**Organisation de simulations pour évaluer l’utilité d’inclure un modèle animal dans QTLMap**

1. Type de données

Nous retenons dans un premier temps l’idée de créer des jeux de données par simulation. Pour chaque simulation, la population comprend trois époques successives :

* Epoque « historique » de centaines ou milliers de générations au cours de laquelle s’installe un équilibre (dit de Wright Fisher) se caractérisant par des fréquences pour les différents marqueurs et un déséquilibre de liaison.
* Epoque « pedigree » de quelques dizaines au plus de générations au cours desquelles les informations pedigree sont disponibles.
* Epoque expérimentale correspondant au protocole de détection de QTL
1. Constitution des jeux de données

Les simulations comprennent deux étapes.

Etape 1 : fabrication d’un fichier (**pedig**) décrivant l’information généalogique maximum connue sur les individus du protocole de détection de QTL. Ce protocole comprend deux générations (**ng+1** et **ng+2**). Les phénotypes, les filiations et les génotypes aux marqueurs sont connus pour **ng+2**.

Pour simplifier la création et le suivi de ces données, nous supposerons que les générations sont séparées. La population comprendra **ng+2** générations. Les **ng** premières correspondent à la partie « historique » le deux dernières au protocole de détection de QTL.

Dans la partie historique (pour chacune des **ng** premières générations), on a **2** x **naped** animaux nés (**namped** males et **nafped** femelles, avec **naped=namped+nafped**) parmi les lesquels seront retenus **nmped** males et **nfped** femelles reproductrices. Pour simplifier on supposera que chaque femelle donne **ndfped** descendants (**nfped.ndfped=naped**) et chaque male est accouplé à **nfmped** femelles (**nmped . nfmped=nfped**)

Les reproducteurs d’une génération sont choisis au hasard parmi les nés :

* **nmped** males parmi les **namped** (g05nbf dans nag, ou RANPER dans datapac pour permuter les **namped** individus et retenir les **nmped** premier)
* **nfped** femelles parmi les **nafped** (g05nbf ou RANPER)

Les animaux de la génération suivante sont générés à partir de ceux de la génération précédente :

* chaque père (1 à **nmped**) est accouplé à **nfmped** femelles. Une manière simple est
	+ - de faire une permutation au hasard des **nfped** femelles (g05naf ou RANPER)
		- d’affecter les femelles selon cette liste permutée aux différents mâles : **1** à **nfmped** pour le premier mâle, **nfmped +1** à **2. nfmped** pour le second etc….
* chaque femelle donne **ndfped** descendants avec son conjoint. Le sexe de chaque descendant est tiré au hasard (g05mjf ou UNIRAN en prenant une probabilité de ½ et en tirant 1 individu doit marcher)

A la fin on a un pedigree sur **ng** générations

Les deux dernières générations correspondent au protocole de détection de QTL. Soit **nmqtl** pères choisis parmi les **namped** jeunes males de la génération **ng** et **nfqtl=nmqtl.nfmqtl** mères choisies parmi les **nafped** jeunes femelles. Leurs accouplements (1 conjoint/femelle et **ndfqtl** descendants / femelle) donnent un total de **ndescqtl** descendants.

Le fichier pedigree « **pedig** » comprend alors **ng+2** générations. Ce fichier est en entrée de LDSO, conformément à l’exemple 5 donné par la notice de ce logiciel.

Le programme **creat\_pedig.f** sur dga2/utoujme/modanim/ génère **pedig**

En entrée, il faut donner les paramètres dans le fichier **parpop**

Fichier **parpop** est donné en entrée de **creat\_pedig.f**, sous ce nom

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 2 | 2 | 6 | 2 | 4 | 10 | ng,nmped,nfmped,ndfped,nmqtl,nfmqtl,ndfqtl | Nbr generations, (nbr males, nbr femelles/male; nbr descendants/ femelle par génération historique), (nbr males, nbr femelles/male; nbr descendants/ femelle dans le protocole qtl)  |
| 123 |  |  |  |  |  |  | iseed | Racine pour les simulations |

Etape 2 : fabrication à l’aide de LDSO des données marqueurs (**geno**) et caractère quantitative (**pheno**) pour les individus du protocole (**ng+1** et **ng+2**), en cohérence avec l’ensemble de la généalogie (**1** à **ng)** et compte tenu d’un déséquilibre de liaison « historique » chez les fondateurs (génération **1**) crée par la transmission des chromosomes pendant **nhis** générations historiques.

