



**UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA**



**DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE  
LAUREA MAGISTRALE IN BIOINGEGNERIA**

Utilizzo e analisi di "Community Simulator", un software per la  
simulazione di comunità microbiche

**Relatore: Prof.ssa Barbara Di Camillo**

**Laureanda: Ada Rossato**

**Correlatore: Dott. Marco Cappellato**

**ANNO ACCADEMICO 2021-2022**

**Data di laurea 11/04/2022**

# Indice

<b>Introduzione</b>	<b>1</b>
<b>1 Caratteristiche di una comunità microbica</b>	<b>3</b>
1.1 Composizione . . . . .	3
1.2 Interazioni microbiche . . . . .	4
<b>2 Community Simulator</b>	<b>6</b>
2.1 Creazione dello stato iniziale: risorse e abbondanze di batteri . . . . .	8
2.2 Equazioni dinamiche . . . . .	9
2.3 Parametri del sistema . . . . .	15
2.4 Simulazioni . . . . .	19
2.4.1 Soluzione delle equazioni dinamiche . . . . .	19
2.4.2 Trasferimento su una nuova piastra . . . . .	19
2.4.3 Identificazione efficiente dei punti di equilibrio . . . . .	20
<b>3 Implementazione di un metodo per definire le interazioni tra specie</b>	<b>23</b>
<b>4 Simulazione di reti di interazioni microbiche</b>	<b>27</b>
4.1 Inizializzazione dello stato di una comunità microbica . . . . .	28
4.2 Analisi delle reti al variare del tempo . . . . .	29
4.3 Simulazione di repliche biologiche . . . . .	31
4.3.1 Confronto tra stati stazionari . . . . .	32
4.3.2 Analisi delle reti . . . . .	33
<b>Conclusioni</b>	<b>35</b>

# Elenco delle figure

2.1	Istanza della classe <i>Community</i> . . . . .	8
2.2	Flussi energetici generati da una cellula microbica di specie <i>i</i> . . . . .	10
2.3	Pozzetto in modalità chemostato . . . . .	12
2.4	Dinamiche intrinseche risorse . . . . .	12
2.5	Tipi di funzioni di risposta $\sigma$ . . . . .	13
2.6	Regolazione metabolica <i>u</i> . . . . .	13
2.7	Campionamento casuale delle preferenze di consumo . . . . .	17
2.8	Campionamento casuale della matrice metabolica. . . . .	18
3.1	Pesi delle interazioni negative . . . . .	24
3.2	Pesi delle interazioni positive . . . . .	25
4.1	Struttura tassonomica e metabolica della comunità simulata. . . . .	27
4.2	Simulazioni: inizializzazione dello stato. . . . .	29
4.3	Reti di interazioni . . . . .	30
4.4	Confronto reti di interazione nel tempo . . . . .	31
4.5	Confronto tra stati stazionari . . . . .	33
4.6	Confronto tra stati stazionari . . . . .	33
4.7	Confronto reti di interazione tra pozzetti . . . . .	34

# Introduzione

Il lavoro di questa tesi si colloca nell'ambito degli studi sul microbiota, in particolare affronta le complesse interazioni che si instaurano tra i diversi membri di una comunità microbica e che ne determinano la sua organizzazione. Comprendere le relazioni tra le diverse specie microbiche presenti in uno specifico ambiente ha un risvolto significativo su vari aspetti della vita umana, tra cui lo stato stesso di salute dell'uomo. Sono ormai evidenti le enormi potenzialità che tali studi hanno in campo diagnostico e terapeutico. Negli ultimi decenni l'avvento delle tecnologie di *Next-Generation Sequencing* ha permesso di risalire alla composizione interna di una comunità microbica presente in un campione. I cosiddetti *Big Data* prodotti dal processo di sequenziamento possono essere analizzati attraverso innumerevoli metodi statistici e computazionali. In letteratura sono presenti diversi metodi di *Reverse Engineering* che, basandosi su approcci provenienti dalla teoria delle reti, cercano di ricavare la rete di interazioni tra le specie microbiche della comunità. La complessità di tale micro-mondo, costituito da un vasto numero di specie e risorse, non permette di osservare sperimentalmente la realtà biologica e quindi di validare in modo diretto le interazioni ricavate dai diversi metodi. Le simulazioni computazionali possono rappresentare un potente strumento per sopperire a tale mancanza.

L'obiettivo di questo elaborato di tesi è di analizzare e utilizzare un simulatore di comunità microbiche, il *Community Simulator* [1], al fine di simulare una rete di interazioni microbiche che costituirà la realtà di base per valutare le performance dei diversi metodi di *Reverse Engineering* presenti in letteratura. Infatti, il presente lavoro rappresenta il primo step di una pipeline di simulazioni, in cui a partire dalla simulazione di una comunità microbica si arriva alla generazione di dati di sequenziamento sintetici. Attraverso il *Community Simulator* si vuole simulare una rete di interazioni tra specie microbiche, il più realistica possibile, e ottenere dati sull'abbondanza delle specie presenti nella comunità. In futuro tali dati si potranno dare in input ad un secondo simulatore, di nome *metaSPARSim* [2], il quale riproduce i processi di campionamento e successivo sequenziamento. Infine, dai dati di sequenziamento sintetici, si applicheranno alcuni dei metodi di *Reverse Engineering*. Attraverso un'analisi comparativa, tra le diverse reti da essi ottenute e la rete 'ground truth' di partenza, si potranno valutarne le performance.

