

# Algorithme génétique

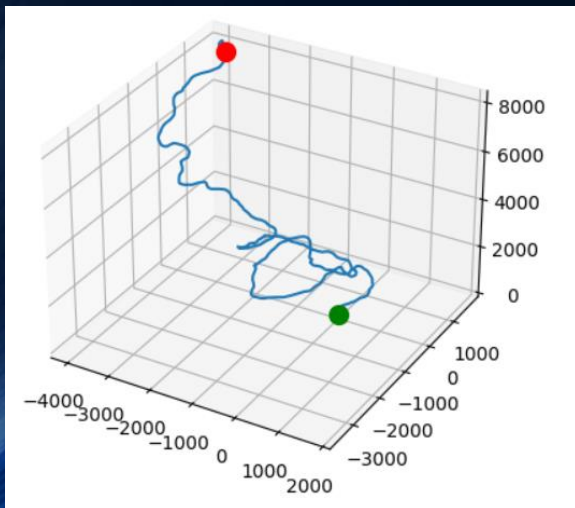
GROUPE 3  
PERZO  
LIOU  
GRANDCHAMP  
COLLEVILLE



CentraleSupélec

# Vous avez dit Algorithme Génétique ?

- Biomimétisme
- Survie d'un individu : Optimisation de la fonction de survie
- Population = {individus}, un individu est une solution potentielle à notre problème
- Notre problème ? Brin d'ADN et une table de correspondance dinucléotide $\Leftrightarrow$  angles . Un angle a un domaine définition.



```
ORIGINAL_ROT_TABLE = {  
  "AA": [35.62, 7.2, -154, 0.06, 0.6, 0],  
  "AC": [34.4, 1.1, 143, 1.3, 5, 0],  
  "AG": [27.7, 8.4, 2, 1.5, 3, 0],  
  "AT": [31.5, 2.6, 0, 1.1, 2, 0],  
  "CC": [33.67, 2.1, -57, 0.07, 2.1, 0],  
  "CG": [29.8, 6.7, 0, 1.1, 1.5, 0],  
  "GC": [40, 5, 180, 1.2, 1.275, 0],  
  "TA": [36, 0.9, 0, 1.1, 2, 0],  
  "TC": [36.9, 5.3, -120, 0.9, 6, 0],  
  "TG": [34.5, 3.5, 64, 0.9, 34, 0],  
}
```

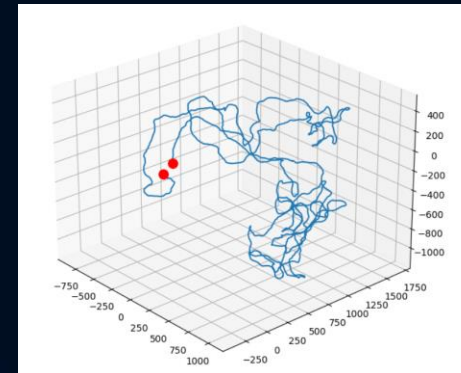
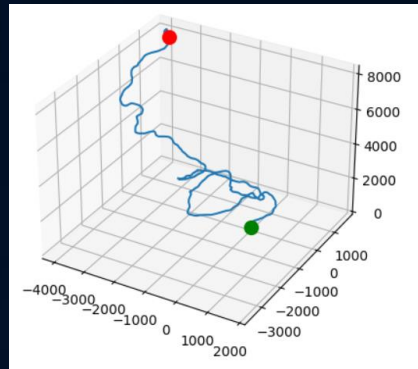
Pour que



```
"AA": ?,  
"AC": ?,  
"AG": ?,  
"AT": ?,  
"CC": ?,  
"CG": ?,  
"GC": ?,  
"TA": ?,  
"TC": ?,  
"TG": ?,
```

# Description globale du produit

- Quoi ? Algorithme d'**optimisation** d'une table de correspondance de chaîne dinucléotide-angle à des fins de cyclisation de représentation 3D d'ADN.
- Pourquoi ? Représenter un **brin d'ADN en 3D avec une fermeture de chaîne**

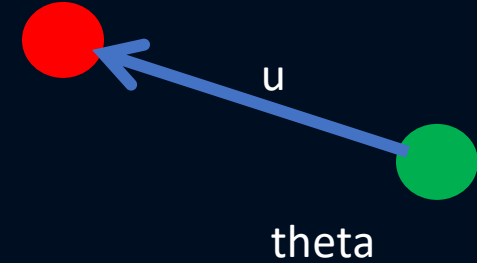


- Comment ? Mise en œuvre d'un **algorithme génétique** pour pallier le temps de recherche de solution exponentiel.