Fichier **general** est donné en entrée de LDSO, sous ce nom

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. | 1 |  | tot\_length, nchrom  | Taille du génome, nombre de chromosomes  |
| 2000 |  |  |  nmq | Nombre de locus |
| 1 |  |  |  nqtl | Nombre de QTL |
| 0 |  |  |  model | Distribution des effets du QTL (1=gamma ; 0=donnés par user) |
| 0  | 0 |  | n\_qtl\_dom, ncpl\_qtl\_interact | Nbr de QTL dominant, de paires en interaction |
| 2 |  |  | store\_cop | Stockage des origines de chaque allèle |
| 0 | 0 | 0 | lect\_map,lect\_gen,further\_use | 0/1 = création/lecture de la carte, de la population, impression de A |
| 0.5 |  |  | eff(1) | effet additif du 1er QTL |
|  |  |  |  |  |
| 2 |  |  | typ\_map | positions des marqueurs et QTL au hasard (1) ou données (2)  |
| 2001 |  |  | max\_loc | nombre maxi de locus (marqueurs +QTL) sur 1 chromosome |
| 2001 |  |  | nloc(1) | nombre de locus sur le 1er chromosome |
| 0.000 | 0 |  | pos\_loc(1,1), typ\_loc(1,1) | Position, type (1=QTL ; 0=marqueur) du locus 1 |
| 0.005 | 0 |  | pos\_loc(1,2), typ\_loc(1,2) | Position, type (1=QTL ; 0=marqueur) du locus 2 |
| ………… |  |  |  |  |
| 0.500 | 1 |  | pos\_loc(1,1001), typ\_loc(1,1001) | Position, type (1=QTL ; 0=marqueur) du locus 1001 (le QTL) |
| ………… |  |  |  |  |
| 1.000 | 0 |  | pos\_loc(1,2001), typ\_loc(1,2001) | Position, type (1=QTL ; 0=marqueur) du locus 2001 |
|  |  |  |  |  |
| 1 |  |  | typ\_mk | type de marqueur (1=SNP) |
| 0.0001 | 0.0001 | 0.0001 | tmutSNP,tmutMST,tmutQTL | Taux de mutations |
| 0 |  |  | deseq | Type de DL initial (1 une seule mutation causale) |
| 0 |  |  | typing\_err | Erreur de génotypage possible (1) |

Fichier **pop1** est donné en entrée de LDSO, sous ce nom

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 |  |  |  | nbpop | Nombre de populations |
| 0 |  |  |  | nb\_bott | Nombre de bottleneck |
| 100 | 100 | 1. | 1. | n1(1),t1(1),sel(0,1,1:2) | Nbr fondateurs, nbr générations, taux sélection par sexe dans la 1er population |
| 0 |  |  |  | qtl\_mut(1) | Nbr de QTL initialement fixé dans la 1er population |
| 1 |  |  |  | mut(1) | Pour biallelique (typ\_mk=1), mutation des SNP possible si mut(1)=1 |
| ……… |  |  |  |  |  |
| 1 |  |  |  | mut(2000) | Pour biallelique (typ\_mk=1), mutation des SNP possible si mut(2000)=1 |
| 2 | 1 |  |  | nball(nmq+1),mut(nmq+1) | Nombre allèles au 1er QTL, mutation possible |
| 0.3 |  |  |  | h2(1) | Héritabilité dans la 1er population |
| 0.5 |  |  |  | ratio\_polQTL | Ratio variances polygenique / additive  |
| 0 |  |  |  | all\_frq | Fréquences alléliques 0=égale,1=données,3=uniforme |
| 0 |  |  |  | direct\_fin | 0=simulation d’une population historique |
| 1 |  |  |  | old\_ped(1) | 1=simulation selon un pedigree après l’historique |
| pedig | XXX |  |  | fic\_ped(1),ind\_ped(1) | Nom du fichier donnant le pedigree de la 1er population, taille du pedigree |
| YYY |  |  |  | last\_ind | Nombre du dernier individu dans pedigree |

Fichier **popfin** est donné en entrée de LDSO, sous ce nom

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 3 |  |  |  | typ\_pop | 1= ggd, 2=BC, 3=pedigree réel,4=full half sib  |

Fichier **files** est donné en entrée de LDSO, sous ce nom

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2 | 1 |  |  | max\_gener\_eval,outfiles | Nbr maxi de générations avec stat en sorties,0 = pas de sorties  |
| n |  |  |  | f\_dl,j,gener\_eval(1,1:j) | Y (yes) si sortie du fichier de tous les DL, si y : nbr et noms des générations |
| n |  |  |  | f\_dlq,j,gener\_eval(10,1:j) | Y (yes) si sortie du fichier des DL avec le QTL, si y : nbr et noms des générations |
| n |  |  |  | f\_dlsum,j,gener\_eval(8,1:j) | Y (yes) si sortie du fichier résumés des DL, si y : nbr et noms des générations |
| n |  |  |  | f\_ibd,j,gener\_eval(2,1:j) | Y (yes) si sortie du fichier longueurs des segments ancestraux, si y : nbr et noms des générations |
| n |  |  |  | f\_ibdq,j,gener\_eval(3,1:j) | Y (yes) si sortie du fichier longueurs des segments ancestraux avec les QTLs, si y : nbr et noms des générations |
| n |  |  |  | f\_csg,j,gener\_eval(4,1:j) | Y (yes) si sortie du fichier consanguinités moyennes aux locus, si y : nbr et noms des générations |
| n |  |  |  | f\_hPIC,j,gener\_eval(5,1:j) | Y (yes) si sortie du fichier des PIC du QTL, si y : nbr et noms des générations |
| n |  |  |  | f\_fqall,j,gener\_eval(6,1:j) | Y (yes) si sortie du fichier des fréquences alléliques aux loci, si y : nbr et noms des générations |
| n |  |  |  | f\_fqcop,j,gener\_eval(7,1:j) | Y (yes) si sortie du fichier des fréquences des id ancestrux à tous les loci, si y : nbr et noms des générations |