La tesi è composta da quattro capitoli.

Nel primo capitolo si introduce brevemente il complesso micro-mondo delle comunità microbiche.

Nel secondo capitolo si analizza uno specifico simulatore di comunità microbiche, il *Community Simulator*.

Nel terzo capitolo, a partire da alcuni parametri del modello matematico implementato nel simulatore, vengono definite matematicamente le interazioni tra le diverse specie microbiche di una comunità simulata. Tali definizioni sono state implementate in un metodo che permette di simulare delle reti di interazioni microbiche.

Nel quarto capitolo si riportano e analizzano le simulazioni effettuate utilizzando il simulatore e il nuovo metodo implementato.

# 1. Caratteristiche di una comunità microbica

I microorganismi (batteri, virus, funghi, protozoi e alghe) popolano quasi tutte le nicchie ecologiche concepibili e rappresentano ampiamente la maggior parte della diversità genetica e metabolica del pianeta [3]. Per questo hanno un impatto cruciale sulla vita dell'uomo.

Uno specifico ambiente ed ecosistema può essere caratterizzato da più comunità microbiche, ciascuna costituita da un insieme di diverse popolazioni di microorganismi della stessa specie. Una comunità batterica è un complesso micro-mondo, caratterizzato da innumerevoli interazioni che ne determinano la sua organizzazione. Principalmente si possono distinguere due tipi di interazioni: microbiche ed ecologiche [4]. Le interazioni ecologiche si instaurano tra le specie e l'ambiente, mentre quelle microbiche tra le diverse specie presenti nell'ambiente. In una comunità i microorganismi di specie diverse non si distribuiscono omogeneamente, ma la loro disposizione spaziale dipende dalle interazioni con le cellule vicine e dall'ambiente abiotico. Essi colonizzano gli ambienti in base al loro fabbisogno metabolico e fisiologico [5]

Informazioni sulla rete di interazioni microbiche e sulla conseguente composizione di una complessa comunità microbica, sono estratte principalmente da dati di sequenziamento. Il recente sviluppo di tecnologie di *Next-Generation Sequencing* ha permesso di ottenere dati ad alta risoluzione temporale, in grado di seguire l'evoluzione di un'intera comunità microbica presente in un campione, per un periodo sufficientemente lungo. Ciò ha favorito l'affermarsi dei metodi di *Reverse Engineering*, i quali per inferire la rete di interazioni utilizzano dati di abbondanza di specie relativi a finestre temporali.

## 1.1 Composizione

Con il termine microbiota si intende la composizione dell'insieme di microorganismi associato ad un determinato ambiente ed ecosistema. L'analisi del corrispondente microbioma (genoma), attraverso piattaforme di *Next-Generation Sequencing*, porta all'identificazione precisa delle specie presenti in un campione. Le due principali tecniche di sequenziamento utilizzate per ricavare informazioni sull'abbondanza delle specie sono la *Whole Genome Sequencing (WGS)* e la tecnica di sequenziamento cosiddetta *targeted* del gene 16S rRNA (*16S rDNA-seq*). Quest'ultima risulta

la più utilizzata ed è basata sul sequenziamento di regioni iper-variabili del gene 16S rRNA batterico, che permettono di discriminare e identificare i diversi membri della comunità presente nel campione [4]. Attraverso successive analisi è possibile arrivare alla descrizione tassonomica del microbiota a diversi livelli, tra cui i principali utilizzati sono: Phylum, Family e Genus/Species.

Il microbiota umano è costituito da 10-100 trilioni di cellule microbiche, che vivono in simbiosi con l'uomo, principalmente nel tratto gastro intestinale [6]. Nel caso del microbiota intestinale, vi è un'elevata diversità di specie, tuttavia analizzandolo ad alti livelli filogenetici si riescono ad individuare due phyla principali, Firmicutes e Bacteroidetes, a cui appartengono il 90% dei microbi dell'ecosistema intestinale. A livello di famiglia, invece, si caratterizza principalmente per la presenza di *Lachnospiraceae*, *Ruminococcaceae* e *Bacteroidaceae*. A tali famiglie appartengono microorganismi che producono acidi grassi a corta catena (SCFA), come l'acetato, il propionato e il butirato, metaboliti dovuti alla fermentazione di polisaccaridi complessi e che hanno un ruolo fondamentale nel mantenimento di un stato fisiologico [7].

## 1.2 Interazioni microbiche

In una comunità microbica si instaurano differenti tipi di interazioni tra le diverse specie presenti. Le tipologie di interazioni sono definite in base all'effetto positivo (+), negativo (-) o neutro (0), apportato a ciascuna specie coinvolta nell'interazione di coppia:

- (0, 0) Neutralismo. Tra le due specie non vi è alcuna relazione o essa risulta trascurabile.
- (+, +) Mutualismo. L'interazione comporta un effetto benefico per entrambe le specie. Questo avviene quando le specie sono legate da rapporti di sintrofismo o alimentazione incrociata, in cui la sussistenza di un microorganismo dipende dai sottoprodotti metabolici dell'altro e viceversa. Un altro caso è quello in cui le specie cooperano per svolgere una comune funzione fisiologica.
- (+, 0) Commensalismo. Una specie trae beneficio dalla presenza di un'altra senza apportare effetti su di essa.
- (-, -) Competizione. Le due specie coinvolte nell'interazione competono per nutrienti in comune, influenzandosi negativamente. Tale interazione può portare all'estinzione della specie più debole o alla sua migrazione verso una nicchia ecologica diversa.