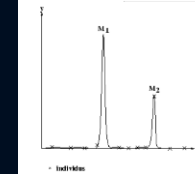
# Algorithme Génétique

Génération aléatoire dans le respect des bornes

$$1 - |\cos \theta| + \frac{||\vec{u}||}{d}$$



- Tournoi
  - Elitisme
- Avec ou sans Scaling



Croisement des valeurs d'angles entre deux tables en conservant les correspondances

Ajout d'un bruit gaussien dans le respect des bornes

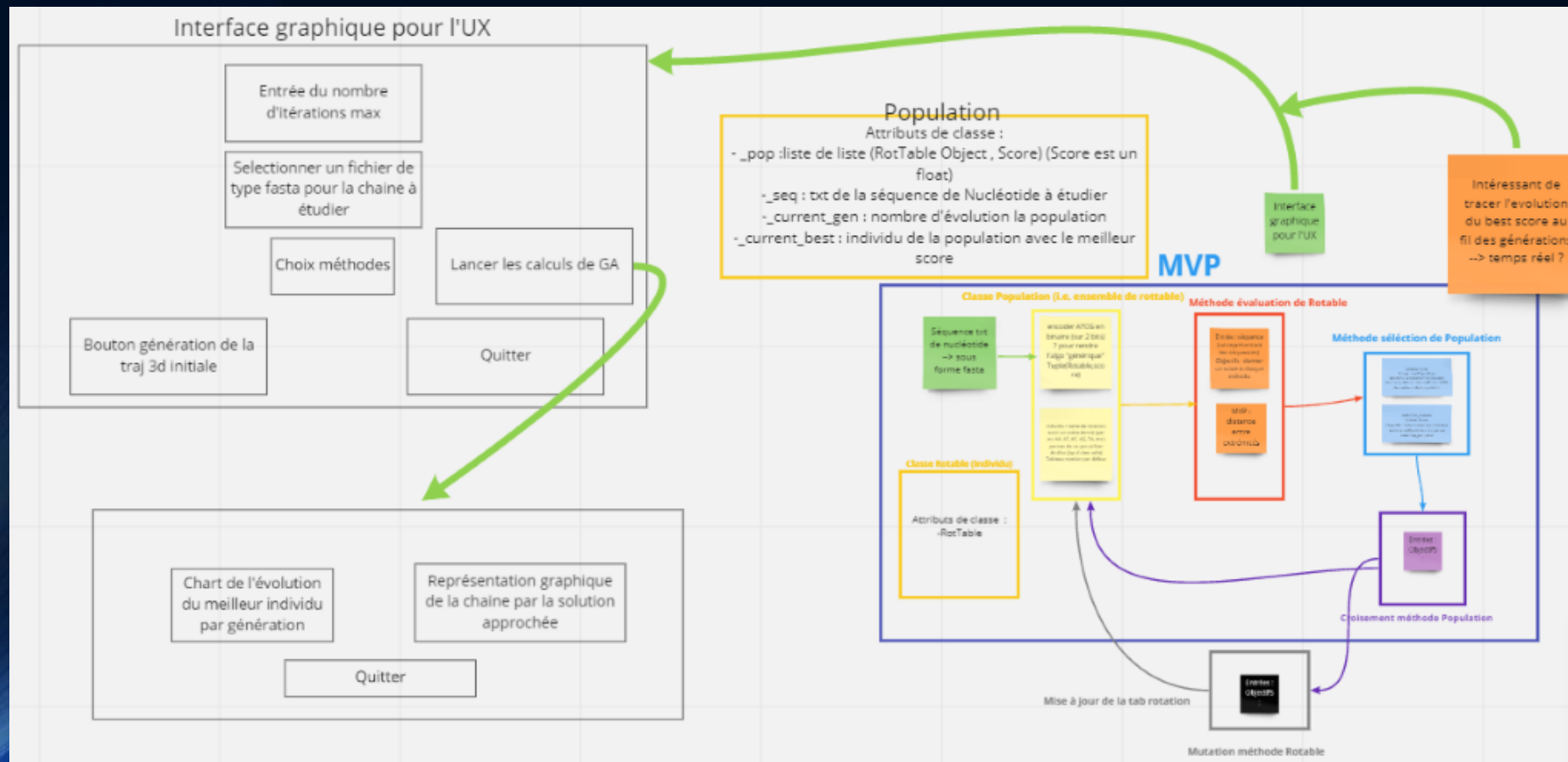
Nombre de génération





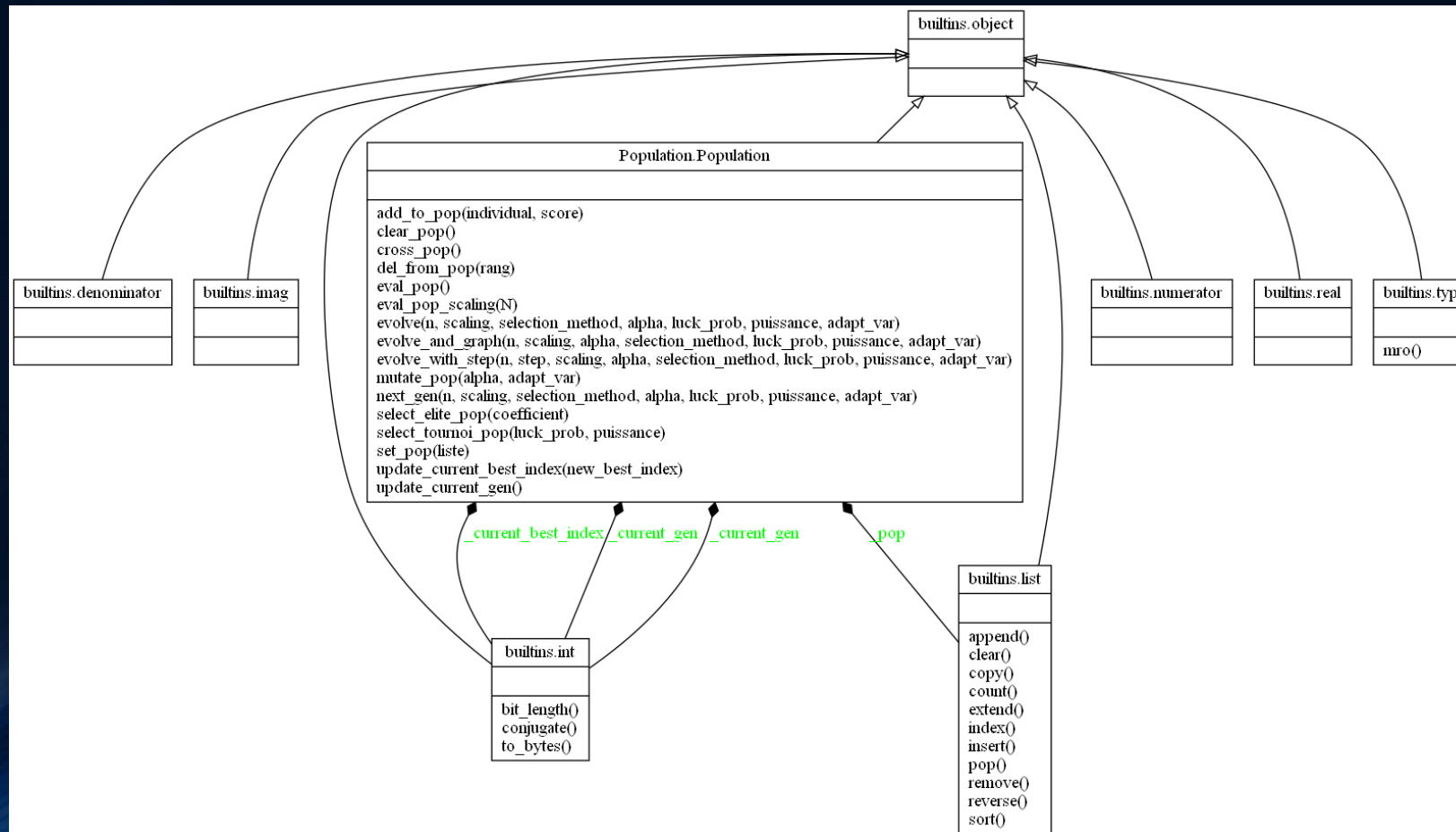
