

NSI

PROJET - CRÉER UNE ARMÉE DE RATS MUTANTS

1 Situation

Vous êtes à la tête d'une équipe de chercheurs renommés. Généticiens, informaticiens, biologistes, statisticiens et zoologistes se bousculent pour travailler avec vous et votre mascotte Schapira - *une bullmastiff de 50kg nommée d'après Georges, votre mentor*. Il faut dire que votre dernière expérience a été couronnée de succès : vous avez réussi à modifier génétiquement des lions et des aigles pour créer des hippogriffes domestiques, révolutionnant ainsi le monde du transport écologique.

Ce dernier projet vous a donc apporté une réserve financière confortable, et vous avez tout le loisir de trouver une nouvelle idée... Ce n'est pas dans vos habitudes mais cette fois-ci, vous restez pourtant songeur et indécis. Comment faire mieux ?

Assis à votre bureau, vous regardez longuement vos notes, cherchez votre chien du regard... Schapira est en train de jouer avec un des rats du laboratoire. C'est la révélation : des rats ! Des rats géants ! Votre nouveau projet consistera à créer une armée de rats de la taille de Schapira à partir de la population de rats bruns déjà présente dans vos locaux scientifiques. *Pourquoi ?* A-t-on vraiment besoin d'une raison ? *Oui ?* Vous et votre équipe la trouverez sans doute plus tard.

Pour vous donner une idée de l'ampleur de la tâche, voici une image de deux de vos collaborateurs, qui regardent Schapira jouer avec son rat.



2 Méthode

Avant de vous embarquer sans réfléchir dans cette entreprise périlleuse - *vous savez apprendre de vos erreurs passées, et les griffures d'aigles sur vos bras n'ont pas encore fini de cicatriser* - vous décidez qu'il est plus prudent de simuler les résultats en Python.

Pour cela, vous allez utiliser ce qu'on appelle un **algorithme génétique**. Votre but est de produire une population de rats pesant en moyenne 50kg à partir d'une population de rats bruns pesant beaucoup (beaucoup) moins.

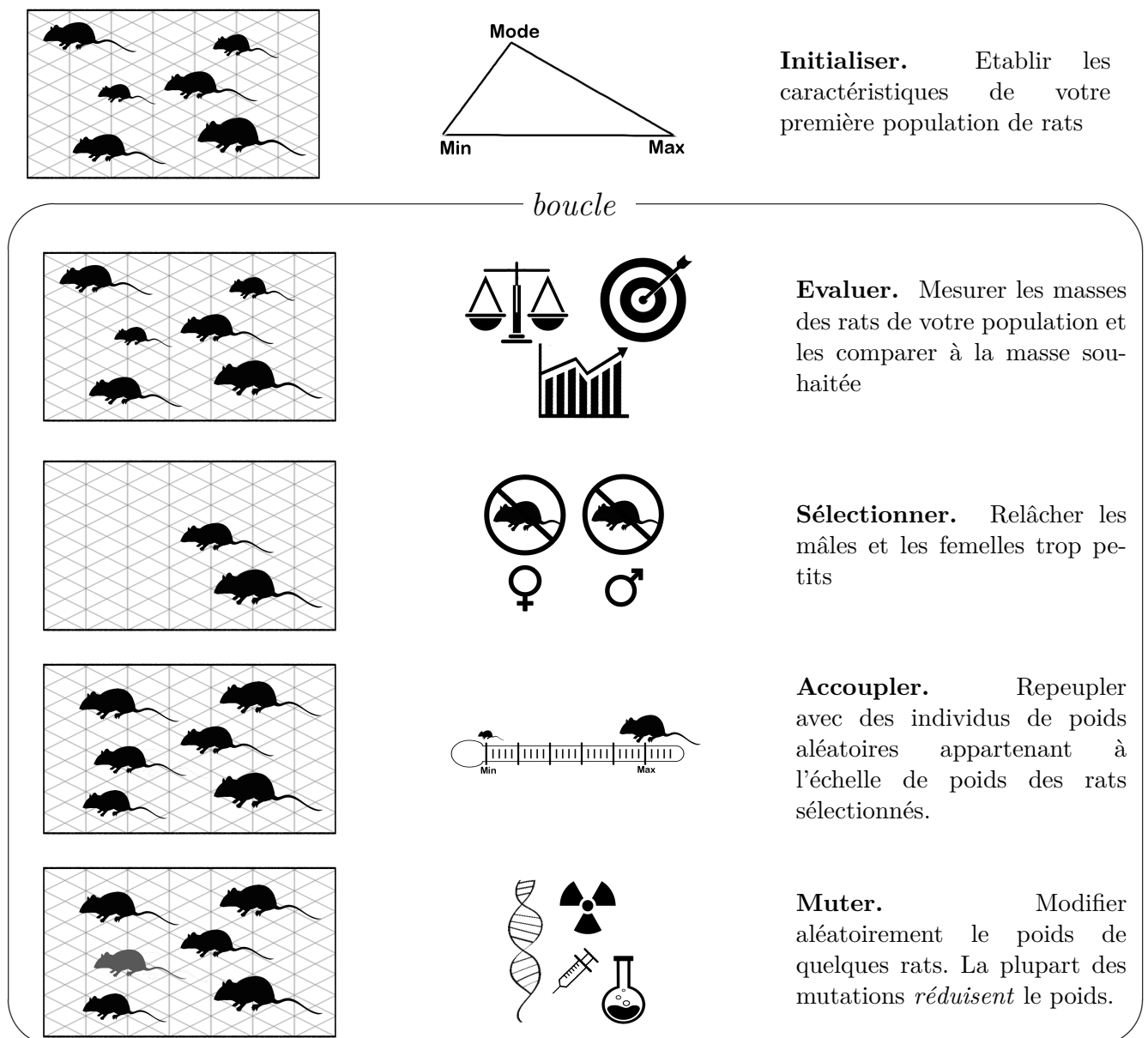
Au fur et à mesure des croisements et des mutations, chaque populations (ou *génération*) de rats représente une possible solution à votre problème. Comme tout bon éleveur qui se respecte, vous relâchez les mâles et les femelles trop petits. Un zoo des horreurs a gentiment proposé de s'en occuper. *Mais si vous souhaitez nourrir vos derniers hippogriffes avec, libre à vous.*

Vous faites ensuite se reproduire les rats restants entre eux. Vous effectuez donc une opération essentielle en programmation génétique : des *croisements*.