- (-, 0) Ammensalismo. Una specie danneggia un'altra senza trarne beneficio o danno. In alcuni casi i prodotti metabolici di una specie rendono l'ambiente sfavorevole per un'altra.
- (+, -) Parassitismo e Predazione. Una specie trae beneficio a spese di un'altra. Nella relazione di parassitismo la specie parassita sopravvive grazie ad un ospite, come nel caso dei batteriofagi che si moltiplicano all'interno di ospiti batterici. Nella predazione un microorganismo si nutre di un altro, come nel caso del *Bdellovibrio bacteriovorus*, il quale si nutre di altri batteri.

Sebbene i diversi approcci, usualmente utilizzati per studiare tali interazioni, cerchino di definire interazioni statiche, in realtà, esse sono dinamiche e dipendenti dal contesto [8]. Le interazioni tra i diversi membri di una comunità microbica possono cambiare in base alle condizioni ambientali (abiotiche e biotiche), allo spazio e dipendere dal tempo [8]. A seconda del contesto, i microbi possono modificare la loro espressione genica. In questo modo, per esempio, alcuni microbi intestinali regolano il loro consumo di risorse in base alla disponibilità nell'ambiente [8]. Inoltre, la struttura della comunità può influire sulle interazioni microbiche, una densità spaziale bassa può rendere difficile l'incontro tra microbi e quindi poco probabile l'instaurarsi di una relazione, mentre in ambienti spazialmente limitati può prevalere una forte competizione [8].

Lo studio delle reti di interazioni potrebbe avere risvolti importanti in applicazioni terapeutiche che cerchino di manipolare il microbiota umano, intervenendo per esempio in situazioni di disbiosi tramite l'uso di probiotici. Infatti, un probiotico per essere efficace dovrà almeno riuscire a colonizzare il suo ospite e ciò dipende strettamente dalle interazioni che instaura con il microbiota naturale [8]. Comprendere come le interazioni vengano modificate a causa dell'introduzione di uno stimolo, potrebbe permettere di intervenire attraverso probiotici per ripristinare uno stato fisiologico (eubiosi) [9].



## 2. Community Simulator

Il *Community Simulator* [1] è un pacchetto Python open-source, sviluppato per facilitare le simulazioni delle complesse comunità microbiche in diversi scenari sperimentali rilevanti. Il pacchetto permette di simulare le dinamiche delle popolazioni microbiche in modo “riproducibile, trasparente e scalabile”[1].

Gli ecosistemi microbici sono caratterizzati da centinaia a migliaia di specie interagenti e da centinaia di metaboliti; si basano su un'alimentazione incrociata e su rapporti sintrofici tra le specie. Per questi motivi i modelli classici di ecologia comunitaria, basati su una dimensionalità ridotta e su una rigida separazione dello strato trofico, risultano inadeguati al loro trattamento. Il pacchetto supera tali problematiche implementando un avanzato modello minimo di ecosistema microbico, il *Microbial Consumer Resource Model (MicroCRM)* [1, 10–12]. Esso è costituito da coppie di equazioni differenziali ordinarie, di complessità inferiore rispetto alla maggior parte dei modelli di ecosistema, ed è quindi in grado di limitare l'onerosità computazionale tipica di tali simulazioni. La modellizzazione di un ecosistema microbico, a causa dell'elevato numero di specie e tipi di risorse, richiede un ampio numero di parametri, i quali devono essere scelti in modo biologicamente realistico. Tale difficoltà viene affrontata campionando i parametri in modo casuale da distribuzioni di probabilità ben definite, che ne garantiscono il significato fisico. Questo approccio ricorre alle teorie della meccanica statistica e delle matrici casuali, basandosi sull'osservazione che sistemi complessi, quali sono gli ecosistemi microbici, hanno comportamenti generici che possono essere ben descritti da ecosistemi casuali [10]. Il *MicroCRM* permette di incorporare, oltre alla competizione per le risorse e al 'cross-feeding', anche processi ecologici come la colonizzazione stocastica, importanti nel determinare la struttura e le dinamiche di una comunità microbica. Il *Community Simulator* velocizza le simulazioni di più comunità microbiche sfruttando l'integrazione in parallelo delle equazioni dinamiche (incompatibile con Windows). Un ulteriore e significativo aumento delle prestazioni viene ottenuto grazie all'implementazione di un nuovo algoritmo di *Expectation-Maximization (EM)*, che permette di trovare gli stati stazionari delle dinamiche di comunità senza l'integrazione diretta.

L'architettura del *Community Simulator*, ispirandosi agli esperimenti in parallelo solitamente eseguiti con piastre a 96 pozzetti, permette di simulare in parallelo le dinamiche di più comunità. Il fulcro del pacchetto è la classe `Community`, una cui istanza è una piastra a  $n$  pozzetti, contenenti  $n$  comunità microbiche ben miscelate e non interagenti (Figura 2.1). L'inizializzazione di una istanza avviene specificandone:

- lo stato iniziale, ossia le dimensioni iniziali della popolazione microbica e le concentrazioni delle risorse in ciascun pozzetto.
- le equazioni che descrivono le dinamiche delle popolazioni e delle risorse, secondo il modello matematico di ecosistema adottato.
- i valori dei parametri delle equazioni dinamiche.

