
This is the **published version** of the bachelor thesis:

Domínguez Ruiz, Pau. *Simulació basada en agents per a l'estudi de l'evolució bacteriana i la resistència als antibiòtics*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/334385>

under the terms of the  license.

Simulació basada en agents per a l'estudi de l'evolució bacteriana i la resistència als antibiòtics

Pau Domínguez Ruiz

29 de juny de 2026

Resum— La resistència bacteriana als antibiòtics és un problema sanitari creixent que emergeix de la interacció entre individus, mutacions i regulació interna. Aquest treball presenta un simulador basat en agents per estudiar l'evolució d'una població bacteriana sota pressió antibiòtica. Cada bacteri es modela com un agent autònom amb membrana cel·lular, xarxa reguladora gènica i metabolisme, en un nivell d'abstracció intermedi que intenta equilibrar la fidelitat mecànica amb la capacitat de simular moltes cèl·lules. La resistència no es defineix com una regla fixa, sinó que emergeix del balanç de fluxos entre subsistemes. S'ha desenvolupat un motor reutilitzable de simulació, visualització i telemetria, aplicat a *Escherichia coli* sota ampicil·lina i ciprofloxacina. Els experiments mostren dinàmiques coherents sense antibiòtic, eliminació o selecció de subpoblacions resistents segons la dosi d'ampicil·lina, i una recuperació més lenta amb ciprofloxacina, associada al dany a l'ADN i a una resposta de reparació simplificada.

Paraules clau— Simulació basada en agents, resistència als antibiòtics, evolució bacteriana, comportament emergent, xarxa reguladora gènica, nivell d'abstracció, modelització computacional.

Abstract— Bacterial antibiotic resistance is a growing public health problem that emerges from the interaction among individuals, mutations, and the internal regulation of each cell. This work presents an agent-based simulator to study how a bacterial population evolves under antibiotic pressure. Each bacterium is modelled as an autonomous agent with a cell envelope, a gene regulatory network, and a metabolism, using an intermediate level of abstraction that balances mechanistic fidelity and scalability. Resistance is not programmed as a fixed rule, it emerges from the balance of fluxes between subsystems. A reusable simulation, visualization, and telemetry engine has been developed and applied to *Escherichia coli* under ampicillin and ciprofloxacin. The experiments show coherent population dynamics without antibiotic, dose-dependent outcomes under ampicillin, and a slower recovery under ciprofloxacin due to DNA damage and a simplified repair response.

Keywords— Agent based simulation, antibiotic resistance, bacterial evolution, emergent behaviour, gene regulatory network, level of abstraction, computational modelling.



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

DURANT el segle XX, els antibiòtics van transformar certes infeccions mortals en problemes rutinaris. Quan pensem en els grans reptes de les pròximes dècades imaginem guerres, pandèmies o canvi climàtic, però n'hi ha un menys visible que ja és aquí: la pèrdua progressiva de l'eficàcia dels antibiòtics. L'OMS estima que el 2019 la resistència bacteriana va causar direc-

tament 1,27 milions de morts [1], xifra que les projeccions preveuen que s'incrementi prop d'un 70% el 2050 [2].

Entendre aquest fenomen és difícil perquè la resistència pot emergir de la interacció entre molts individus, les mutacions, la regulació interna de cada cèl·lula i la pressió selectiva del propi antibiòtic. Els models poblacionals agregats, que descriuen la població amb variables globals, deixen escapar bona part d'aquesta dinàmica. La simulació basada en agents ofereix una alternativa on es representa cada bacteri com una entitat autònoma, i els fenòmens col·lectius com poden ser el creixement o l'aparició de soques resistents emergeixen de regles individuals.

Tota simulació simplifica la realitat, una sola cèl·lula bacteriana conté milers de gens, proteïnes i reaccions operant ara, i reproduir-ho tot no és computacionalment viable

- E-mail de contacte: 1671780@uab.cat
- Menció: Computació
- Treball tutoritzat per: Yolanda Benítez Fernández (Àrea de Ciències de la Computació i Intel·ligència Artificial)
- Curs 2025/26

quan es simulen milers d'agents simultàniament. Escollir el nivell d'abstracció adequat, què conservar i què descartar, és una de les decisions més determinants del disseny. Treballs previs en aquest àmbit cobreixen un ampli ventall d'enfocaments, però sovint representen cada bacteri amb un comportament intern molt limitat o bé queden massa lligats a un cas biològic concret.

2 OBJECTIUS

2.1 Objectiu principal

L'objectiu principal d'aquest treball és estudiar, des d'una perspectiva computacional, com evoluciona una població bacteriana davant la presència d'un antibiòtic, utilitzant la simulació no només com a eina d'implementació sinó com a instrument d'anàlisi per extreure conclusions sobre el comportament del sistema.

2.2 Objectius secundaris

- Modelar el comportament individual d'un bacteri com a agent autònom amb subsistemes interns, analitzant com la interacció entre múltiples agents i un entorn amb antibiòtics genera comportaments emergents a nivell poblacional.
- Avaluar la capacitat del model per reproduir dinàmiques observades en bacteris reals, com *Escherichia coli*, comparant variables com taxes de creixement, proporció d'individus resistents i evolució temporal de la població sota diferents concentracions d'antibiòtics.
- Desenvolupar una eina informàtica parametrizable que faciliti l'experimentació amb diferents escenaris i ofereixi una base extensible per a l'estudi computacional de sistemes biològics.

3 ESTAT DE L'ART

L'estudi computacional de poblacions bacterianes s'ha abordat històricament des de dues grans famílies. D'una banda, els models agregats, que descriuen la població mitjançant variables globals i equacions que governen la seva evolució, els quals són útils per capturar tendències i s'han validat experimentalment en plataformes com eVOLVER [3], però dilueixen l'individu i la seva heterogeneïtat interna en mitjanes poblacionals. De l'altra, els models basats en agents, que representen cada bacteri com una entitat autònoma i deixen que la dinàmica col·lectiva emergeixi de les regles individuals. La qüestió de fons en qualsevol d'aquests enfocaments no és tant la tècnica utilitzada, sinó quant detall conservar de la cèl·lula real.

A l'extrem de màxim detall hi ha el paradigma *whole-cell*. El model de *Mycoplasma genitalium* de Karr, Covert *et al.* (Stanford, 2012) [4] representa l'intent més ambiciós de simular computacionalment una cèl·lula completa. Es va escollir *M. genitalium* precisament perquè és l'organisme autoreplicant amb el genoma més petit conegut (525 gens), però malgrat això el model va requerir més de 60 paràmetres per mòdul i dècades de treball experimental per parametritzar-lo. Integra 28 processos cel·lulars basats en

més de 900 fonts primàries i ofereix una fidelitat molecular notable. Tanmateix, per a organismes més complexos com *E. coli* amb més de 4.000 gens, un enfocament equivalent és computacionalment inviable en el context de simulacions multiagent on s'instancien milers de cèl·lules simultàniament.

