*xxxxxx*.csv”
   という名前のファイルが作成される。

   ＜解説＞ "*遺伝子型ファイル名_形質値ファイル名_x*CV*y_手法_任意の値*.csv"

   x：データを何分割して交差検証を行ったかを示す（デフォルトは 10）。

   y：交差検証の反復回数を示す（デフォルトは 1）。

   任意の値：トレーニングデータを分割する際に使用したシード値。特定の数値を指定すれば、分割のパターンが同じになる。

3. ファイルをエクセルなどで開いて、結果を確認。

   ＜具体例＞

	TFW	SSC
1	0.286172344	0.519280542

   ＊TFW、SSC は形質の名前。その下の数値は、交差検証における予測値と実測値との相関係数。

＊ 細かいパラメータの設定

交差検証を実施する CV では以下のような形式で細かい設定が変更できる。
CV("*遺伝子型ファイル名*", "*形質値ファイル名*", "*手法*", Fold=*分割数*, Repeat=*反復回数*, Seed=*シード値*, Trait=*個別の形質名*)

分割数：交差検証において、データを何分割するかを指定する。デフォルトは 10。分割数とトレーニング集団の個体数が同じ場合は、leave-one-out cross validation（LOOCV）と呼ばれる。

反復回数：交差検証を何回繰り返すかを指定する。デフォルトは 1 。実際の研究では、分割のパターンが異なる交差検証を何回か実施し、その平均値を指標に予測精度を評価する。5 回くらいが推奨。LOOCV の場合は 1 回で OK（＊分割パターンは 1 種類しかない）。

シード値：データの分割パターンの決定には、乱数を用いている。そのため、同じデータを用いて同じ手順の解析を行っても、試行ごとに結果は微妙に異なる。シード値を指定すれば、分割のパターンを一致させることができるため、同じ手順の解析を行えば、まったく同じ結果を再現することができる。

個別の形質名：特に指定がなければ、形質値ファイルに含まれるすべての形質について解析が行われるが、特定の形質のみ解析したい場合は、ここで指定する。1 つの場合は、Trait="*形質 1*"、2 つ以上の場合は、Trait=c("*形質 1*","*形質 2*")のように入力する。

＜具体例＞
CV("trainG.txt", "trainP.txt", "GBLUP", Fold=10, Repeat=5, Seed=123, Trait=c("SSC", "TFW"))

＜2-2. GS モデルの構築と利用＞

交差検証によって適切な統計的手法が決定すれば、その手法を用いて GS モデルを構築する。（形質値が未知の）育種選抜用集団について、GS モデル構築に用いた際と同じ DNA マーカーセットを使って遺伝子型情報を取得し、GS モデルを当てはめることで形質値を予測し、優れた個体を選抜する。

＊ とりあえずやってみる

1. GS モデルを構築する。コンソール画面に以下のようなテキストを入力

   ```
   WGP("trainG.txt", "trainP.txt", "GBLUP", "SSC")
   ```

   ＜解説＞ WGP("*遺伝子型ファイル名*", "*形質値ファイル名*", "*手法*", "*形質*")

   遺伝子型ファイル名：テキストファイルの名前（拡張子付き）

   形質値ファイル名：テキストファイルの名前（拡張子付き）

   手法："GBLUP", "RKHS", "RF"の３種類から選択

   形質：モデルを構築したい形質名。

2. フォルダ内に、'trainG_trainP_SSC_GBLUP.bin' という名前のファイルが作成される。これが GS モデルのファイル。

   ＜解説＞ "*遺伝子型ファイル名_形質値ファイル名_形質名_手法*.bin" という構成。

3. GS モデルを用いて、形質値予測を行う。コンソール画面に以下のようなテキストを入力

   ```
   ans <- calcGEBV("testG.txt","trainG_trainP_SSC_GBLUP.bin")
   ```

   ＜解説＞ calcGEBV("*予測したい遺伝子型ファイル名*", "*モデルファイル名*")

   予測したい遺伝子型ファイル名：テキストファイルの名前（拡張子付き）

   モデルファイル名：GS モデルのファイルの名前（拡張子付き）

   ans <- ：calcGEBV の結果を ans というコードネームで「R」上に一時的に保存する

4. ans の中身を確認するために、コンソール画面に以下のようなテキストを入力

   ```
   ans
   ```

5. すると、コンソール画面上には、以下のように表示される

```
    gid     Yhat
1   gid1 5.799269
2   gid2 5.683069
        ...
```

＜解説＞

gid：各個体の ID

Yhat：各個体の予測値

6. 必要に応じて、この結果（ans）をテキストファイルとして保存。以下のようなテキストを入力

```
write.table(ans, "ans.txt", row.names=FALSE, quote=FALSE)
```

＜解説＞ write.table(*データのコードネーム*, "*出力するファイル名*", ...)