I metodi di istanza applicabili alla piastra virtuale, una volta che è stata inizializzata, sono principalmente tre; solo i primi due aggiornano lo stato corrente:

- `Community.Propagate(T)`, propaga il sistema per un tempo  $T$  integrando le equazioni dinamiche .
- `Community.Passage(f)`, inizializza una nuova piastra trasferendo determinate frazioni (indicate nella matrice  $f$ ) del contenuto di ciascun vecchio pozzetto a ciascun nuovo pozzetto.
- `Community.SteadyState()`, trova i punti di equilibrio delle dinamiche del sistema in modo efficiente, senza l'integrazione numerica.

Se si usa il *Microbial Consumer Resource Model*, lo stato iniziale, la legge dinamica e i parametri, possono essere generati automaticamente sulla base di un dizionario di assunzioni. Il pacchetto permette all'utente di impostare manualmente tali assunzioni o di utilizzare quelle di default contenute nel dizionario `a_default`.

Nelle sezioni successive verranno affrontate le funzioni atte a generare automaticamente i tre elementi chiave per l'inizializzazione di un'istanza e si approfondiranno i tre metodi.

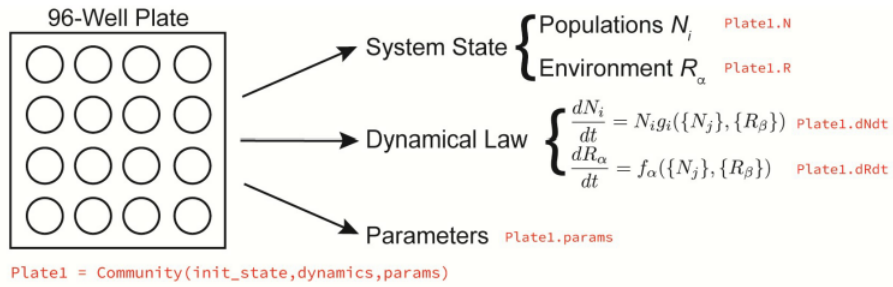


Figura 2.1: Un'istanza della classe `Community` è una piastra a  $n$  pozzetti, i cui attributi sono: lo stato corrente, le equazioni e i parametri che descrivono le dinamiche delle popolazioni e delle risorse. Immagine tratta da [1].

## 2.1 Creazione dello stato iniziale: risorse e abbondanze di batteri

Lo stato di una piastra virtuale a  $n$  pozzetti è rappresentato da due data frame Pandas. Essi descrivono la composizione, sia qualitativa che quantitativa, della comunità microbica e dell'ambiente in ciascun pozzetto, in un determinato istante; uno dei due data frame ha dimensione  $S_{tot} \times n$  e contiene le dimensioni della popolazione microbica  $N_i$ , per ogni specie  $i$  ( $i=1, 2, \dots, S_{tot}$ , con  $S_{tot}$  numero totale di specie considerate) e per ogni pozzetto; analogamente l'altro ha dimensione  $M \times n$  e contiene le concentrazioni  $R_\alpha$  di ciascuna risorsa  $\alpha$  ( $\alpha=1, 2, \dots, M$ , con  $M$  numero dei diversi tipi di risorse), per ogni pozzetto.

La coppia di data frame `N0` e `R0`, contenenti lo stato iniziale, può essere generata automaticamente con la funzione `MakeInitialState`. Questa funzione, per ogni pozzetto, campiona un sottoinsieme di  $S$  specie da un insieme complessivo di  $S_{tot}$  specie, in modo totalmente casuale. Le dimensioni di popolazione delle  $S$  specie selezionate vengono poste a 1, indicando invece con lo 0 l'assenza nel pozzetto delle restanti specie. Le dimensioni possono successivamente essere riscalate. In tale modo, viene simulato l'effetto della colonizzazione stocastica, processo dovuto a diversi meccanismi dispersivi.

Le concentrazioni iniziali delle diverse risorse vengono impostate secondo uno scenario sperimentale di piastra *Biolog*, in cui l'ambiente di ciascun pozzetto viene inizialmente rifornito da una sola fonte di carbonio. Quindi, `MakeInitialState` permette di variare la quantità e il tipo di risorsa fornita nei diversi pozzetti, ma se ad un pozzetto si vuole fornire più di una risorsa, bisogna creare `R0` manualmente.

Inoltre, un'importante scelta progettuale, ai fini di introdurre un minimo di struttura metabolica, è quella di dividere le  $M$  risorse e le  $S_{tot}$  specie in specifiche classi. Ossia, le risorse possono essere suddivise in  $T$  classi (per es. zuccheri, lipidi, ecc.), ciascuna contenente  $M_A$  tipi di risorsa, con  $A = 1, \dots, T$  e  $\sum_A M_A = M$ . Le specie possono essere assegnate a  $F$  famiglie, con  $F \leq T$ , ciascuna corrispondente a specie che consumano preferenzialmente risorse di una determinata classe. In aggiunta, si può considerare la presenza di una famiglia generalista che include le  $S_{gen}$  specie con nessuna classe di risorse preferenziale, in modo che  $S_{gen} + \sum_A S_A = S_{tot}$ . Questa suddivisione in classi avrà un impatto nel campionamento dei parametri fondamentali per descrivere il metabolismo delle specie, quali le preferenze di consumo e i coefficienti stechiometrici dei sottoprodotti metabolici.

