

# Evaluation des facteurs influençant la précision de la sélection génomique

Boîte à outil

Données génotypiques simulées par le coalescent  
**A faire varier avec les paramètres du coalescent**

Simulation données génotypiques

Données phénotypiques simulées  
**A faire varier pour modifier l'architecture génétique du trait (polygénique ou infinitésimale, héritabilité...)**

Simulation effet SNP et phénotypes

$X_{\text{mark}}$

$K = A_{\text{mark}}$  (si modèle add)

$Y$

Modèles Mixtes :

RRBLUP :  $Y = X_{\text{mark}} g + e$

$Y$  = Vecteur des données phénotypiques corrigées des effets environnementaux (ici absent)

$X_{\text{mark}}$  = Matrice des marqueurs génotypiques pour tous les individus à chaque loci

$g$  = Vecteur des effets aléatoires de chaque marqueur

$Xg = \text{GEBV}$  effet génétique **additif** aléatoire de chaque lignée

Validation croisée

=

GBLUP :  $Y = XB + ZU + e$ ,  $\text{var}(U) = A_{\text{mark}} \sigma_A^2$

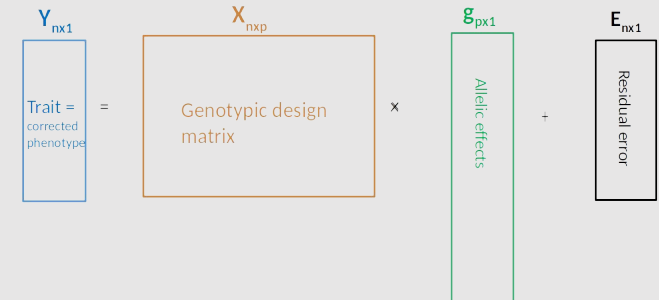
$Y$  = Vecteur des données phénotypiques (pas de corr env)

$X$  = matrice de liaison des effets fixes (environnementaux, ici absent)

$B$  = matrice des effets fixes (environnementaux, ici absent)

$U = \text{GEBV}$  = effet aléatoire de chaque lignée,

$Z$  = Identité si 1 obs/ind



Accuracy de prédiction en fonction de la variable modifiée

# Validation croisée

Population totale

1.

Population d'entraînement

$Y_{\text{train}}$   $X_{\text{train}}$

Population de validation

$Y_{\text{test}}$

1. Diviser le jeu de données
2. Ajuster un modèle de prédiction & prédire les valeurs des individus non phénotypés
3. Mesurer la qualité de prédiction

Répéter ces étapes avec une autre division

2.

RRBLUP

$\hat{g}$

$X_{\text{test}}$   $\hat{g}$

3.

Accuracy =  $\text{corr}(Y_{\text{test}}, X_{\text{test}} \hat{g})$

$Y_{\text{nx1}}$   
Trait =  
corrected  
phenotype

$X_{\text{nxp}}$

Genotypic design  
matrix

$\times$

$g_{\text{px1}}$

Allelic effects

$+$

$E_{\text{nx1}}$

Residual error

# Comprendre la généalogie pour simuler des “likely” données génotypiques

Introduction très (très) rapide à la notion de coalescence

- Il s’agit d’expliciter la généalogie de gènes (et non d’individus) dans un échantillon de copies de ces gènes (des séquences individuelles)
- Cette généalogie est basée sur la probabilité que deux séquences de la génération  $n$  proviennent de la même séquence à la génération  $n-1$
- En remontant dans le passé, un ancêtre commun à toutes ces séquences apparaît
- Pour rendre compte du polymorphisme dans l’échantillon, il est nécessaire de distribuer des mutations sur la généalogie qui mène jusqu’à l’ancêtre