データのコードネーム： 今回の場合は ans。

出力するファイル名：最後が.txt で終われば、何でもよい。

＊row.names や quote については、ここでは説明を割愛。「**R**」の機能を紹介するウェブサイトなどを参考。

＊ 細かいパラメータの設定

WGP や calcGEBV については、細かいパラメータ設定は用意していない。

3. 育種シミュレーション

GS モデルはあくまで「形質値を予測する」ものであって、「どのような育種をすべきか？」を教えてくれるものではない。
この問題を解決するために、「思いつく様々な育種戦略を仮想的に実施して、どのような結果になるかを予測してみる」という方法がある。

＊ とりあえずやってみる

1. コンソール画面に以下のようなテキストを入力。この操作により、遺伝子型ファイルに含まれるハプロタイプに基づいた仮想ゲノムが作成される。

```
setSimGenome("trainG.txt")
```

＜解説＞ `setSimGenome("`*遺伝子型ファイル名*`")`

遺伝子型ファイル名：テキストファイルの名前（拡張子付き）。シミュレーションの基礎となる情報（スタートの集団や染色体数、組換え頻度など）は、このデータをもとに決定される。

2. コンソール画面に以下のようなテキストを入力

```
SimGenomeTable[[1]]          # この構文の解説は省略
```

するとコンソール画面上は、次のような表示される。

```
          gid HapCoode
1         SL1        1
2       SL1.1        2
3         SL2        3
4       SL2.1        4
5         SL3        5
```

＜解説＞

gid：遺伝子型ファイルに含まれるハプロタイプの ID。このプログラムでは、各個体の遺伝子型情報はハプロタイプごとに整理されるため（P3 参照）、SL1 の遺伝子型は、２つのハプロタイプ SL1 と SL1.1 によって示される。それ以外の品種についても同様。

HapCode：このプログラム上で、各ハプロタイプを識別する番号。

3. 品種 SL22 と品種 SL41 の交雑後代で、どのような個体が得られるかを予測したい。コンソール画面に以下のようなテキストを入力

```
ProgenyPop <- makeProgenies(SL22, SL41, 96)
```

＜解説＞ makeProgenies(*親 1*, *親 2*, *個体数*)

親 1 & *親 2*：交雑親の gid。

個体数：その交雑によって作成する個体数。

ProgenyPop <- ：makeProgenies の結果を ProgenyPop というコードネームで「R」上に一時的に保存

4. シミュレーションにより作成された品種 SL22 と品種 SL41 の交雑後代の遺伝子型情報をまとめたファイルを作成する。

```
makeSimGenoFile(ProgenyPop, "SL22xSL41n96.txt")
```

＜解説＞ makeSimGenoFile(*データのコードネーム*, "*遺伝子型ファイル名*")

データのコードネーム： 遺伝子型ファイルを作成したい集団のコードネーム。今回の場合は ProgenyPop。

遺伝子型ファイル名：出力するテキストファイルの名前（拡張子付き）。最後が.txt で終われば、何でもよい。

5. シミュレーションで作成された集団の形質値予測に使用する GS モデルを構築。コンソール画面に以下のようなテキストを入力

```
WGP("trainG.txt","trainP.txt","GBLUP", "SSC")
```

＜解説＞ P9 参照

6. コンソール画面に以下のようなテキストを入力

```
TePop <-
calcGEBV("SL22xSL41n96.txt","trainG_trainP_SSC_GBLUP.bin")
TrPop<- calcGEBV("trainG.txt","trainG_trainP_SSC_GBLUP.bin")
```

＜解説＞ calcGEBV("*遺伝子型ファイル名*", "*予測モデルファイル名*")