## 2.2 Equazioni dinamiche

Il pacchetto consente di utilizzare qualunque legge dinamica che sia espressa da coppie di equazioni differenziali nella forma seguente:

$$\frac{dN_i}{dt} = N_i g_i(\{N_j\}, \{R_\beta\})$$

$$\frac{dR_\alpha}{dt} = f_\alpha(\{N_j\}, \{R_\beta\})$$

Ossia, le derivate nel tempo,  $\frac{dN_i}{dt}$  e  $\frac{dR_\alpha}{dt}$ , sono funzioni dello stato del sistema e possono essere customizzate dall'utente, assegnando una forma analitica alle generiche funzioni qui indicate con  $g_i(\{N_j\}, \{R_\beta\})$  e  $f_\alpha(\{N_j\}, \{R_\beta\})$ .

Le funzioni `MakeConsumerDynamics` e `MakeResourceDynamics` costruiscono automaticamente le equazioni dinamiche utilizzando il *Microbial Consumer Resource Model (MicroCRM)* [1, 10, 11]. Tale modello adatta il classico *Consumer Resource Model* di MacArthur e Levins al contesto delle comunità microbiche, permettendo lo scambio e il consumo di sottoprodotti metabolici tra gli organismi. Il *MicroCRM* è un modello termodinamico che garantisce la conservazione dell'energia, grazie all'assunzione che gli  $M$  tipi di risorse siano sostituibili e possano essere convertiti in una valuta energetica comune. Questa assunzione ha come naturale contesto di validità, quello in cui tutti i componenti essenziali della biomassa sono forniti in eccesso, in modo che la crescita dipenda unicamente dal rifornimento di energia utilizzabile e nel modello si possano quindi esplicitare solamente le risorse da cui i microorganismi riescono a trarre energia. Tutti gli altri nutrienti e

i prodotti di scarto da cui non è più ricavabile energia, non sono inclusi nelle  $M$  risorse, ma vengono trattati implicitamente nel modello; per esempio assumendo che una parte di energia assorbita venga utilizzata dal microorganismo per rifornirsi dei composti necessari alla sua crescita.

La sostituibilità delle risorse, inoltre, permette di non dover entrare troppo nel dettaglio di come le risorse interagiscono tra loro attraverso il metabolismo del consumatore.

Il *MicroCRM* descrive le dinamiche e il metabolismo delle specie in termini di flussi energetici mediati dal consumo, dalla secrezione e dalla trasformazione chimica di  $M$  risorse sostituibili, da parte di ciascun microbo. Tali flussi dipendono dalla specie microbica considerata e per questo, in seguito, li indicheremo sempre con un indice  $i$  indicativo della specie. Ciascuna cellula microbica di specie  $i$  si rifornisce energeticamente attraverso il consumo di risorse, generando per ogni risorsa  $\alpha$  un flusso energetico  $J_{i\alpha}^{in}$  in ingresso alla cellula stessa. Una frazione  $l_\alpha$  di questa energia viene rilasciata nell'ambiente attraverso sottoprodotti metabolici  $\beta$ , mentre l'energia rimanente,  $J_{i\alpha}^{grow}$ , è utilizzata per la crescita e la replicazione della cellula. In questo modo si garantisce la conservazione dell'energia:

$$J_i^{in} = J_i^{grow} + J_i^{out} \quad (2.1)$$

dove,  $J_i^{in} = \sum_\alpha J_{i\alpha}^{in}$ ,  $J_i^{out} = \sum_\alpha l_\alpha J_{i\alpha}^{in}$  e  $J_i^{grow} = \sum_\alpha J_{i\alpha}^{grow}$  sono i flussi energetici complessivi, rispettivamente in ingresso a una cellula di specie  $i$ , in uscita da essa e quello usato per la sua crescita.

Per ogni sottoprodotto metabolico espulso  $\beta$ , si ha un flusso in uscita  $J_{i\beta}^{out}$  espresso nella forma

$$J_{i\beta}^{out} = \sum_\alpha D_{\beta\alpha} l_\alpha J_{i\alpha}^{in} \quad (2.2)$$

dove con  $D_{\beta\alpha}$  si indica la frazione di energia in uscita che ritorna nell'ambiente sotto forma di metabolita  $\beta$  quando una cellula consuma la risorsa  $\alpha$ . Per definizione si ha quindi  $\sum_\beta D_{\beta\alpha} = 1$ . In figura 2.2 si riporta uno schema dei flussi energetici sopra discussi.

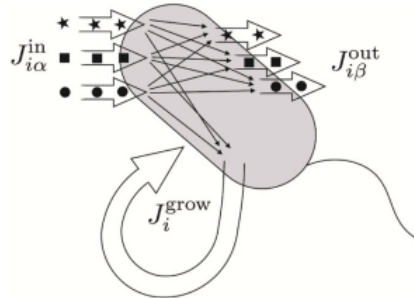


Figura 2.2: Flussi energetici generati da una cellula microbica di specie  $i$ . Immagine tratta da [1].

Dai flussi energetici si può passare ai corrispondenti flussi di massa:

$$v_{i\alpha}^{in} = J_{i\alpha}^{in}/w_{\alpha} \quad v_{i\alpha}^{grow} = J_{i\alpha}^{grow}/w_{\alpha} \quad v_{i\beta}^{out} = J_{i\beta}^{out}/w_{\beta} \quad (2.3)$$

con  $w_{\alpha}$  e  $w_{\beta}$  si indica la densità di energia della risorsa  $\alpha$  in ingresso e della risorsa  $\beta$  in uscita, ma tali indici sono intercambiabili, in quanto rappresentano la densità dello stesso insieme di risorse  $M$ .