Pour comprendre : jouez avec

<https://phytools.shinyapps.io/coalescent-plot/>

# Théorie de la coalescence pour simuler une évolution neutre de séquences nucléotidiques

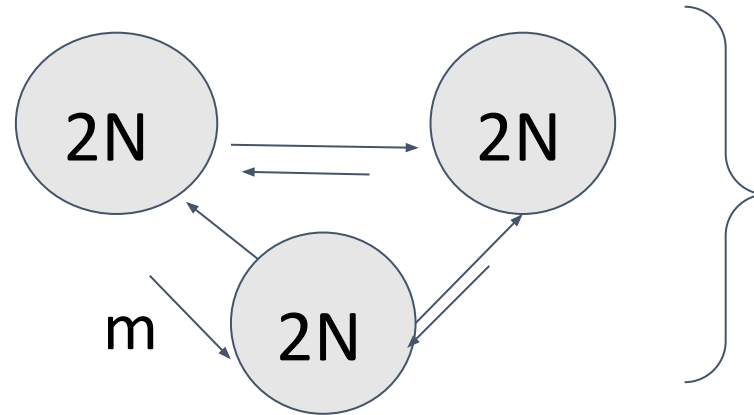
La taille : force de la dérive,  $2N$  en panmixie pour une diploïde,

La mutation :  $\mu$  par base,  $4N\mu$  : le nombre de mutants par génération

La migration :  $m$  le taux,  $4Nm$  le nombre de migrants par génération

La recombinaison :  $r$  par base,  $4N r$ , le nombre de recombinaison par génération

La dérive locale &  
l'isolement reproducteur  
entre populations  
créent de  
l'apparentement



Matrice de données ( $n \times k$ )

- $n$  individus ( $n \ll N$ )
- $k$  polymorphismes

Rappel général sur la sélection

## Données phénotypiques :

1. Simulées par la fonction vs.
2. Mesurées aux champs

Y

A<sub>mar</sub>

X

k

## Données génotypiques :

1. Simulées par la fonction simul coalescent vs.
2. Mesurées en laboratoire par séquençage

Architecture du trait

Traits multigéniques  
Modèle infinitésimal  
Loci à faible effet  
Ex : rendement

Trait monogénique ou faiblement polygénique  
Loci à forts effets  
Ex : Résistance aux maladies

Calcul des GEBVs (RRblup) :

$$Y = Xg + e$$

X = Marqueur

Y = Vecteur des phénotypes corrigés

g = effets des marqueurs,  $\sigma^2_A$ ,  $h^2$ , accuracy

$$GEBV = Xg$$

Prédire individus non phénotypés



Qualité de la valeur phénotypique

Phénotypage d'une partie seulement d'une génération ?  
Phénotypage et génotypage d'une pop d'entraînement très diverse + prédiction sans phénotypage des individus ?

Classement des individus pour les futures croisements :

- Avec la meilleure **GEBV** ?
- Avec quel **QTL / Marqueur** ?
- Selon quels caractères ?

Détection des gènes majeurs (GWAS) :

$$Y = XB + ZU + e$$

Y = vecteur des phénotypes

X = Marqueur (+ mu)

U = Effet polygénique des individus,  $\text{Var}(U)$

A<sub>mark</sub>  $\sigma^2_A$

B = Effets des marqueurs

GWAS

On fait tourner un modèle pour chaque marqueur. Et on regarde les marqueurs dont l'effet est sign selon un seuil

Plus de phénotypage une fois les gènes majeurs trouvés on génotype seulement au marqueur ?  
Mais dans la vraie vie contournement de résistance

1/ Données

2/ Statistique

3/ Sorties

# L'équation du sélectionneur pour gérer les cycles de sélection

$$R_{\text{trait}} = i \cdot r \cdot \sigma_A^2$$

$R_{\text{trait}}$  = gain génétique (en unité de trait)

$i$  = intensité de sélections (quantile de la loi normale)  $\Rightarrow$  Épuisement de la div génétique

$r$  = cor(True breeding value, estimateur de la True Breeding value)  $\Rightarrow$  précision/qualité de la sélection