# Qualité du code : Structuration

- Arbre de travail et structure + POO



# Qualité du code : Structuration

- Arbre de travail et structure + POO



# Qualité du code : Couverture & tests

Coverage report: 85%

<i>Module</i> ↑	<i>statements</i>	<i>missing</i>	<i>excluded</i>	<i>coverage</i>
Population.py	206	42	0	80%
RotTable.py	122	26	0	79%
Traj3D.py	35	10	0	71%
test_Population.py	108	0	0	100%
test_RotTable.py	44	0	0	100%
<b>Total</b>	<b>515</b>	<b>78</b>	<b>0</b>	<b>85%</b>



# Qualité du code : Documentation

- ReadMe.md
- Manuel utilisation
- En tête et format fixés des méthodes et fonctions
- Appui sur la norme pep8 afin de formater le code correctement



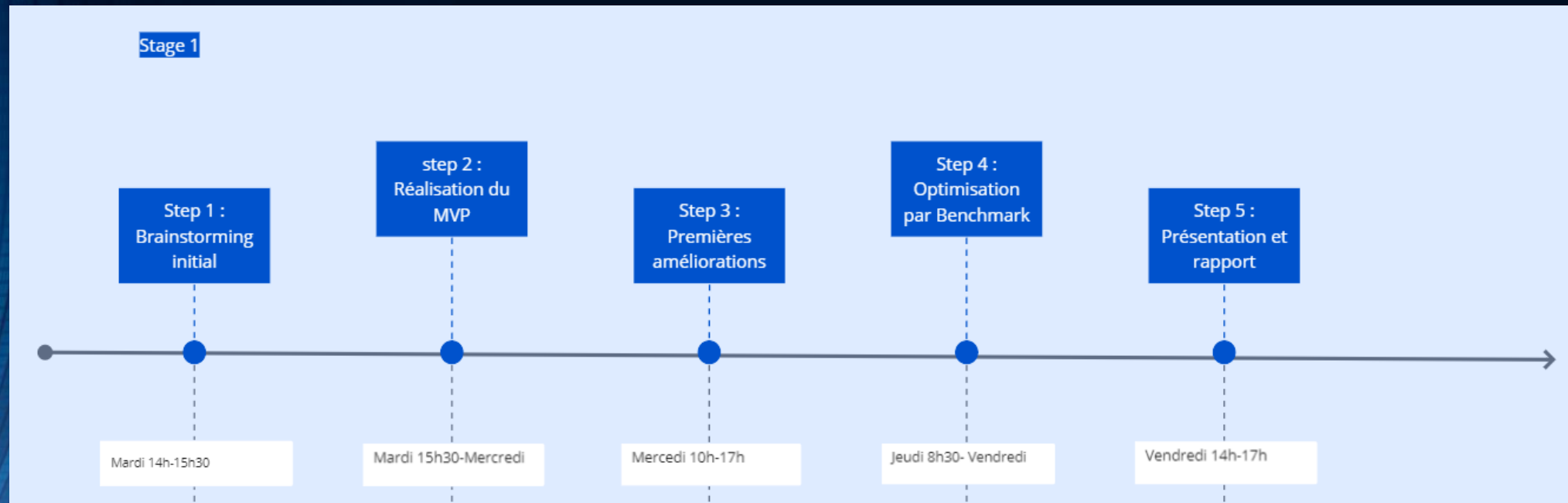
```
def best_score(nbindiv, nbgen, methode="Tournoi", alpha=0.59):  
    """ Permet de calculer le score du meilleur individu de la population pour une méthode donnée  
    à partir d'un nombre d'individu et d'un nombre de génération """
```