L'evoluzione nel tempo dell'abbondanza dei consumatori  $N_i$  appartenenti alla specie  $i$  e della concentrazione di risorse  $\alpha$ ,  $R_{\alpha}$ , sono modellate dalle equazioni differenziali

$$\frac{dN_i}{dt} = g_i N_i [J_i^{grow} - m_i] \quad (2.4)$$

$$\frac{dR_{\alpha}}{dt} = h_{\alpha}(R_{\alpha}) + \sum_i N_i v_{i\alpha}^{out} - \sum_i N_i v_{i\alpha}^{in} \quad (2.5)$$

L'equazione (2.4) deriva dall'assunzione che, per ogni cellula, una minima quantità di flusso energetico  $m_i$  sia necessaria per mantenere stabile la popolazione della specie  $i$  e che il tasso di crescita di una cellula ( $\frac{dN_i}{dt} \frac{1}{N_i}$ ) sia proporzionale al restante flusso energetico ( $J_i^{grow} - m_i$ ), con costante di proporzionalità  $g_i$ .

Le dinamiche di risorsa (2.5) sono descritte essenzialmente da due termini: uno, dipende dai flussi di massa in ingresso  $v_{i\alpha}^{in}$  e in uscita  $v_{i\alpha}^{out}$  di ciascun consumatore presente nell'ambiente (tali flussi, infatti, sottraggono e riforniscono l'ambiente di risorse); l'altro,  $h_{\alpha}(R_{\alpha})$ , descrive le dinamiche intrinseche della risorsa  $\alpha$ , ossia le dinamiche in assenza di consumatori. Il *Community Simulator* permette di impostare tre possibili dinamiche  $h_{\alpha}(R_{\alpha})$  in base alla scelta di come le risorse vengano fornite ai pozzetti:

- $h_{\alpha}^{external}(R_{\alpha}) = \tau_{\alpha}^{-1}(R_{\alpha}^0 - R_{\alpha})$ , quando il rifornimento avviene dall'esterno con un flusso costante  $\Sigma_{\alpha} \tau_{\alpha}^{-1} R_{\alpha}^0$  di specifiche risorse in ingresso al pozzetto e tutte le risorse sono diluite ad una velocità  $\tau_{\alpha}^{-1}$ . Tale dinamica viene adottata per simulare colture in chemostato (si veda figura 2.3).
- $h_{\alpha}^{self-renewing}(R_{\alpha}) = r_{\alpha} R_{\alpha}(R_{\alpha}^0 - R_{\alpha})$ , quando le risorse si autorigenerano (si veda figura 2.4 per un confronto tra i due tipi di dinamiche).
- $h_{\alpha}^{off}(R_{\alpha}) = 0$ , quando le risorse vengono fornite solo inizialmente e si vuole quindi simulare l'esaurimento delle risorse. Questa impostazione viene utilizzata nelle simulazioni di colture batch.

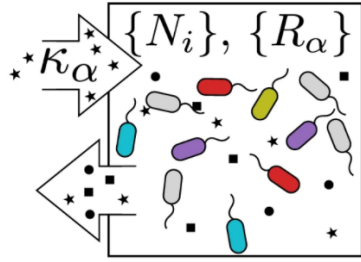


Figura 2.3: Pozzetto in modalità chemostato, ossia viene rifornito costantemente da un flusso di risorse di tipo  $\alpha$  e tutte le risorse sono diluite ad una fissata velocità. Immagine tratta da [10]

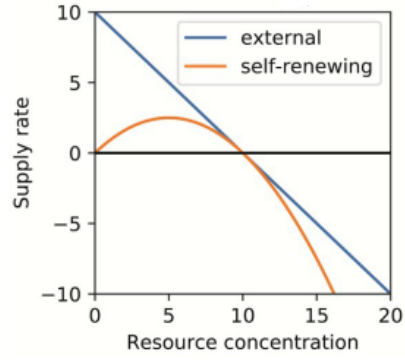


Figura 2.4: Modello lineare e modello logistico per le dinamiche intrinseche delle risorse; rispettivamente corrispondenti ad un rifornimento di risorse esterno e ad un auto-rinnovamento delle risorse. Immagine tratta da [1]

Si affrontano ora nel dettaglio le espressioni analitiche dei flussi di massa. Un individuo di specie  $i$  assorbe un tipo di risorsa  $\alpha$  con una velocità che dipende, sia dalla concentrazione  $R_\alpha$  della risorsa, che dalla preferenza di consumo  $c_{i\alpha}$  della sua specie per tale risorsa  $\alpha$ . Questa dipendenza viene implementata nell'espressione del flusso di massa in ingresso  $v_{i\alpha}^{in}$ :

$$v_{i\alpha}^{in} = u_{i\alpha}^{in} \sigma(c_{i\alpha} R_\alpha) \quad (2.6)$$

dove il termine  $u_{i\alpha}^{in}$  può essere impostato a 1, in modo che le velocità di consumo dei diversi tipi di risorse siano indipendenti tra loro, oppure può rappresentare una forma di regolazione metabolica che favorisce il consumo di tipi risorse con una fruibilità di energia maggiore (Figura 2.6).

Quindi, nel caso in cui non vi sia regolazione metabolica si ha

$$v_{i\alpha}^{in} = \sigma(c_{i\alpha} R_\alpha)$$

dove  $\sigma(x)$  rappresenta la funzione di risposta, ossia specifica la relazione tra il tasso di consumo delle risorse e la loro concentrazione. Di conseguenza, determina anche il tipo di relazione tra il tasso di crescita pro-capite e la concentrazione delle risorse.