La progéniture de ces rats sera globalement de la même taille que leurs parents, vous devez donc effectuer des mutations génétiques sur quelques uns d'entre eux. Cependant, les mutations sont rares et les résultats sont souvent neutres ou non bénéfiques. Dans notre cas, cela signifie que la plupart des mutations ne feront pas varier la masse des rats, et la fera même baisser. Dans de rares cas, vous obtiendrez cependant des rats plus gros !

Vous répétez ce processus jusqu'à obtenir une population de rats dont le poids moyen correspond à votre objectif, ou que vous n'en puissiez plus des rats - *admettons le, ils sont biens moins mignons que des gros chats.*

Cet algorithme est résumé par la figure suivante :



3 Données

Pour que votre simulation informatique soit correcte, vous allez avoir besoin de quelques données statistiques. Vous savez déjà que la masse moyenne d'une femelle bullmastiff est de 50 000g, mais votre connaissance des rats est médiocre. Après quelques recherches, vous trouvez les données suivantes :

Paramètres	Valeurs
Masse minimale	200g
Masse moyenne (femelle)	250g
Masse moyenne (mâle)	300 – 350g
Masse maximale	600g
Nombre de petits par portée	8 – 12
Nombre de portée par an	4 – 13
Durée de vie (à l'état sauvage, en captivité)	1 – 3, 4 – 6

Les rats en captivité sont souvent mieux nourris que les rats sauvages, et pèsent alors plus lourds, s'accouplent plus et ont plus de petits. Vous pouvez donc choisir les valeurs maximales quand c'est possible. Pour ce projet, vous utiliserez les variables suivantes :

Variables et valeurs	Remarques
BUT=50000	Cible à atteindre : on veut une masse moyenne de 50kg
NB_RATS=20	Nombre total de rats <i>adultes</i> que votre laboratoire peut accueillir
INIT_MIN=200	Masse minimum d'un rat de la population initiale, en gramme
INIT_MAX=600	Masse maximum d'un rat de la population initiale, en gramme
INIT_MODE=300	Masse la plus courante pour rat de la population initiale, en gramme
PROBA_MUT=0.01	Probabilité qu'un rat mute
MIN_MUT=0.5	Dans le pire des cas, une mutation multiplie la masse du rat par 0.5
MAX_MUT=1.2	Dans le meilleur des cas, une mutation multiplie la masse par 1.2
NB_PETITS=8	Nombre de petits pour l'accouplement d'un couple de rats
NB_PORTEES=10	Nombre de portées par an par couple de rat
LIM_GENERATION=500	Le programme s'arrête automatiquement après 500 générations

Les rats se reproduisent à une fréquence très rapide, vous n'aurez donc pas à vous préoccuper de leur durée de vie. Même si vous gardez quelques parents d'une génération précédente, ils seront très rapidement remplacés au fur et à mesure que le poids moyen de leurs progénitures augmente.

4 Remarques

- L'algorithme génétique présenté ici devra être codé en python et retournera le nombre de générations nécessaire pour attendre la masse cible de 50kg. Avec les données du tableau précédent, votre algorithme ne devrait pas mettre plus de 5s à fournir un résultat.

- Il est bien sûr nécessaire de tester plusieurs fois votre algorithme et de conserver les résultats afin de pouvoir faire une étude statistique et en tirer des conclusions importantes.

Comme par exemple le temps moyen nécessaire pour obtenir des rats qui puissent servir de vrais compagnons de jeu à Schapira, mais à vous de déterminer ce qui vous semble important. C'est vous le scientifique, après tout.

- Une **analyse de sensibilité** s'effectue en changeant de nombreuses fois la valeur d'un seul paramètre et en observant les résultats obtenus. C'est une étape importante de l'analyse d'un algorithme génétique. Il vous faudra cependant être prudent : certaines variables dépendent les unes des autres. De plus, comme les résultats sont aléatoires, vous devrez tester de nombreuses fois votre algorithme pour chaque valeur du paramètre.

Les deux choses que vous pouvez contrôler dans votre programme sont :

- le nombre de rats (NB_RATS). *Ne pas occuper toutes les cages ou en acheter de nouvelles, cela reste à votre portée.*
- la probabilité d'apparition d'une mutation chez un rat (PROBA_MUT). Il vous faudra pour cela modifier le régime alimentaire des rats ou encore leur temps d'exposition aux radiations. *A vous de voir. Au risque de se répéter, c'est vous le chef de projet.*

En effectuant une analyse de sensibilité de votre algorithme pour ces variables, vous pourrez donc juger de leur impact sur le temps nécessaire à l'accomplissement de votre projet.

- Des graphiques sont bien entendu les bienvenus pour illustrer vos résultats.
Vous conviendrez que lire de longues explications sans dessins, c'est beaucoup moins intéressant.