D'una altra banda, tenim els entorns de simulació basats en agents. Els frameworks genèrics com NetLogo [5], Repast o Mesa [6] proporcionen la infraestructura per definir agents, gestionar l'entorn i visualitzar resultats, però no són específics de biologia: el comportament intern de cada cèl·lula sol quedar reduït a unes poques regles que l'investigador ha de construir des de zero. Entre aquests i el detall molecular hi ha eines especialitzades en bacteris, com BSim [7], gro [8] o iDynoMiCS [9]. BSim, per exemple, incorpora trets bacterians predefinitos i dinàmiques internes descrites mitjançant xarxes reguladores en forma d'equacions diferencials ordinàries, mentre que iDynoMiCS està orientat a comunitats i biofilms i arriba a escalar fins a milions d'agents en 3D. Són plataformes molt capaces, però sovint queden lligades a un context concret, biologia sintètica, morfologia de biofilm, i el model intern de la cèl·lula tendeix a ser fix o secundari respecte a la finalitat de l'eina.

Pel que fa específicament a la resistència antibiòtica, hi ha desenvolupaments com ARES [10], basat en computació amb membranes (P-systems), on la cèl·lula no es modela com una entitat amb estructura interna pròpia sinó com objectes abstractes com gens de resistència, plasmidis i antibiòtics, situats dins compartiments jeràrquics. La resistència emergeix de regles que els transformen. Aquest tipus d'enfocaments capturen bé la dinàmica evolutiva, però la maquinària interna de cada bacteri queda abstreta, de manera que la resistència es representa com un objecte o una taxa més que no pas com una conseqüència del seu funcionament intern.

Els models més detallats no escalen a poblacions, els entorns d'agents escalables tendeixen a mantenir la cèl·lula simple o a fixar el comportament, i els models centrats en resistència abstraen l'individu. Queda relativament poc explorat un punt intermedi: un agent amb prou estructura interna, regulació gènica, metabolisme i embolcall, perquè els mecanismes de resistència emergeixin de la interacció entre aquests subsistemes en lloc d'estar programats com una regla fixa, mantenint alhora un nivell d'abstracció prou gros per simular moltes cèl·lules alhora. És en aquest espai on s'emmarca el present treball.

4 METODOLOGIA

Durant el desenvolupament s'ha seguit una metodologia àgil, iterativa i incremental, prenent Kanban com a referència. La gestió s'ha portat amb un tauler de Trello adaptat a les dimensions del projecte utilitzant un flux de columnes. La simulació es planteja com una eina experimental i d'anàlisi que no es pot definir completament des de l'inici, sinó que ha de créixer per capes validades, on en alguns casos, cada ampliació depèn del que s'observa en l'anterior.

Partint d'una versió mínima del model, amb cada bacteri representat com un agent amb funcions bàsiques, s'han incorporat progressivament noves capes de complexitat, tant a la infraestructura de simulació com al model biològic. En cada iteració s'ha treballat en el disseny i la implementació

d'una nova funcionalitat o la millora d'un component existent. Un cop consolidada una base sòlida del motor, s'han executat simulacions sota diferents escenaris per analitzar l'impacte dels canvis en la dinàmica global i avaluar progressivament la coherència del model.

El codi s'ha gestionat en un repositori Git, integrant els canvis mitjançant pull requests revisades abans de fusionar-les, per mantenir un historial clar de l'evolució del sistema i una branca principal estable sobre la qual iterar.

5 PLANIFICACIÓ

El treball es va organitzar al voltant d'una planificació inicial estructurada en set fases parcialment solapades. Aquesta planificació s'ha anat ajustant a mesura que els resultats de cada fase concretaven l'abast de la següent. L'exemple més significatiu és la decisió, presa després de l'exploració inicial, de desenvolupar un motor de simulació propi en lloc de partir d'un framework existent, cosa que va obligar a avançar part del treball d'arquitectura i a redimensionar la càrrega associada al motor.

Fase exploratòria i preparació

15-02-2026 – 07-03-2026

Anàlisi comparativa de diferents alternatives tecnològiques per a la simulació basada en agents, amb desenvolupament de proves de concepte i simulacions mínimes. Aquesta fase va fixar la base tècnica del projecte i es va prendre la decisió d'implementar un motor propi.

Fase de definició del model base i del nucli del motor

08-03-2026 – 11-04-2026

Definició formal del model inicial del bacteri com a agent autònom i de l'entorn mínim de simulació. En paral·lel es va desenvolupar una primera base del motor, amb els components inicials de renderitzat i una estructura bàsica d'interfície que permetia executar i visualitzar simulacions.

Fase d'arquitectura i estructura extensible

28-03-2026 – 08-05-2026

Consolidació d'una arquitectura modular que separa clarament l'aplicació principal, el motor de simulació, la capa de visualització i el model biològic. Es va definir la interfície interna entre mòduls i l'API del motor, i es va reforçar la gestió de dades, la telemetria i la documentació interna.

Fase de modelat intern progressiu del bacteri

19-04-2026 – 30-05-2026

Definició de l'estructura interna de l'agent i dels seus mecanismes, mantenint la separació entre la lògica del motor i la implementació biològica concreta. L'agent passa de ser purament reactiu a tenir una estructura interna coherent i ampliable.

Fase d'introducció de l'antibiòtic i pressió selectiva

10-05-2026 – 07-06-2026

Implementació d'un sistema de difusió sobre un subespai discret per representar la distribució espacial de substàncies com els nutrients i l'antibiòtic, modelat de l'efecte de l'antibiòtic sobre els bacteris sensibles i incorporació progressiva dels mecanismes de resistència, la mutació i l'herència.

Fase d'experimentació i anàlisi computacional

31-05-2026 – 21-06-2026

Execució de simulacions sota diferents escenaris per validar el comportament del model i obtenir resultats, recollint mètriques com la dinàmica poblacional, el consum de recursos i l'expressió gènica sota diferents concentracions d'antibiòtic.

Fase de consolidació i preparació final

14-06-2026 – 04-07-2026

Revisió de codi, proves i documentació tècnica, preparació dels gràfics i anàlisis.

6 NIVELL D'ABSTRACCIÓ ADOPTAT

6.1 El problema de simular vida

La biologia computacional enfronta un dilema, i és que la realitat biològica és incommensurablement complexa. Una sola cèl·lula d'*Escherichia coli* conté aproximadament 4.000 gens, milers de tipus de proteïnes i més de 2.000 reaccions metabòliques caracteritzades [14] i processos fisicoquímics que operen en escales temporals que van des dels femtosegons fins a les hores. Simular fidelment tots aquests fenòmens amb la física i química subjacent està fora de l'abast computacional actual, especialment quan es tracta de representar milers d'agents bacterians simultàniament.

Aquesta limitació no és exclusiva de la biologia. Un paral·lelisme és la predicció meteorològica: l'atmosfera obeeix lleis físiques, però modelar directament totes les seves interaccions microscòpiques és computacionalment inviable. La meteorologia moderna va sorgir quan es va adoptar models macroscòpics basats en variables agregades com la temperatura, pressió, humitat, velocitat del vent, juntament amb equacions que descriuen la seva evolució. Investigadors com Edward Lorenz [13] van demostrar que fins i tot sistemes simplificats poden exhibir dinàmiques caòtiques extremadament sensibles a les condicions inicials.