遺伝子型ファイル名：形質値を予測したい遺伝子型ファイル名（拡張子付き）

予測モデルファイル名：形質値予測に使用する、GS モデルファイル名（拡張子付き）

7. 箱ひげ図を使って、シミュレーションにより作成された集団の形質値と現在の品種群の形質値とを比較する。コンソール画面に以下のようなテキストを入力

```
boxplot(TePop$Yhat, TrPop$Yhat)
```

＜解説＞ boxplot(*データ 1*, *データ 2*)

データ 1 & *データ 2*：データは数字の集合。今回の例では、「TePop というデータの（$)、Yhat という列にまとめられているデータ」という意味。

8. 必要に応じて結果（TePop）をテキストファイルとして保存。以下のようなテキストを入力

```
write.table(TePop, "TePop.txt", row.names=FALSE, quote=FAL
SE)
```

＜解説＞ P10 参照

4. 解析例

＊「R」について、やや専門的な知識を必要とする。

＜目的＞

トマトの F_1 品種 96 点について、337 個の SNP マーカーによる遺伝子型データ（trainG.txt）と、"SSC"と"TFW"の２つの形質について調査したデータ（trainP.txt）がある。

ここで 96 品種の中から、いずれかの品種について組換え自殖系統群（RILs）を作成し、その中から"SSC"の高い系統を選抜したい。

すべての品種について組換え自殖系統群を作成するわけにはいかないので、GS モデルを使ったシミュレーションにより、"SSC"の高い系統を輩出しそうな品種を、今の段階で選びたい。

ここから "### ここまで" を、"R"コンソール上にコピー&ペーストすれば、同じ解析結果が得られる。

#1 解析の準備。R を起動し、コンソール画面に以下のテキストを入力

```
setwd("~/desktop/SL96")
source("SL96.R")
```

#2 "GBLUP", "RKHS", "RF"のうち、どの手法で GS モデルを構築すべきかを検討する。10 分割交差検証を５回繰り返し、予測精度を比較する。

```
CV("trainG.txt", "trainP.txt", "GBLUP", Fold=10, Repeat=5, Seed=123,
Trait="SSC")
CV("trainG.txt", "trainP.txt", "RKHS", Fold=10, Repeat=5, Seed=123,
Trait="SSC")
CV("trainG.txt", "trainP.txt", "RF", Fold=10, Repeat=5, Seed=123,
Trait="SSC")
```

#3 結果（trainG_trainP_10CV5_~~_123.csv）を確認したところ、形質"SSC"については"GBLUP"が、最も予測精度が高いと推定された（相関係数 r=0.592）。そこで、この手法を用いて GS モデルを構築する。

```
WGP("trainG.txt", "trainP.txt", "GBLUP", "SSC")
```

#4 現品種のゲノム育種価（GS モデルが予測する形質値）についても計算しておく。

```
Train <- calcGEBV("trainG.txt","trainG_trainP_SSC_GBLUP.bin")
```

#5 各品種について、96 系統からなる RILs の育成をシミュレーションする。

```
setSimGenome("trainG.txt")
```

#5-1 F1 品種である SL1 の F2 は、言い換えれば花粉親を SL1、種子親も SL1 とした交雑後代である。よって、このプログラムでは、以下のようになる。

```
f2SL1 <- makeProgenies(SL1, SL1, 96)
```

#5-2 この F2 集団 96 個体を単粒系統法で 5 世代進め、組換え自殖系統群を作成する。このプログラムでは、makeRILs(スタート集団, 進める世代数)という機能を用意している。

```
riSL1 <- makeRILs(f2SL1, 5)
```

#5-3 こうして作成された SL1 の RILs について、"SSC"のゲノム育種価を計算する。

```
makeSimGenoFile(riSL1, "riSL1.txt")
riSL1 <- calcGEBV("riSL1.txt","trainG_trainP_SSC_GBLUP.bin")
```

#6 品種群（トレーニング集団）と RILs（シミュレーション集団）とでゲノム育種価を比較する。

```
boxplot(Train$Yhat, riSL1$Yhat, main="SSC", names=c("Train","RILs"))
```

という作業を SL1 から SL103 まで延々と繰り返すわけだが、これは非常に面倒である。そこで、#5 以下を、次のようなコマンドに変更する。

```
setSimGenome("trainG.txt", List=TRUE)
```

