

Détail du projet : évolution d'une population de poissons clowns

1 Généralités

Les poissons clowns vivent de manière sédentaire dans des anémones de mer. Chaque anémone est habitée par une famille constituée d'une femelle, d'un mâle et de juvéniles. Il existe une relation de symbiose entre les animaux : l'anémone protège la famille qu'elle héberge et celle-ci défend son hôte contre les poissons coralivores. Un poisson clown ne peut pas subsister en dehors d'une anémone, et réciproquement une anémone ne peut subsister si elle n'est pas habitée par une famille de poissons clowns.

2 Cycle de vie des individus

La vie du poisson clown est structurée en stade et non en âge. Arrivant à l'état de larve dans une anémone, il ne se développe en individu juvénile que s'il existe une place disponible. Lorsque la femelle meurt, le mâle grossit et change de sexe pour devenir femelle. Le plus âgés des juvéniles se développe alors pour devenir un mâle sexuellement mûre. Cette dernière modification se produit aussi lors de la mort du mâle, sans que la femelle soit nécessairement morte. Lorsqu'un juvénile meurt ou se transforme en mâle, il se crée une place disponible dans l'anémone pour laisser croître une nouvelle recrue (larve). On supposera, en première approximation, que les changements se font instantanément.

3 Topographie

Les anémones sont regroupées au sein n_L lagons. Les larves issues d'oeufs pondus au pied d'une anémone par la femelle peuvent voyager jusqu'à une autre anémone. On parle d'auto-recrutement si l'anémone appartient au même lagon, ou de recrutement dans le cas contraire. Enfin, si des larves arrivent d'une autre île par le plancton pélagique, on parle d'immigration. Lorsque l'afflux de nouvelles larves, soit par autorecrutement, soit par recrutement, soit par immigration, est insuffisant pour compenser les pertes par mortalité, la population de l'anémone diminue pour atteindre 0. À ce moment, l'anémone meurt. Il faut noter que si la famille se trouve réduite à une femelle, celle-ci devient infertile et ne participera pas au recrutement des autres anémones.

4 Données

L'utilisateur biologiste dispose d'un fichier de données composé de n_A lignes. Chaque ligne contient les données initiales relatives à une anémone, à savoir

- son identifiant,
- l'identifiant du lagon auquel elle appartient,
- sa capacité d'accueil,
- sa population réelle au début de la simulation.

5 Paramètres biologiques

Un des objectifs du biologiste est d'évaluer *in silico* l'impact de certains paramètres sur la persistance de la population. Il doit donc pouvoir spécifier, soit par un fichier de paramètre, soit par une saisie dans l'interface graphique les paramètres spécifiques à chaque lagon :

- le taux de mortalité μ_F (resp. μ_M, μ_J) des femelles (resp. des mâles, des juvéniles),
- le taux de recrutement par femelle, pour chaque lagon $\theta_1, \dots, \theta_{n_L}$,
- le taux d'immigration dans chaque lagon, supposé constant au cours du temps λ .

On pourra prendre comme jeu de paramètres typiques les chiffres suivant :

		Lagon	μ_F	μ_M	μ_J		
		1	0.184	0.164	0.187		
		2	0.22	0.095	0.159		
		3	0.168	0.203	0.184		
		4	0.219	0.25	0.15		
		5	0.174	0.166	0.191		
		6	0.203	0.243	0.203		
		7	0.267	0.243	0.264		
Lagon	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7
1	0.451	0.236	0.09	0.081	0.163	0.135	0.065
2	0.138	0.108	0.087	0.026	0.054	0.13	0.126
3	0.076	0.055	0.173	0.044	0.008	0.	0.094
4	0.086	0.058	0.115	0.069	0.059	0.086	0.146
5	0.266	0.208	0.167	0.25	0.405	0.438	0.03
6	0.012	0.016	0.	0.019	0.019	0.093	0.023
7	0.109	0.067	0.149	0.107	0.102	0.	0.13

Dans le tableau ci-dessus, chaque chiffre est à interpréter comme un taux de fertilité d'une femelle du lagon colonne, à destination du lagon ligne. Par exemple, sur la première ligne, on lit que chaque femelle du lagon 2 génère un flux de 0.236 larve par an dans le lagon 1.

Lagon	1	2	3	4	5	6	7
λ	33.207	28.577	11.097	29.676	46.026	4.472	31.633

Dans ce dernier tableau, on lit que le lagon 2 reçoit un flux de 28.577 larves par an par immigration.