Els sistemes biològics plantegen un repte semblant: són adaptatius, no lineals i presenten comportaments emergents que no sempre es poden deduir dels seus components individuals. A més, a diferència de l'atmosfera, les cèl·lules evolucionen i modifiquen el seu comportament al llarg del temps.

6.2 Nivell d'abstracció adoptat

El model que s'ha proposat i implementat opera al nivell de xarxes reguladores i fluxos metabòlics agregats. Aquest nivell, més gros que el molecular però més fi que el purament fenotípic, té precedents ben establerts en la literatura: models de Flux Balance Analysis (FBA) [12], models d'equacions diferencials ordinàries de vies de senyalització, i models d'agents com PhysiCell [11] operen en rangs similars.

Les decisions d'abstracció que considerem més rellevants són quatre:

Concentracions contínues, no comptatges discrets. Cada espècie química es representa com una variable real en mil·limolar sobre un volum efectiu de cèl·lula o de cel·la de l'entorn. No es simulen molècules individuals ni es manté un comptatge enter, la dinàmica intracel·lular es descriu per equacions diferencials ordinàries integrades a cada pas de simulació. Aquesta representació simplifica dramàticament

el cost computacional sense perdre la capacitat de capturar comportaments macroscòpics.

Expressió gènica com a nivell d'activació normalitzat. No es modela explícitament la transcripció d'ARN ni la traducció a proteïnes. En lloc d'això, cada gen es representa com un valor continu normalitzat que indica el nivell d'activitat funcional en cada instant. La xarxa reguladora gènica modula aquests nivells en resposta a senyals de l'estat metabòlic cel·lular.

Reaccions metabòliques agregades amb cinètiques seleccionables. Cada reacció del graf metabòlic no representa una col·lisió molecular individual sinó una via agregada, la glicòlisi completa, la fosforilació oxidativa, o la síntesi de paret cel·lular com a unitats úniques. El flux de cada reacció es calcula utilitzant una cinètica enzimàtica triada per reacció (Michaelis-Menten, Hill o lineal), parametritzada amb constants com $V_{\max}$ i K_m . L'actualització de concentracions és una operació vectorial que suma la contribució de cada reacció en cada pas temporal. D'aquesta forma, el cost computacional per agent és $O(k)$, on k és el nombre de reaccions.

Dany estructural com a metabòlit ordinari. La integritat de la paret cel·lular i de l'ADN no es representen com a estats especials sinó com a fraccions contínues, tractades com metabòlits del mateix graf. Apareixen com a producte de reaccions de síntesi (síntesi de paret, reparació d'ADN) i com a substrat de reaccions de degradació (autòlisi, dany espontani). L'efecte d'un antibiòtic β -lactàmic no es programa com una regla fixa. En lloc d'això, l'antibiòtic redueix l'activitat efectiva de l'enzim de síntesi de paret en proporció a la seva ocupació de la diana. El dany emergeix del desequilibri entre síntesi i degradació, propagat per la dinàmica del graf metabòlic. Tota la fenomenologia de resistència i mort cel·lular es redueix a balanços de flux.

D'aquesta manera, els mecanismes de resistència emergeixen de les interaccions entre els diferents subsistemes de la cèl·lula, en lloc de ser definits explícitament com a comportaments fixos. La sensibilitat o resistència d'un bacteri davant d'un antibiòtic és, per tant, una conseqüència de l'estat intern i de la dinàmica del model.

6.3 Cas d'estudi: *Escherichia coli*

La implementació s'ha centrat en *E. coli*, un bacteri gram-negatiu [16] àmpliament utilitzat com a organisme model. S'ha escollit perquè és l'organisme procariota més estudiat, els seus mecanismes de resistència a antibiòtics són de rellevància clínica directa [15], i existeix abundant informació experimental que permet validar les dinàmiques simulades. Els antibiòtics d'estudi, ampicil·lina i ciprofloxacina, representen dos mecanismes de resistència distints: inhibició de síntesi de paret i inhibició de síntesi d'ADN, respectivament.

7 ARQUITECTURA DEL SIMULADOR

7.1 Visió general

El simulador està organitzat com una solució de Visual Studio composta per sis projectes amb responsabilitats clarament delimitades (Fig. 1). Aquesta estructura permet mantenir el nucli de simulació completament independent de la

interfície gràfica, de manera que la lògica biològica pot evolucionar sense dependre del renderitzat. Tots els projectes es compilen com a llibreries estàtiques, excepte App, que actua com a punt d'entrada i genera un executable.

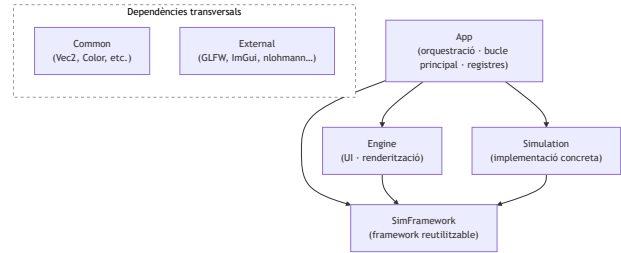


Fig. 1: Organització en projectes de la solució i les seves dependències.

App és l'únic projecte que depèn de tots els altres. Inicialitza els subsistemes, carrega o crea un projecte, registra els models disponibles i executa el bucle principal de l'aplicació.

Engine gestiona tot el relacionat amb la presentació: finestra nativa, context OpenGL, renderitzat 2D, càmera, interfície d'usuari amb panells acoblables. Consumeix les dades de la simulació en mode de només lectura, de manera que mai pot alterar l'estat del model.

SimFramework és el nucli reutilitzable del sistema. Defineix les abstraccions fonamentals per a la simulació multiagent: Agent (entitat autònoma), Model (gestor de col·lecció d'agents), Microenvironment (espai físic amb camps escalars), SimulationSession (control del temps), SimulationManager (gestió de projectes), i Telemetry (sistema de mètriques). No té cap coneixement del domini biològic ni del sistema de renderitzat, i podria reutilitzar-se per a simular qualsevol altre sistema multiagent.

Simulation és la implementació concreta del model bacterià. Hereta de les abstraccions de SimFramework i és on resideix tota la lògica d'*E. coli*: subsistemes biològics, configuració basada en dades, comportament dels agents.

Common agrupa tipus de dades compartits entre projectes, com vectors 2D o paletes de colors.

External compila totes les dependències de tercers com a llibreria estàtica: GLFW (finestra nativa), GLAD (cargador OpenGL), Dear ImGui (interfície d'usuari), nlohmann/json (serialització), i utilitats gràfiques.

Pel que fa a la reproductibilitat tècnica, tots aquests components queden compilats com a biblioteca estàtica dins la mateixa solució. L'únic requisit per compilar és Visual Studio 2022 o posterior amb el Windows SDK 10.0 i suport per a C++17, obrint el fitxer `.sln` i compilant la solució completa es genera l'executable.