Il simulatore implementa tre tipi di funzioni di risposta (Figura 2.5):

- Tipo-I, funzione lineare:

$$\sigma_I(x) = x$$

- Tipo-II, funzione di Monod che va a saturazione all'aumentare della concentrazione di risorse:

$$\sigma_{II}(x) = \frac{x}{1 + \frac{x}{\sigma_{max}}}$$

- Tipo-III, funzione di Hill con un andamento sigmoidale che va a saturazione:

$$\sigma_{II}(x) = \frac{x^n}{1 + \frac{x^n}{\sigma_{max}}}$$

$\sigma_{max}$  rappresenta il massimo flusso di massa in ingresso, nelle funzioni di tipo II e III;  $n$  è il coefficiente di Hill che regola la pendenza della risposta funzionale di tipo III ( $n > 1$ ).

Anche per la regolazione metabolica si hanno due opzioni di scelta:

- regolazione basata sulla frazione di energia utilizzabile contenuta nella risorsa  $\alpha$  :

$$u_{i\alpha}^{in} = \frac{(w_{\alpha}c_{i\alpha}R_{\alpha})^{n_{reg}}}{\sum_{\beta}(w_{\beta}c_{i\beta}R_{\beta})^{n_{reg}}}$$

- regolazione basata sulla frazione di biomassa contenuta nella risorsa  $\alpha$ :

$$u_{i\alpha}^{in} = \frac{(c_{i\alpha}R_{\alpha})^{n_{reg}}}{\sum_{\beta}(c_{i\beta}R_{\beta})^{n_{reg}}}$$

$n_{reg}$  rappresenta il coefficiente di Hill per la regolazione metabolica.

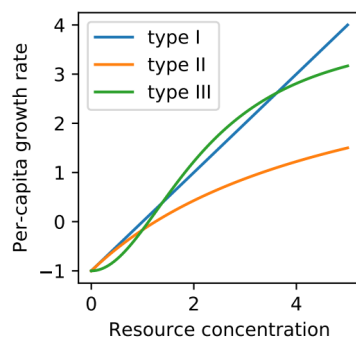


Figura 2.5: Tipi di funzioni di risposta  $\sigma$

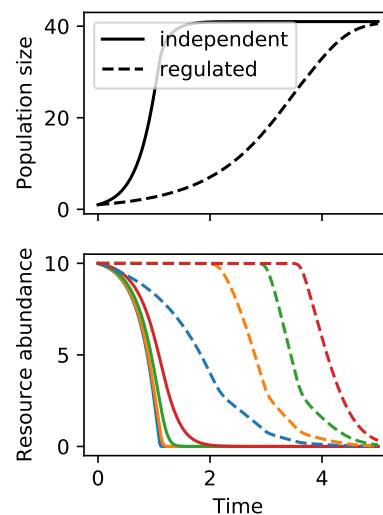


Figura 2.6: Regolazione metabolica  $u$



Per la conservazione dell'energia (Eq.2.1), esplicitando  $J_i^{in}$  (con Eq. 2.6 e 2.3) si ha

$$J_i^{grow} = \sum_{\alpha} (1 - l_{\alpha}) J_{i\alpha}^{in} = \sum_{\alpha} (1 - l_{\alpha}) w_{\alpha} u_{i\alpha}^{in} \sigma(c_{i\alpha} R_{\alpha}) \quad (2.7)$$

Dall'espressione di  $J_{i\beta}^{out}$  (Eq. 2.2), passando al flusso di massa (Eq. 2.3) ed esplicitando  $v_{i\alpha}^{in}$  (Eq. 2.6) si ha

$$v_{i\beta}^{out} = \sum_{\alpha} D_{\beta\alpha} l_{\alpha} \frac{w_{\alpha}}{w_{\beta}} u_{i\alpha}^{in} \sigma(c_{i\alpha} R_{\alpha}) \quad (2.8)$$

Alla fine, esplicitando le equazioni dinamiche (Eq. 2.4 e 2.5) con le equazioni 2.7 e 2.8, si ottiene:

$$\begin{aligned} \frac{dN_i}{dt} &= g_i N_i \left[ \sum_{\alpha} (1 - l_{\alpha}) w_{\alpha} u_{i\alpha}^{in} \sigma(c_{i\alpha} R_{\alpha}) - m_i \right] \\ \frac{dR_{\alpha}}{dt} &= h_{\alpha}(R_{\alpha}) + \sum_{i\beta} N_i u_{i\beta}^{in} \sigma(c_{i\beta} R_{\beta}) \left[ D_{\alpha\beta} l_{\beta} \frac{w_{\beta}}{w_{\alpha}} \right] - \sum_i N_i u_{i\alpha}^{in} \sigma(c_{i\alpha} R_{\alpha}) \end{aligned}$$

Queste equazioni vengono generate automaticamente specificando la modalità di rifornimento delle risorse  $h$ , la funzione di risposta  $\sigma$  e la modalità di regolazione  $u$  dei flussi in ingresso ai consumatori.

