

BIO2045 – Séance 3

Génétique des populations

Contenu

Concepts principaux	1
Fonctions anonymes	1
Mapping et slicing	2
Filtres	2
Changement de dimensions	2
Simulation: maintien du polymorphisme	2
Suggestions pour le premier devoir	4
Paysage circulaire	4
Sélection génétique	4
Déséquilibre de liaison	4
Traits quantitatifs	4

Concepts principaux**Fonctions anonymes**

Dans la séance précédente, nous avons commencé à écrire des fonctions. Dans certaines situations, il est nécessaire d'utiliser des fonctions, mais puisqu'elles sont très simples et à usage unique, on dispose d'une syntaxe simplifiée, **argument** → **operation**:

```
x → 2*x + 1
```

```
#2 (generic function with 1 method)
```

Le symbole → est simplement - puis >.

Ces fonctions peuvent s'utiliser comme des fonctions régulières:

```
(x → 2*x + 1)(3)
```

7

Mapping et slicing

On peut automatiser certaines opérations *via* la fonction `map`:

```
import Random
Random.seed!(2045)
x = rand(1:5, 10)
```

```
10-element Vector{Int64}:
 2
 5
 1
 2
 4
 5
 4
 4
 4
 3
```

```
map(v -> sqrt(v), x)
```

```
10-element Vector{Float64}:
 1.4142135623730951
 2.23606797749979
 1.0
 1.4142135623730951
 2.0
 2.23606797749979
 2.0
 2.0
 2.0
 1.7320508075688772
```

Filtres

Changement de dimensions

Simulation: maintien du polymorphisme

```
using CairoMakie
CairoMakie.activate!(px_per_unit=2.0)

import StatsBase
```

```

cells = 100
generations = 501
mutation = 1e-4
parents_distance = 3

```

3

État initial

```

lattice = zeros(Bool, (cells, generations, 3))
for i in Base.OneTo(cells)
    lattice[i,1,:] = rand(Bool, 3)
end

```

Première génération

```

for generation in 2:generations
    for i in Base.OneTo(cells)
        parents_possibles = filter(p → 1 ≤ p ≤ cells, (i-
parents_distance):(i+parents_distance))
        parents = StatsBase.sample(parents_possibles, 2,
replace=false)
        for gene in 1:3
            lattice[i,generation,gene] =
lattice[rand(parents),generation-1,gene]
            if rand() ≤ mutation
                lattice[i,generation,gene] = !
lattice[i,generation,gene]
            end
        end
    end
end
end

```

Fonction pour les couleurs

```

colormap = dropdims(mapsllices(x → CairoMakie.Colors.RGB(x...),
lattice, dims=3), dims=3);

```

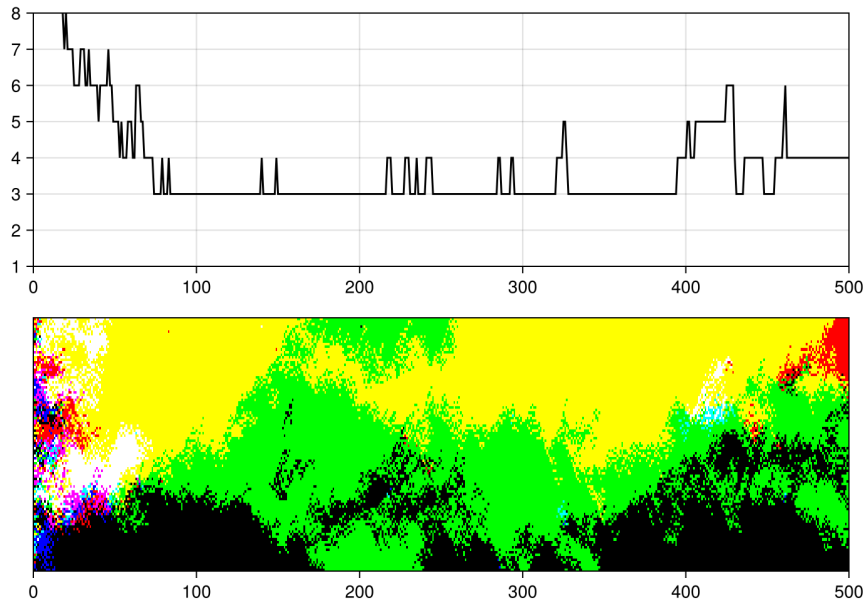
Heatmap et diversité

```

f = Figure(; size=(700, 500))
ax = Axis(f[2,1])
heatmap!(ax, 0:(generations-1), 1:cells, permutedims(colormap))
hideydecorations!(ax)
plax = Axis(f[1,1])
lines!(plax, 0:(generations-1), vec(mapsllices(x →
length(unique(x)), colormap, dims=1)), color=:black)
ylims!(plax, 1, 8)

```

```
xlims!(plax, 0, generations-1)
xlims!(ax, 0, generations-1)
f
```



Suggestions pour le premier devoir

Paysage circulaire

Sélection génétique

Déséquilibre de liaison

Traits quantitatifs