7.2 Framework de simulació

El SimFramework defineix els components fonamentals:

Agent és la unitat bàsica de simulació. Cada instància posseeix un cos físic amb posició, velocitat, orientació i forma de col·lisió. Exposa tres punts d'extensió que les subclasses concretes han d'implementar: `step()` per a la lògica de comportament per tick, `collectInspectionData()` per a proveir dades al panell d'inspecció de la UI, i `collectTelemetryData()` per a alimentar el sistema de mètriques.

Model gestiona la col·lecció d'agents i coordina cada tick de simulació. Entre altres responsabilitats, manté el sistema determinista amb una llavor (seed), integra la física bàsica dels cossos (moviment, resolució de col·lisions mitjançant una malla espacial), i permet configurar l'ordre d'actualització dels agents.

Microenvironment representa l'espai físic com una graella 2D de camps escalars continus. Cada camp pot modelar qualsevol substància (nutrient, antibiòtic, etc.) amb difusió espacial i decaïment temporal. Els agents poden injectar o extreure quantitat de qualsevol camp en la seva posició, mostrar concentracions o calcular gradients per orientar el seu moviment.

SimulationSession envolta el model actiu i gestiona el temps de simulació. Desacobla el temps real (dt de cada fotograma) del temps simulat: un acumulador permet avançar diversos ticks per fotograma a diferents velocitats, pausar, o executar pas a pas. Després de cada avanç, produeix un FrameSnapshot (una còpia serialitzable de l'estat visible) que consumeixen el renderitzat i la telemetria.

SimulationManager és un gestor que manté un registre dels tipus de model disponibles i una biblioteca de presets (configuracions guardades). Gestiona projectes (crear, carregar, guardar en disc en JSON) i exposa la sessió activa a la resta de l'aplicació.

Telemetria és el sistema de mètriques definides per l'usuari. En lloc de codificar què es mesura, el framework defineix un esquema de dades que els agents publiquen cada tick. Des del motor es poden definir mètriques mitjançant expressions avaluades sobre aquestes dades, que s'acumulen com a sèries temporals que es poden visualitzar en gràfiques o exportar. D'aquesta manera no cal modificar el codi per afegir noves mètriques.

7.3 Motor de visualització

El projecte Engine aïlla tot el relacionat amb la presentació. El Renderer gestiona la finestra nativa, el context OpenGL i dibuixa la vista de simulació com una textura en un viewport amb càmera 2D navegable. La GuiLayer hi construeix la interfície amb Dear ImGui, amb un sistema de finestres acoblables que l'usuari pot reorganitzar lliurement.

Les finestres rellevants reben únicament el FrameSnapshot i les dades de telemetria, sense accés a l'estat intern del model, de manera que la UI és independent de la lògica de simulació.

Model Manager gestiona presets: permet crear, duplicar, eliminar i editar configuracions amb els paràmetres del model i les condicions inicials.

Viewport és la finestra principal, mostra en temps real l'estat de la simulació agents i entorn, amb navegació per zoom i desplaçament i selecció d'agents.

Simulation és el panell de control: botons de reproducció, pausa, avanç pas a pas i control de velocitat, i informació d'estat com el temps simulat i el nombre d'agents actius.

Inspector mostra en temps real l'estat intern de l'agent o regió seleccionada (variables exposades del bacteri o concentracions de l'entorn), i s'utilitza per a anàlisi i depuració.

Telemetry és el gestor de mètriques: permet definir-les mitjançant expressions avaluades sobre les dades publicades, que es validen i actualitzen en temps real.

Metric Chart visualitza mètriques temporals; es poden obrir múltiples instàncies, cada una amb una mètrica distinta, i exportar gràfiques com a imatge o dades en CSV.

8 MODEL COMPUTACIONAL DEL BACTERI

8.1 Definició formal de l'agent

Cada bacteri és un agent computacional autònom definit com una tupla:

$$A = \langle S, G, \Phi, \Xi \rangle$$

TAULA 1: ELEMENTS PRINCIPALS DE LA REPRESENTACIÓ FORMAL DE L'AGENT

Component	Tipus	Contingut
S	Vector d'estat	Concentracions per compartiment, pools metabòlics, nivells GRN, edat i progrés de divisió.
G	Vector mutable	Genoma: paràmetres reals que modulen l'expressió de transportadors i enzims.
Φ	Funció de transició	$\Phi : (S, G, E, \Delta t) \rightarrow S'$. Pipeline de cinc capes que actualitzen l'estat.
Ξ	Predicats de control	Regles que determinen la persistència, divisió o mort de l'agent.

L'agent s'organitza en dos compartiments físics (periplasma i citoplasma) i tres subsistemes: l'embolcall cel·lular, que regula l'intercanvi amb l'entorn; la xarxa reguladora gènica, que actua com a pla de control intern; i la xarxa metabòlica, que transforma matèria i energia. Els predicats de control operen sobre aquests subsistemes, que no es comuniquen entre si mitjançant missatges sinó llegint i escrivint sobre una representació única i compartida de l'estat cel·lular.

8.2 Representació de l'estat

L'estat intern S es compon de les estructures següents:

8.2.1 Concentracions per compartiment

El periplasma i el citoplasma mantenen cadascun un mapa $\text{ChemicalId} \rightarrow \text{concentració } (\mu\text{M})$ de cada molècula present. El periplasma conté les substàncies que han travessat la membrana externa (antibiòtics, nutrients, etc.), mentre que el citoplasma conté els químics que participen en les reaccions internes (glucosa, piruvat, ATP/ADP, NADH/NAD⁺, peptidoglicà, entre d'altres).

8.2.2 Nivells d'expressió gènica

Un mapa $\text{GeneId} \rightarrow \text{nivell continu}$ que indica la quantitat funcional de cada proteïna. Aquests nivells determinen la

cinètica de les reaccions metabòliques (escalant $V_{\max}$ de cada enzim), la permeabilitat de transportadors i bombes a l'embolcall, i la producció de senyals globals que governen el creixement i la divisió.

8.2.3 Senyals derivades

Valors escalars que emergeixen de la GRN i governen comportaments globals: `growth_rate_modifier`, que modula el creixement segons l'estat de la cèl·lula; `cell_division_gate`, que permet o bloqueja la divisió; i `mutation_rate`, la taxa de mutació vigent, amplifiable en estrès.

8.2.4 Escalars temporals i energètics

Altres valors escalars que registren la història temporal i l'estat energètic de la cèl·lula: `age_seconds` (edat total); `growth_progress`, acumulador que creix en condicions favorables i determina quan es divideix; `atp_charge`, la proporció $ATP/(ATP+ADP)$ que representa l'energia disponible i s'usa en tots els subsistemes; i `generation` (nombre de divisions acumulades).

8.2.5 Genoma heretable (separat)

El genoma G es guarda independentment de l'estat dinàmic perquè es copia i muta entre generacions: `basal_levels` (nivells basals d'expressió per gen, heretats i mutables), `target_kd_mult` (multiplicadors de K_d de les dianes de fàrmacs, resistència de diana) i `mutation_rate` (taxa de mutació base de la cèl·lula).