Fichier **param** est donné en entrée de LDSO, sous ce nom

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 123 | 581 | 1547 |  | Seed1,seed2,seed3 | Les racines pour les simulations |

En sortie de LDSO, on a deux fichiers intéressants

* **SimhaploNoQTL** qui comprend **2(nmqtl+nfqtl+ndescqtl)** lignes. Chaque individu (les **nmqtl** pères puis les **nfqtl** mères puis les **ndescqtl** descendants du protocole QTL) a deux lignes correspondant à ses deux chromosomes homologues (pour **nchrom=1**) : les chromosomes sont « phasés » (on oubliera cette information). La première ligne (1er chromosome comprend l’ID de l’individu puis, pour chaque marqueur, l’allèle présent). Idem pour le deuxième ligne.

ID(i) m(i,1,1) m(i,2,1) m(i,3,1) … m(i,M,1)

ID(i) m(i,1,2) m(i,2,2) m(i,3,2) … m(i,M,2)

Le ficher génotype de QTLMap se construit à partir de cette information :

ID(i) m(i,1,1) m(i,1,2) m(i,2,1) m(i,2,2) … m(i,M,1) m(i,M,2)

* **Simperf** qui comprend **(nmqtl+nfqtl+ndescqtl)** lignes. Pour chaque individu (les **nmqtl** pères puis les **nfqtl** mères puis les **ndescqtl** descendants du protocole QTL) on a son ID, sa performance et sa valeur génétique. Le fichier performance de QTLmap se construit à partir des **ndescqtl** dernières lignes et 2 premières colonnes.

Le fichier carte de QTLMap sera celui d’une carte de **nmq** (dans l’exemple 2000) marqueurs équidistant sur 1 Morgan. Donc dans l’exemple, sachant que le QTL prend une 2001ème position (ici 0.500) :

marqueur1 0.000 0.000 0.000 1

marqueur2 0.005 0.005 0.005 1

…

marqueur500 0.495 0.495 0.495 1

marqueur501 0.505 0.505 0.505 1

…

marqueur2000 1.000 1.000 1.000 1

On devrait peut être éviter ce trou de marquage à 0.500, en positionnant le QTL par exemple à 0.4955.

Le fichier pedigree de QTL est **pedig** produit par **creat\_pedig.f**

1. Cas étudiés

On va explorer diverses configurations, définis par exemple par les paramètres tels que l’héritabilité **h2(1)** ou le nombre de descendants **ndescqtl**. Pour chaque configuration on fera **nsim** simulations.

Les simulations doivent servir à établir les propriétés de QTLMap avec modèle animal par rapport au QTLMap classique. Ces propriétés concernent

* la puissance, mesurée par le pourcentage de détections, c'est-à-dire de LRT > seuil parmi les **nsim** simulations.
* la précision, mesurée par la « Mean Squared Error » , MSE où moyenne des carrés des écarts entre la vraie position du QTL , donnée par le **pos\_loc** correspondant au **typ\_loc** = 1 du fichier **general** et la position estimée, correspondant au LRTmax de QTLMap
* la robustesse, mesurée par le pourcentage de fausses détections, c'est-à-dire de LRT > seuil parmi les **nsim** simulations réalisées sans effet QTL (**eff(1)=0.**), la question étant posée si l’utilisateur donne une valeur fausse de l’héritabilité.

Les configurations à étudier pourraient être :

 Pour quelques (choix à voir en fonction des résultats) protocole définis par **nmqtl**, **nfqtl** et **ndescqtl**, influence du nombre de générations prises en compte dans le pedigree (0,2,5,10,20 ?), et de l’héritabilité, oui sans doute plutôt d’un rapport entre effet du QTL et héritabilité.

 Pour un choix de pedigree, influence du type de protocole défini par **nmqtl**, **nfqtl** et **ndescqtl**

 Pour quelques protocoles et pedigree, impact d’une erreur sur l’héritabilité en termes de puissance et précision (sous H1), robustesse sous H0.

Il faudra avant de lancer le gros des simulations caler le paramètrage de la population historique (nombre de fondateurs, de générations, mutation..) Le plus simple serait de reprendre les choix des autres. Par exemple dans Meuwissen et Goddard (2000) 100 générations, 100 fondateurs. Cette étude exploratoire doit rendre compte du DL (distribution du r2 en fonction de la distance), des fréquences aux SNP (nombre de marqueurs fixés, nombre de marqueurs avec une MAF < 0.10) et au QTL.