=  $h$  si on sélectionne sur des données phénotypiques

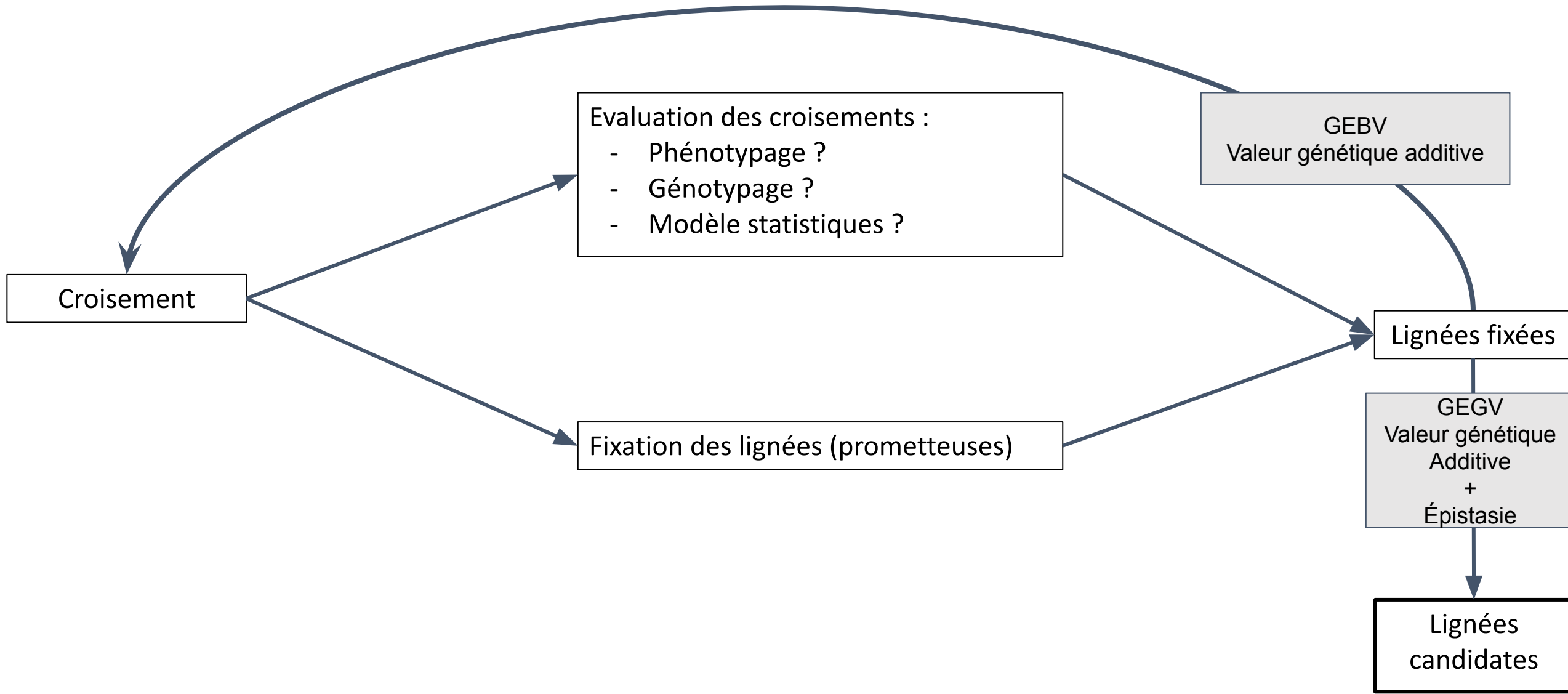
= accuracy de prédiction /  $h$  si on utilise des données prédites avec un modèle de prédiction

génomique

$\sigma_A^2$  = variance génétique disponible pour la sélection  $\Rightarrow$  Potentiel pour la sélection



# Stratégie de sélection



# Pensez-y (Moi j'oublie souvent)

- Mettre les noms des individus dans le bon ordre dans chacune des matrices (on peut utiliser la fonction `match`)
  - Pensez à bien donner à manger les bons objets aux fonctions (attention `data.frame != data.table != table != matrix`)
  - Ordonnez les marqueurs (par chromosome et par position génétique) pour la GWAS
  - Fonction pour exporter une matrice en fichier Excel ou en fichier csv
- ⇒ `write.csv` (data à exporter en csv, file = « Nom.csv »)
- ⇒ `write.xlsx` (data à exporter en xlsx, file = « Nom.xlsx ») # package xlsx