8.3 Pipeline d'actualització Φ

En cada tick, l'agent segueix un pipeline ordenat de cinc etapes (Fig. 2) on cadascuna llegeix de l'estat compartit i en deixa un de consistent per a la següent. L'ordre és fix i canviar-lo no invalida la simulació però altera lleugerament la trajectòria de cada tick.

8.4 Subsistema 1: Embolcall cel·lular

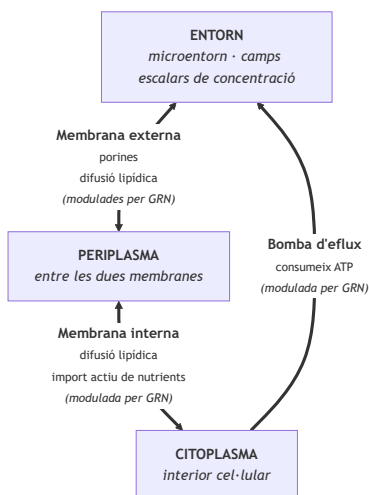


Fig. 3: Model del subsistema d'embolcall cel·lular

L'embolcall (Fig. 3) modela el transport de substàncies com un graf de fluxos entre tres compartiments: l'entorn

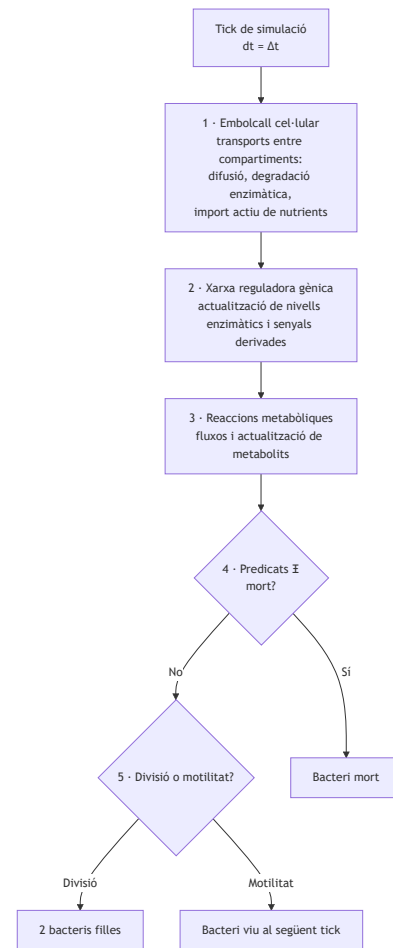


Fig. 2: Pipeline d'actualització de l'agent durant un tick de simulació

(camps escalars del microentorn), el periplasma (l'espai entre les membranes externa i interna dels bacteris gram-negatius [16]) i el citoplasma (l'interior cel·lular), on tota molècula transportable segueix el mateix recorregut.

El travessament de cada membrana es resol com un flux net proporcional a la diferència de concentració entre els dos compartiments, ponderat per una permeabilitat efectiva. Aquesta permeabilitat agrega una difusió lipídica passiva fixada per propietats de la molècula, i la contribució de transportadors com les porines a la membrana externa, import actiu de nutrients i bombes d'eflux a la interna. A diferència de la difusió, el cabal dels transportadors s'escala pel nivell d'expressió del gen que els codifica, de manera que la GRN pot regular la permeabilitat i la capacitat d'expulsió de la cèl·lula. L'eflux és transport actiu i utilitza energia per a bombejar, afegint el cost de fitness de la resistència.

8.5 Subsistema 2: Xarxa reguladora gènica

La xarxa reguladora gènica (GRN, Fig. 4) és un graf dirigit sobre els gens que actua com el pla de control intern de la cèl·lula. Cada gen té un nivell d'expressió continu que indica quanta proteïna funcional codifica en cada instant, i aquest nivell s'ajusta segons l'estat de la cèl·lula. Per a cada gen es calcula primer un nivell objectiu a partir de tres factors: un nivell basal propi (heretable i, per tant, mutable entre generacions), la resposta als senyals que percep com la concentració d'un antibiòtic, la quantitat d'un metabolit o la

càrrega energètica de la cèl·lula traduïda mitjançant funcions de Hill que poden activar o inhibir el gen, i la influència d'altres gens, que el poden empènyer amunt o avall amb un cert pes. El nivell del gen no salta a aquest objectiu immediatament, sinó que hi tendeix de manera gradual amb una constant de temps τ , cosa que dona a la regulació una inèrcia realista i evita canvis bruscos.

Aquests nivells d'expressió són els que governen la resta de l'agent, i s'apliquen de tres formes: sobre el metabolisme, escalant la velocitat de l'enzim que cada gen codifica, sobre l'embolcall, regulant la permeabilitat de les porines i l'activitat de les bombes d'eflux i sobre els predicats de cicle (Ξ), a través de senyals globals com el modulador de creixement, la comporta de divisió o la taxa de mutació.

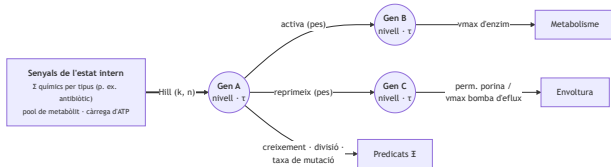


Fig. 4: Esquema de la xarxa reguladora gènica (GRN) de l'agent

8.6 Subsistema 3: Xarxa metabòlica

El metabolisme (Fig. 5) es representa com un graf bipartit entre dos tipus de nodes: els metabòlits, cadascun amb un pool de concentració en mil·limolar, i les reaccions, cadascuna amb la seva estequiometria (quins metabòlits consumeix i produeix, i en quina proporció) i la seva cinètica associada. Una mateixa reacció pot tenir diversos substrats i diversos productes, de manera que el graf connecta els dos conjunts de nodes en una xarxa de fluxos.

A cada tick l'actualització segueix dues fases. Primer es calcula el flux de cada reacció a partir de les concentracions actuals dels seus substrats, la seva cinètica (Michaelis-Menten, Hill o lineal) i el nivell enzimàtic que la GRN hi imposa, que escala la velocitat màxima de la reacció. Després els metabòlits s'actualitzen en bloc aplicant aquests fluxos sobre l'estequiometria, descomptant el que cada reacció consumeix i sumant el que produeix.

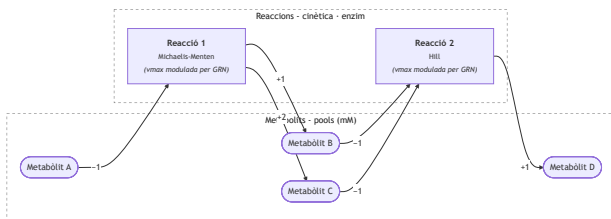


Fig. 5: Esquema de la xarxa metabòlica de l'agent

8.7 Subsistema 4: Cicle cel·lular i resposta a l'entorn

Sobre els tres subsistemes anteriors operen dos mecanismes que completen el comportament de l'agent: els predicats de cicle (Ξ) i la resposta motora a l'entorn. Els predicats de cicle decideixen, a cada tick, si la cèl·lula persisteix, es divideix o mor a partir del seu estat intern.

