

# Projets L3 informatique

Sabine Peres

[sabine.peres@univ-lyon1.fr](mailto:sabine.peres@univ-lyon1.fr)



Lyon 1

*Inria*



Biometry and Evolutionary Biology  
University Claude Bernard Lyon 1 - France

2025-2026

# SP1: Application web d'extraction de paramètres pour la modélisation métabolique

## Objectif:

Développer une application web permettant d'interroger une ou plusieurs bases de données (locales ou distantes) afin de récupérer automatiquement des paramètres nécessaires à la modélisation de système biologique (ex. concentrations, flux, cinétiques, conditions expérimentales, identifiants de réactions/métabolites, annotations).

# SP1: Fonctionnalités principales

Proposer une interface permettant à l'utilisateur de :

- ▶ choisir une source (base SQL / fichier / API si extension),
- ▶ définir une requête (formulaire guidé + requêtes avancées),
- ▶ sélectionner les variables/paramètres à extraire,
- ▶ appliquer des filtres (organisme, tissu, conditions, etc.).
- ▶ détecter les valeurs manquantes/aberrantes.
- ▶ Générer une sortie sous forme de fichier “modulable” selon les besoins

- ▶ **Interface web conviviale** avec une navigation intuitive. Options d'exportation des graphiques.
- ▶ Développement backend : Utiliser Python avec Flask pour gérer les données. Manipuler les données avec pandas.
- ▶ Visualisation frontend : Créer des graphiques interactifs avec Plotly ou Dash pour des diagrammes dynamiques.

## Compétences mises en œuvre:

- ▶ Science des données : pandas, NumPy, ...
- ▶ Visualisation : matplotlib, Plotly/Dash, ...
- ▶ Développement web : Flask, HTML/CSS/JS ou React ...

# SP2: Construction d'un automate cellulaire pour simuler les interactions entre cellules biologiques

## Objectif

- ▶ Développement d'un simulateur basé sur un **automate cellulaire** ou **un système multi-agents** permettant de modéliser les interactions entre cellules biologiques (exemple : prolifération cellulaire, communication, et réponse aux signaux environnementaux).
- ▶ Peut être appliqué à des domaines tels que la biologie des tissus, la modélisation des tumeurs, ou la réponse immunitaire.

## Langage et bibliothèques:

Python, Dash, NumPy, Pygame ...

## SP2: Description générale

L'idée est de construire un environnement où les cellules sont représentées par des agents ou des cellules sur une grille. Chaque cellule suit un ensemble de règles ou de comportements pour évoluer en fonction de :

- ▶ Ses propres caractéristiques internes (état, énergie, type, etc.).
- ▶ Les interactions avec ses voisins.
- ▶ Les signaux ou stimuli provenant de l'environnement.

Par exemple :

- ▶ Une cellule saine peut se diviser si elle a suffisamment de ressources.
- ▶ Une cellule tumorale peut proliférer rapidement et émettre des signaux pour perturber l'environnement.
- ▶ Une cellule immunitaire peut détecter et attaquer des cellules anormales.

# SP2: Fonctionnalités attendues

- ▶ Simulation d'un automate cellulaire ou d'un système multi-agents: - Une grille 2D ou 3D représente l'espace dans lequel les cellules interagissent. - Chaque cellule a des états possibles : sain, proliférant, apoptotique (mort), tumoral, immunitaire, etc.
- ▶ Modélisation des règles d'interaction: - Les cellules interagissent avec leurs voisines selon des règles simples : - Communication chimique (par gradients de molécules). - Transitions entre états en fonction de signaux reçus.
- ▶ Évolution dans le temps: - Visualisation dynamique de la croissance, de la mort ou de la propagation des cellules.
- ▶ Personnalisation des paramètres: - Taille de la grille ou de l'environnement. - Probabilités associées à certains événements (division, mort, etc.).
- ▶ Visualisation et export des résultats - Affichage graphique des états cellulaires au cours du temps. - Export des données (CSV, images) pour analyse ultérieure.

# SP3: Plateforme web SBML : édition, export et validation automatique

## Objectif

- ▶ Développer une **plateforme web** permettant de créer/éditer des modèles biologiques au format `SBML` (format xml pour les modèles métaboliques), de les exporter, et de les valider automatiquement
- ▶ en intégrant des outils Python externes, en particulier `memote` (suite de tests qualité pour modèles métaboliques) et `cobraPy` (pour prédire des comportements biologiques).

## Langage et bibliothèque (proposés):

Python, Flask, CobraPy Memote, Dash, Plotly



# SP3: Fonctionnalités attendues

- ▶ Édition SBML “web-friendly” : - Upload d'un SBML existant \*\*ou\*\* création à partir d'un gabarit minimal. - Édition guidée via formulaires (sans manipuler directement le XML)
- ▶ Validation automatique: - Lancer validation (Exécution côté serveur de memote) - Page “Résultats” interactive
- ▶ Corrections manuelles du modèle dans l'interface: - Modification guidée via formulaires (sans manipuler directement le XML). - Mettre à jour le modèle dans le SBML et lancer la validation.
- ▶ Exportation du modèle en SBML.