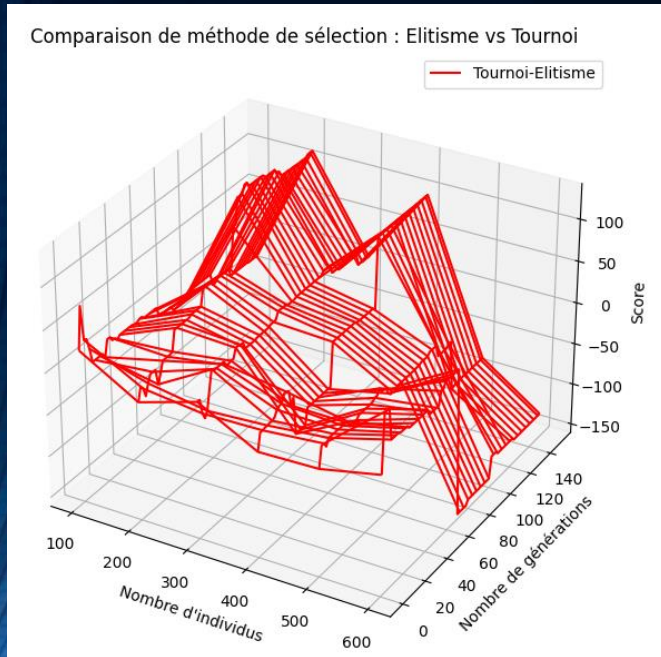




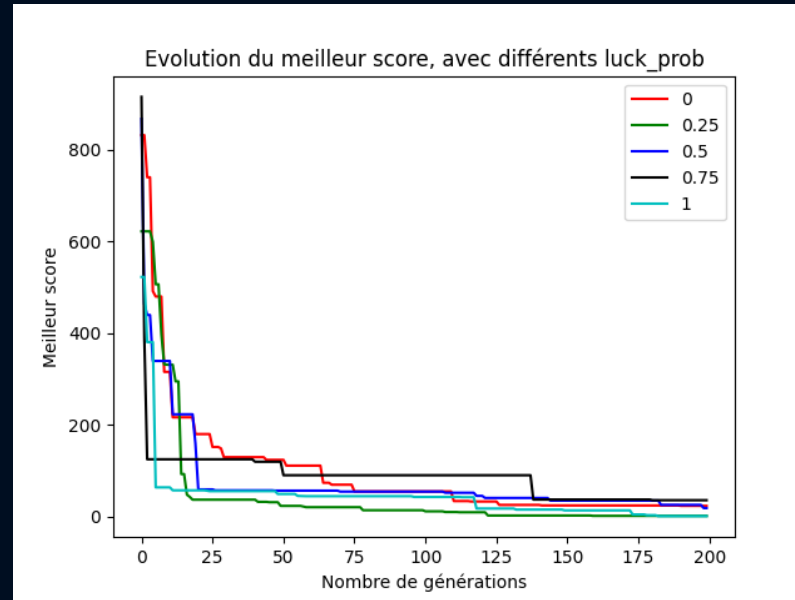
# Organisation



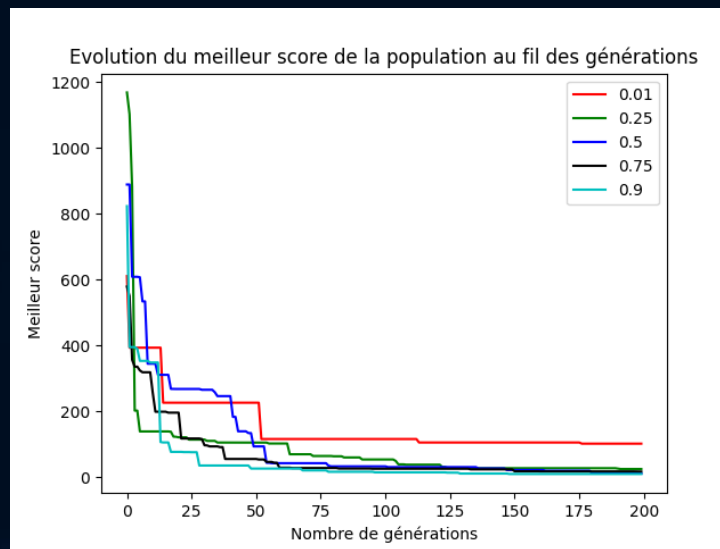
# Benchmark



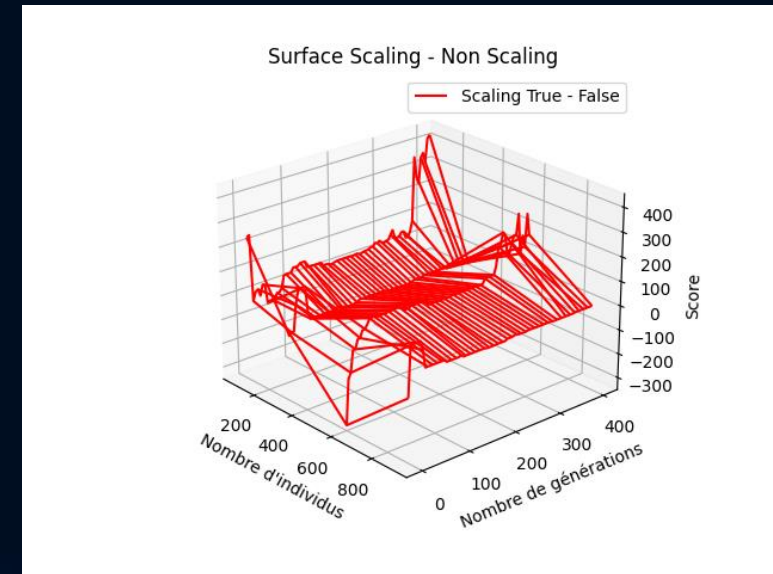
Elitisme VS Tournoi



Probabilité de chance