8.7.1 Divisió

La cèl·lula acumula un progrés de creixement que depèn alhora del seu estat energètic, de la regulació gènica i de la integritat de la paret i de l'ADN: si qualsevol d'aquests factors falla, el progrés s'atura. Quan supera un límit i es compleixen les condicions mínimes d'edat i d'energia, la cèl·lula es divideix en dues filles, cadascuna amb una còpia del genoma que pot patir petites mutacions.

8.7.2 Mort

En lloc d'un llindar abrupte, la mort es modela com un risc que augmenta a mesura que la integritat de la paret, de l'ADN o l'energia cel·lular s'acosten a valors crítics, i que es tradueix en una probabilitat de morir a cada tick.

8.7.3 Resposta motora

El moviment es modela amb una quimiotaxi simplificada on la cèl·lula compara la concentració actual de nutrient amb una memòria de la que ha trobat recentment i ajusta la seva orientació segons si l'entorn millora o empitjora. A més, la velocitat depèn del balanç energètic, de manera que una cèl·lula sense energia deixa de moure's.

9 RESULTATS

9.1 El simulador com a plataforma

El primer resultat del treball és la pròpia eina: una aplicació d'escriptori funcional i estable que executa simulacions multiagent completes en temps real, amb la lògica del model separada de la infraestructura tècnica. El nucli de simulació no coneix ni la biologia ni el renderitzat, de manera que el model del bacteri és només una de les implementacions possibles sobre una base reutilitzable: qualsevol altre sistema multiagent, com difusió d'opinions, trànsit o dinàmiques ecològiques, es podria implementar sense tocar el motor.

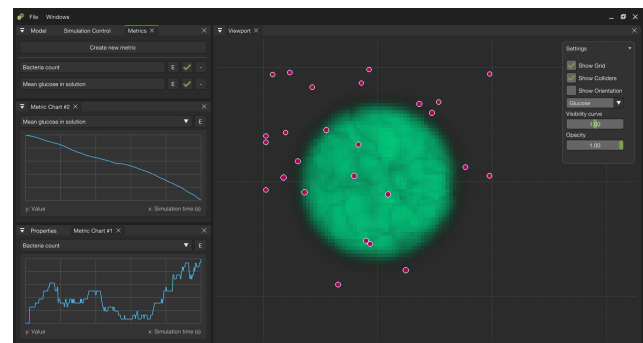


Fig. 6: Captura de l'aplicació en execució. El viewport mostra la població d'agents sobre el camp de glucosa de l'entorn, mentre que els panells laterals recullen el control de simulació, la definició de mètriques i les gràfiques de telemetria.

Sobre aquesta base, l'aplicació cobreix el cicle complet d'experimentació: gestiona el cicle de vida dels agents i l'avanç temporal, la interacció amb un entorn físic de camps escalars, i la persistència de projectes i configuracions en disc, cosa que permet desar, recuperar i compartir escenaris.

L'execució és determinista a partir d'una llavor perquè els experiments siguin reproduïbles i comparables entre si.

La capa de visualització (Fig. 6) fa l'eina utilitzable i interpretable: el viewport renderitza en temps real els agents i els camps de l'entorn amb una càmera 2D navegable, i una interfície de panells acoblables permet combinar el control de la simulació, la inspecció d'agents i la consulta de mètriques en una sola pantalla reorganitzable.

Finalment, el sistema de telemetria transforma l'eina en un instrument d'anàlisi: en lloc de codificar quines variables es mesuren, l'usuari defineix mètriques mitjançant expressions avaluades sobre les dades que els agents publiquen a cada tick, sense recompilar, i s'acumulen com a sèries temporals visualitzables en gràfiques o exportables en CSV. Tots els resultats que es presenten a continuació, com ara corbes de població, consum de glucosa o expressió gènica, s'han obtingut només definint les mètriques adequades.

9.2 Rendiment i escalabilitat

Per avaluar els recursos que consumeix el simulador s'ha instrumentat el bucle principal amb mesures de temps per tick i memòria privada del procés, executant el model *E. coli* en configuració Release sobre un AMD Ryzen 7 8700F (8 nuclis / 16 fils, 4,1 GHz), 16 GB de RAM DDR5 i Windows 11. El pas dels agents és seqüencial: tot el càlcul biològic recau sobre la CPU i depenent de la velocitat de simulació pot saturar un sol nucli, mentre que la GPU intervé en el renderitzat. La simulació parteix de 1 agent i es deixa créixer lliurement fins a superar els 2 500 agents.

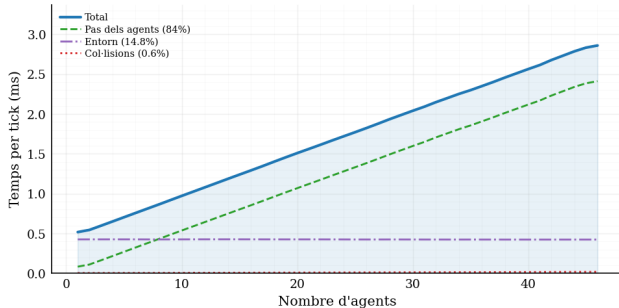


Fig. 7: Temps de càlcul per tick desglossat per fases del bucle de simulació, en el rang baix de població (extracte fins a 40 agents) on les fases menors es fan visibles.

El pas dels agents (Fig. 7) creix de manera lineal i acaba dominant el cost, cada agent integra a cada tick tota la seva maquinària interna, el graf metabòlic de k reaccions, la xarxa reguladora i l'embolcall, de manera que el cost total és $O(N \cdot k)$ i escala proporcionalment al nombre d'agents. L'entorn en canvi, té un cost gairebé constant (uns 0,42 ms) independent de la població, perquè resol una graella d'escalars de resolució fixa on les col·lisions creixen amb N , i superen l'entorn cap als 900 agents. Tot i així, les dues fases són negligibles davant del pas dels agents gràcies a la malla espacial. En el run principal, portat fins a uns 2 500 agents, el cost per agent es manté pràcticament constant (al voltant de 45 μ s per agent i tick). El llindar de temps real $1\times$ es situa cap als 2 000 agents. El motor s'ha arribat a executar a l'ordre de 25 000 agents, com que el càlcul és monofil, les vies per ampliar aquest límit es comenten a les línies obertes (secció 11).

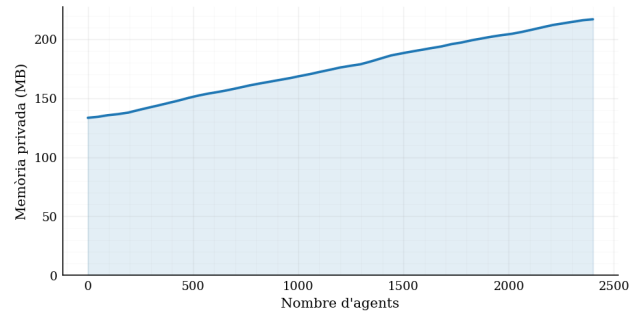


Fig. 8: Memòria privada del procés en funció del nombre d'agents.