6 Algorithme

L'algorithme d'évolution a la structure suivante :

```

initLagons()
initAnemones()
initParametresSimu()
initHorloge()
 $T \leftarrow 0$ 
while  $T < T_{\max}$  do
```

```

 $\lambda_{\text{global}} \leftarrow \text{tauxGlobal}(\text{horloge})$ 
 $T \leftarrow T + \text{tireExp}(\lambda_{\text{global}})$ 
 $e \leftarrow \text{tireEvent}(\text{horloge})$ 
 $\text{appliqueEvent}(e)$ 
 $\text{miseAJour}(\text{horloge})$ 
end while

```

L'horloge peut être vue comme un lieu de stockage, dans lequel on enregistre les taux d'occurrence de tous les événements possibles. Pour une anémone a , on a les $n_L + 4$ événements possibles

- mort de la femelle, du mâle, d'un juvénile, notés respectivement $e_F(a)$, $e_M(a)$ et $e_J(a)$;
- arrivée d'une larve par recrutement ou autorecruitment, noté $e_L(a, l)$, où l est le lagon d'origine de la larve. On a donc n_L événements de ce type pour chaque anémone;
- arrivée d'une larve par immigration, noté $e_I(a)$.

À chaque instant, chacun des $n_A (n_L + 4)$ événements ainsi définis peuvent se produire. On peut maintenant détailler la procédure `initHorloge` :

```

for  $a = 1 \dots n_A$  do
   $l_a \leftarrow \text{lagon}(a)$ 
   $\text{horloge}(a, \text{mortFemelle}) \leftarrow \text{tauxMortFemelle}(l_a)$ 
   $\text{horloge}(a, \text{mortMale}) \leftarrow \text{tauxMortMale}(l_a)$ 
   $\text{horloge}(a, \text{mortJuvenile}) \leftarrow \text{tauxMortJuvenile}(l_a) * \text{nombreJuvenile}(a)$ 
  for  $l = 1 \dots n_L$  do
     $\text{horloge}(a, \text{arriveeLarve}(l)) \leftarrow \text{tauxRec}(l, l_a) * \text{nombreFemelle}(l)$ 
  end for
   $\text{horloge}(a, \text{imig}) \leftarrow \text{tauxImig}(l_a)$ 
end for

```

Le taux global, renvoyé par `tauxGlobal(horloge)` est la somme des taux de tous les événements. Cependant, on peut distinguer un taux par anémone a , qui serait renvoyé par `tauxAnemone(horloge, a)`. On a donc

$$\text{tauxGlobal}(\text{horloge}) = \sum_{a=1}^{n_a} \text{tauxAnemone}(\text{horloge}, a) .$$

Ainsi, on peut décomposer `tireEvent(horloge)` qui doit renvoyer un événement en deux étapes. D'abord on tire aléatoirement l'anémone a concernée par l'événement en exécutant

$$a \leftarrow \text{tireAnemone}(\text{horloge}).$$

Ce tirage est fait selon les proportions données par les `tauxAnemones`. Ensuite, pour cette anémone, on tire aléatoirement l'événement qui va la concerner en exécutant `tireEvent(a)`. Ce dernier tirage se fait entre les $n_L + 4$ événements définis pour chaque anémone. Si l'on assimile l'horloge à un tableau où chaque ligne correspond à une anémone et chaque colonne à un événement sur cette anémone, alors les `tauxAnemone(horloge, a)`, représentent les sommes pour chaque ligne. On peut donc décomposer le tirage de l'événement en sélectionnant d'abord la ligne comme précédemment, puis la colonne selon les proportions données par cette ligne.

En sortie de `tireEvent(horloge)`, on dispose d'un événement e concernant une anémone a . L'exécution de `appliqueEvent(e)` consiste en

- s’il s’agit de la mort de la femelle, du mâle ou d’un juvénile, on met à jour la population de l’anémone selon les règles biologiques définies plus haut.
- S’il s’agit de l’arrivée d’une larve, on ajoute un juvénile à l’anémone seulement si la capacité d’accueil de celle-ci n’est pas atteinte. Dans la cas contraire, la larve n’est pas autorisée à devenir juvénile. Elle est jetée dehors pour être mangée par un crabe.

Enfin, on doit mettre à jour l’horloge par `miseAJour(horloge)` en fonction de la nouvelle population. Cela n’arrive que s’il y a rupture de couple dans une anémone, c’est-à-dire mort d’une femelle ou d’un mâle sans juvénile en réserve. Dans ce cas on modifie le `initHorloge` décrit ci-dessus.