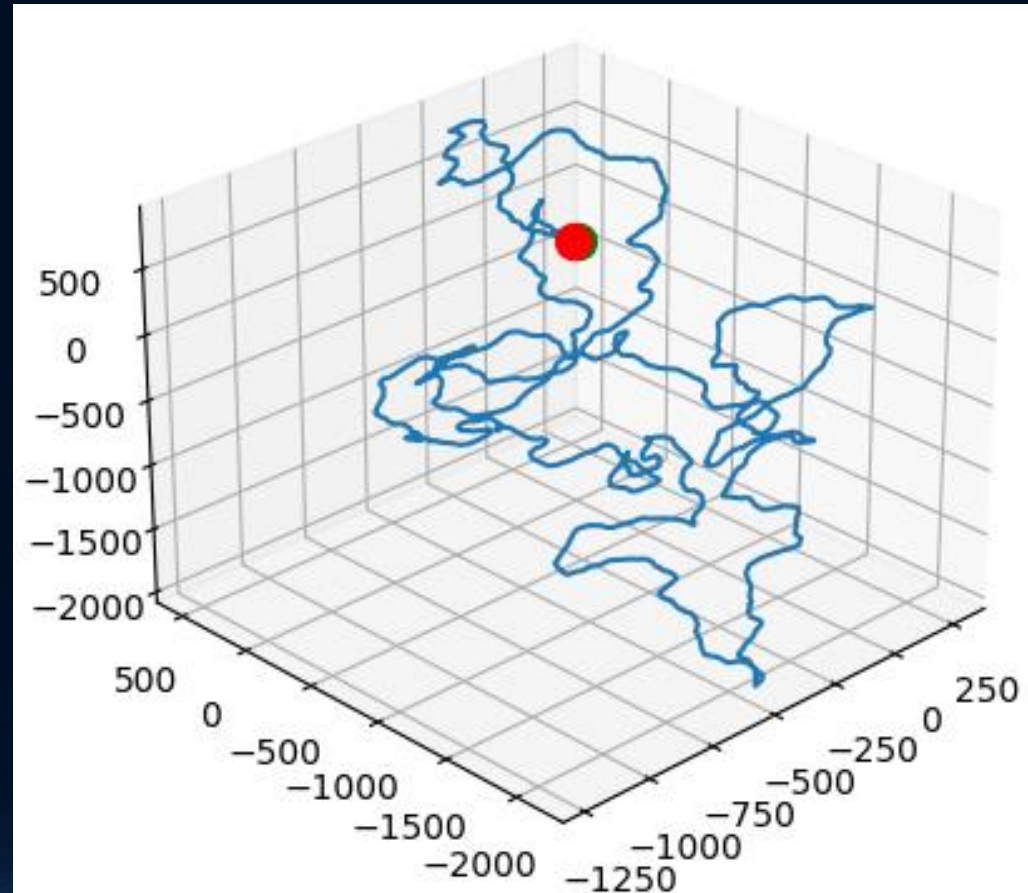
Probabilité de mutation

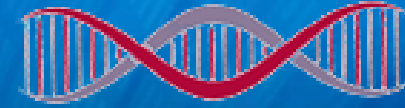


Avec et sans scaling mettre la relation d'ordre

# Résultats

- Meilleure distance obtenue : **1,56**





GENALG

# Démonstration & Conclusion