La memòria privada (Fig. 8) parteix d'uns 130 MB de base (motor, interfície i textures) i creix lleugerament, uns 35 kB per agent addicional, fins als 220 MB a 2 500 agents, atribuïble a l'estat intern de cada cèl·lula.

9.3 Experiments realitzats

S'han definit escenaris de prova per validar el comportament global de l'eina i comprovar que els components del motor responen de manera coherent davant diferents condicions de simulació.

Nota. Per als experiments descrits a continuació s'ha augmentat deliberadament la taxa de mutació gènica respecte a valors biològicament realistes, per comprimir en una escala de temps simulable (centenars de segons) dinàmiques evolutives que en un cultiu real requereixen molt més temps.

9.3.1 Experiment 1: Dinàmica poblacional sense estrès

Configuració. S'ha executat una simulació de control en un entorn amb glucosa limitada (65 mM per cel·la) i sense antibiòtics, per comprovar que el model reproduceix el comportament esperat d'un cultiu bacterià en condicions favorables. Aquest entorn és la base compartida per la resta d'experiments, de manera que les comparacions posteriors siguin consistents.

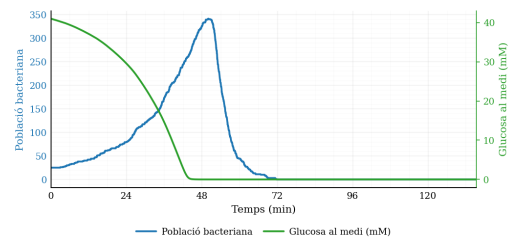


Fig. 9: Experiment 1: evolució de la població bacteriana (eix esquerre) i de la glucosa mitjana al medi (eix dret).

Resultats. La població mostra una fase d'adaptació molt breu, en què regula els nivells d'activació dels seus gens segons els químics que absorbeix, pràcticament imperceptible a causa de la inicialització per defecte del model, seguida d'un creixement exponencial mentre el medi conserva recursos suficients. A mesura que la glucosa s'esgota, la població entra en una fase transitòria d'estabilitat també breu i posteriorment comença el declivi fins la seva desaparició. La corba de glucosa acompanya aquesta interpretació, ja que descendeix de manera logarítmica en ser consumida pels agents durant el creixement.

9.3.2 Experiment 2: Exposició a ampicil·lina

Configuració. Sobre l'escenari base s'ha introduït ampicil·lina al minut 20 en dues concentracions, 100 μM i 65 μM , per estudiar la pressió selectiva de l'antibiòtic i l'aparició de resistència, i es comparen amb el control per visualitzar l'efecte de la dosi.

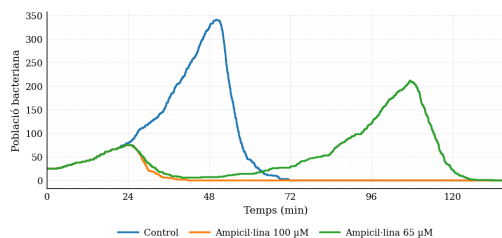


Fig. 10: Experiment 2: evolució de la població bacteriana per a les diferents dosis i control.

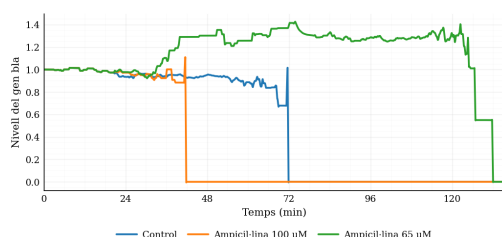


Fig. 11: Experiment 2: nivell mitjà d'expressió del gen *bla* (β -lactamasa TEM-1) per a cada condició.

Resultats. Fins al minut 20, totes les corbes presenten el mateix comportament, ja que res no ha canviat. Tanmateix, davant la presència de l'antibiòtic, la corba de 100 μM mostra una caiguda ràpida de la població fins a la seva eliminació completa, cosa que indica que la pressió antibiòtica supera la capacitat d'adaptació de la colònia. A 65 μM la població també descendeix amb força a l'inici, però sobreviuen uns pocs bacteris amb una expressió més alta de *bla*, que han mutat per atzar en el temps previ. Aquest gen codifica la β -lactamasa TEM-1, que degrada l'ampicil·lina al periplasma abans que assoleixi la seva diana, redueix l'ocupació de les PBP i permet recuperar la síntesi de paret cel·lular. Per això la població resistent acaba recuperant-se i donant lloc a una nova soca resistent. Es pot observar com, al gràfic del gen *bla*, la corba d'ampicil·lina 65 μM augmenta el nivell del gen que caracteritza aquesta resistència.

9.3.3 Experiment 3: Exposició a ciprofloxacina

Configuració. Sobre el mateix escenari base s'ha injectat ciprofloxacina al minut 20. En aquest cas s'analitza la resposta a un antibiòtic que afecta l'ADN i que, a més, incrementa la taxa de mutació, accelerant l'aparició de variants resistents.

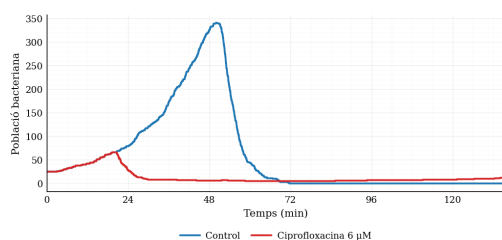


Fig. 12: Experiment 3: evolució de la població bacteriana comparada amb el control.

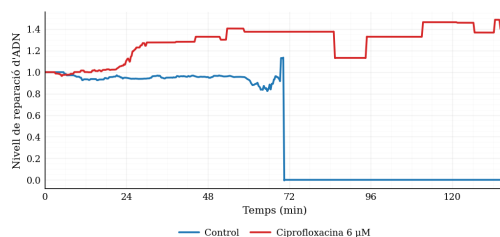


Fig. 13: Experiment 3: nivell mitjà del gen de reparació de l'ADN per a cada condició.

Resultats. La ciprofloxacina provoca una caiguda inicial de la població en augmentar el dany a l'ADN i frenar-ne la reparació. Els bacteris sensibles desapareixen ràpidament, mentre que les poques variants amb més capacitat de reparació sobreviuen i acaben repoblant el sistema de manera lenta. La recuperació és més progressiva que amb l'ampicil·lina perquè aquí l'antibiòtic no només mata, sinó que també compromet la integritat genètica i activa una resposta SOS que penalitza la divisió i fa més costosa l'expansió poblacional. A la simulació, aquesta resposta es modela mitjançant un gen abstracte de reparació de l'ADN, incorporat com a simplificació del sistema real de resposta SOS i manteniment genòmic d'*E. coli*.

10 CONCLUSIONS

El motor de simulació desenvolupat disposa d'un framework i sistema de visualització reutilitzables en altres casos d'ús de simulació basada en agents. És una eina que facilita l'experimentació amb diferents paràmetres i escenaris i ofereix una base extensible per a l'estudi d'organismes biològics i l'anàlisi del comportament del sistema modelat.