List=TRUE というオプションを付けると、スタート集団の仮想ゲノム（SL1, SL2,...）は、TP というオブジェクトにまとめられる（TP[[1]]が SL1, TP[[2]]が SL2,...）。

```
All <- c()
for (i in 1:96) {
f2 <- makeProgenies(TP[[i]], TP[[i]], 96)
ri <- makeRILs(f2, 5)
makeSimGenoFile(ri, "ri.txt")
riSSC <- calcGEBV("ri.txt", "trainG_trainP_SSC_GBLUP.bin")
All <- cbind(All, riSSC$Yhat)
}
```

```
    colnames(All) <- names(TP)
    boxplot(All, main="SSC", names=colnames(All))
    abline(h=c(min(Train$Yhat), mean(Train$Yhat), max(Train$Yhat)),
    col="red")
### ここまで
```

この解析結果は図 4 のようになる。

96 品種の中でも、RILs を育成した場合に現在の品種よりも SSC が高い系統を輩出しそうなのは、SL22、SL41 をはじめとした 10 品種にも満たないことがわかる。

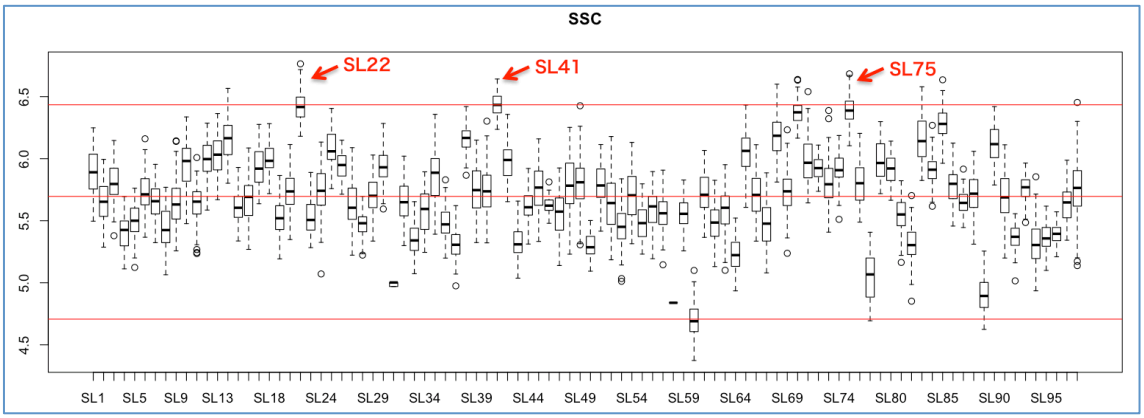


図 4. シミュレーションにより育成された RILs における SSC の分布。

赤線は上から、現在の品種群の最大値、平均値、最小値。

5. 参考文献

＜使用したソフトウェアについて＞

> R

- R Core Team. R: A Language and Environment for Statistical Computing. *R Foundation for Statistical Computing. Vienna, Austria.* http://www.R-project.org/ (2014)

> rrBLUP

- Endelman, J.B. Ridge regression and other kernels for genomic selection with R package rrBLUP. *Plant Genome* **4,** 250-255 (2011).

> randomForest

- Breiman, L. Random forests. *Machine Learning* **45,** 5-32 (2001).

＜GS モデル構築のための統計手法について＞

> GBLUP

- Habier, D., Fernando, R.L. & Dekkers, J.C. The impact of genetic relationship information on genome-assisted breeding values. *Genetics* **177,** 2389-2397 (2007)

> RKHS

- Gianola, D., Fernando, R.L., & Stella, A. Genomic-assisted prediction of genetic value with semiparametric procedures. *Genetics* **173,** 1761-1776 (2006).
- Gianola, D. & van Kaam, J.B. Reproducing kernel Hilbert spaces regression methods for genomic assisted prediction of quantitative traits. *Genetics* **178,** 2289-2303 (2008).
- de Los Campos, G., Gianola, D. & Rosa, G.J.M. Reproducing kernel Hilbert spaces regression: a general framework for genetic evaluation. *J. Anim. Sci.* **87,** 188 (2009).

> Random forest

- Breiman, L. Random forests. *Machine Learning* **45,** 5-32 (2001).