La representació de cada bacteri com a agent autònom amb subsistemes interns permet modelar la interacció entre múltiples components i el seu efecte a nivell poblacional, això ens permet entendre millor com evoluciona una població bacteriana davant la presència d'un antibiòtic. A partir de regles locals i d'una abstracció adequada dels processos interns, el model és capaç de reproduir dinàmiques globals coherents, com el creixement poblacional, així com comportaments emergents no definits explícitament, com la resistència davant l'ampicil·lina o la ciprofloxacina quan es configuren adequadament les condicions de l'entorn, els gens i les reaccions. Tanmateix, la capacitat de predir el seu comportament sota diferents condicions queda limitada pel nivell d'abstracció adoptat, ja que la resposta del sistema depèn en gran mesura de la parametrització dels elements modelats i de les simplificacions assumides.

Pel que fa a la planificació, el desenvolupament s'ha seguit en general segons el que estava previst. Tanmateix, en la definició dels passos següents ha estat necessari ajustar l'abast del treball, reduint part de l'ambició inicial per adaptar-se a la complexitat del problema. Això ha implicat, per exemple, centrar-se en el modelatge d'una única espècie bacteriana i limitar algunes funcionalitats.

A nivell personal, aquest projecte m'ha aportat experiència acadèmica i tècnica en l'ús d'eines àmpliament utilitzades en l'àmbit laboral, com OpenGL i ImGui. També m'ha permès comprendre millor el funcionament dels simuladors, especialment els basats en agents, i ha contribuït a agilitzar la meua manera de pensar i d'utilitzar estructures

de dades per representar models computacionals complexos, com en aquest cas un bacteri. A més, és un treball que convida a reflexionar sobre la importància de simplificar els sistemes i d'escollir correctament el nivell d'abstracció a l'hora de modelar qualsevol fenomen de manera computacional.

11 LÍNIES OBERTES

En una continuació del treball, considerem una prioritat les línies següents:

Parametrització formal. Definir i ampliar amb més precisió la parametrització del model, amb unitats, rangs vàlids i justificació biològica per a cada valor, assegurant la coherència amb dades experimentals.

Calibració i validació. Comparar els resultats de simulacions amb més dades experimentals reals, ajustar els paràmetres a partir d'aquestes observacions i validar el model en condicions i escenaris independents. Realitzar anàlisis de sensibilitat per identificar quines variables influeixen més en la dinàmica simulada.

Ampliació de l'abast. Estendre el model a més soques i antibiòtics, incorporar transferència horitzontal de gens mitjançant plasmidis, i introduir heterogeneïtat genètica en la població inicial, mantenint la coherència biològica dels mecanismes modelats.

Escalabilitat. Paral·lelitzar els processos del pas dels agents, independents entre cèl·lules per aprofitar els nuclis disponibles i elevar el llindar de temps real més enllà dels 2.500 agents actuals.

ÚS DE LA IA GENERATIVA

Durant el desenvolupament del treball s'ha utilitzat la intel·ligència artificial generativa com a eina de suport puntual en tasques de programació. Concretament, s'ha empleat per agilitzar la implementació d'alguns elements de la interfície d'usuari del motor, assistir en la resolució d'errors de compilació i depuració en C++, i donar suport a la formulació d'expressions matemàtiques senzilles relacionades amb la simulació. Les decisions finals de disseny, implementació i validació s'han dut a terme manualment.

REFERÈNCIES

- [1] World Health Organization, "Antimicrobial resistance," fitxa informativa. <https://www.who.int/news-room/fact-sheets/detail/antimicrobial-resistance>
- [2] GBD 2021 Antimicrobial Resistance Collaborators, "Global burden of bacterial antimicrobial resistance 1990–2021: a systematic analysis with forecasts to 2050," *The Lancet*, vol. 404, no. 10459, pp. 1199–1226, 2024. [https://doi.org/10.1016/S0140-6736\(24\)01867-1](https://doi.org/10.1016/S0140-6736(24)01867-1)
- [3] B. G. Wong *et al.*, "Precise, automated control of conditions for high-throughput growth of yeast and bacteria with eVOLVER," *Nature Biotechnology*, vol. 36, pp. 614–623, 2018. <https://doi.org/10.1038/nbt.4151>
- [4] J. R. Karr *et al.*, "A whole-cell computational model predicts phenotype from genotype," *Cell*, vol. 150, no. 2, pp. 389–401, 2012. <https://doi.org/10.1016/j.cell.2012.05.044>
- [5] U. Wilensky, "NetLogo," Center for Connected Learning and Computer-Based Modeling, Northwestern University, 1999. <https://ccl.northwestern.edu/netlogo/>
- [6] Project Mesa, "Mesa: Agent-based modeling in Python." <https://mesa.readthedocs.io/>
- [7] T. E. Gorochowski *et al.*, "BSim: an agent-based tool for modeling bacterial populations in systems and synthetic biology," *PLoS ONE*, vol. 7, no. 8, e42790, 2012. <https://doi.org/10.1371/journal.pone.0042790>
- [8] S. Jang *et al.*, "Specification and simulation of synthetic multicelled behaviors," *ACS Synthetic Biology*, vol. 1, no. 8, pp. 365–374, 2012. <https://doi.org/10.1021/sb300034m>
- [9] L. A. Lardon *et al.*, "iDynoMiCS: next-generation individual-based modelling of biofilms," *Environmental Microbiology*, vol. 13, no. 9, pp. 2416–2434, 2011. <https://doi.org/10.1111/j.1462-2920.2011.02414.x>
- [10] M. Campos *et al.*, "A membrane computing simulator of antibiotic resistance evolution dynamics (ARES)," *Biology Direct*, vol. 10, no. 41, 2015. <https://doi.org/10.1186/s13062-015-0070-9>
- [11] A. Ghaffarizadeh *et al.*, "PhysiCell: an open source physics-based cell simulator for 3-D multicellular systems," *PLoS Computational Biology*, vol. 14, no. 2, e1005991, 2018. <https://doi.org/10.1371/journal.pcbi.1005991>
- [12] J. D. Orth, I. Thiele, and B. Ø. Palsson, "What is flux balance analysis?," *Nature Biotechnology*, vol. 28, pp. 245–248, 2010. <https://doi.org/10.1038/nbt.1614>
- [13] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of the Atmospheric Sciences*, vol. 20, no. 2, pp. 130–141, 1963. https://journals.ametsoc.org/view/journals/atsc/20/2/1520-0469_1963_020_0130_dnf_2_0_co_2.xml
- [14] J. M. Monk *et al.*, "iML1515, a knowledgebase that computes *Escherichia coli* traits," *Nature Biotechnology*, vol. 35, pp. 904–908, 2017. <https://doi.org/10.1038/nbt.3956>
- [15] J. M. A. Blair *et al.*, "Molecular mechanisms of antibiotic resistance," *Nature Reviews Microbiology*, vol. 13, pp. 42–51, 2015. <https://doi.org/10.1038/nrmicro3380>
- [16] N. Ruiz España, "Bases de la resistència als antibiòtics en *Serratia marcescens*: contribució de la proteïna OMP1," tesi doctoral, Universitat de Barcelona, 2002.