Tabella 2.1: Parametri e unità di misura del MicroCRM

$N_i$	densità di popolazione della specie $i$ (individui/volume)
$R_{\alpha}$	concentrazione della risorsa $\alpha$ (massa/volume)
$c_{i\alpha}$	velocità di consumo della specie $i$ , per concentrazione unitaria di risorsa $\alpha$ (volume/tempo)
$D_{\alpha\beta}$	frazione di sottoprodotti dovuti al consumo di $\beta$ e convertita in $\alpha$ (adimensionale)
$g_i$	fattore di conversione da assorbimento energetico a tasso di crescita, per specie $i$ (1/energia)
$m_i$	minimo assorbimento di energia per il mantenimento della specie $i$ (energia/tempo)
$w_{\alpha}$	densità energetica della risorsa $\alpha$ (energia/massa)
$l_{\alpha}$	frazione di perdita per risorsa $\alpha$ (adimensionale)
$R_{\alpha}^0$	concentrazione di equilibrio per la dinamica intrinseca della risorsa $\alpha$ (massa/volume)
$\tau_{\alpha}$	scala temporale per il ricambio delle risorse fornite dall'esterno (tempo)
$r_{\alpha}$	tasso di auto-rinnovo delle risorse (volume/massa/tempo)
$n$	coefficiente di Hill per risposta funzionale di tipo III (adimensionale)
$\sigma_{max}$	flusso massimo in ingresso ad una cellula (massa/tempo)
$n_{reg}$	coefficiente di Hill per regolazione metabolica (adimensionale)

## 2.3 Parametri del sistema

La complessità delle comunità microbiche, formate da un elevato numero di specie e tipi di risorse, porta ad affrontare il problema di come settare gli innumerevoli parametri del *MicroCRM*. Il problema si pone, in particolare, per le preferenze dei consumatori  $c_{i\alpha}$  e i coefficienti stechiometrici dei sottoprodotti metabolici  $D_{\beta\alpha}$ , che compaiono nell'espressioni dei flussi energetici in ingresso e in uscita a/da ciascun consumatore di specie  $i$ , attraverso ciascun tipo di risorsa  $\alpha/\beta$ :

$$J_{i\alpha}^{in} = w_{\alpha} u_{i\alpha}^{in} \sigma(c_{i\alpha} R_{\alpha})$$

$$J_{i\beta}^{out} = \sum_{\alpha} D_{\beta\alpha} l_{\alpha} J_{i\alpha}^{in}$$

con  $i = 1, \dots, S_{tot}$  e  $\alpha = 1, \dots, M$ . La funzione `MakeMatrices` permette di creare la matrice delle preferenze di consumo  $c_{i\alpha}$  ( $S_{tot} \times M$ ) e la matrice metabolica  $D_{\beta\alpha}$  ( $M \times M$ ), campionando i coefficienti da specifiche distribuzioni di probabilità (Figura 2.7 e 2.7). Questo approccio permette di gestire un numero ridotto di parametri, ossia quei parametri globali necessari al campionamento, e nel contempo, di garantire un significato fisico ai coefficienti. Inoltre, tramite il controllo di pochi parametri, permette di regolare quanta struttura tassonomica e metabolica introdurre nella comunità microbica simulata.

**Matrice delle preferenze di consumo  $c_{i\alpha}$ .** Ciascuna riga della matrice indica il livello di preferenza di una diversa specie microbica per ciascun tipo di risorsa. La suddivisione delle specie in  $F$  famiglie, ciascuna specializzata nel consumo di una classe di risorse diversa, consente di introdurre delle correlazioni nelle preferenze di consumo delle specie appartenenti alla stessa famiglia. I coefficienti di consumo di ciascuna famiglia possono essere campionati da una delle seguenti quattro distribuzioni di probabilità:

- Gaussiana
- Gamma
- Bernulli
- Uniforme

I parametri fondamentali per la costruzione di tutte queste distribuzioni sono la media  $\mu_c$  e la varianza  $\sigma_c$  della capacità di consumo totale  $\sum_{\alpha} c_{i\alpha}$  (somma dei coefficienti su una riga della matrice).

I modelli, con matrice delle preferenze di consumo Gaussiana e Gamma, sono parametrizzati da  $\mu_c$  e  $\sigma_c$ , e da un parametro  $q$ , che rappresenta la frazione di capacità di consumo che le famiglie specializzate allocano alla loro classe di risorse preferita. Indicando con  $c_{i\alpha}^A$  i coefficienti di consumo per una famiglia specializzata, con classe di risorse preferenziale A, e con  $c_{i\alpha}^{gen}$  i coefficienti per la famiglia generalista, il campionamento avviene da una distribuzione con media e varianza

$$\langle c_{i\alpha}^A \rangle = \begin{cases} \frac{\mu_c}{M} \left[ 1 + \frac{M-M_A}{M_A} q \right], & \text{se } \alpha \in A \\ \frac{\mu_c}{Mc_1} (1 - q), & \text{altrimenti,} \end{cases}$$

$$\langle (\delta c_{i\alpha}^A)^2 \rangle = \begin{cases} \frac{\sigma_c^2}{M} \left[ 1 + \frac{M-M_A}{M_A} q \right], & \text{se } \alpha \in A \\ \frac{\sigma_c^2}{Mc_1} (1 - q), & \text{altrimenti} \end{cases}$$

per le famiglie specializzate;  $M_A$  è il numero di risorse appartenenti alla classe A. In questo modo, la media e la varianza dei coefficienti sono maggiori per le risorse appartenenti alla classe di preferenza e, impostando  $q = 1$ , si possono ottenere famiglie specializzate che consumano solo risorse della loro classe. Nella famiglia generalista, la media e la varianza sono le stesse per tutte le risorse:

$$\langle c_{i\alpha}^{gen} \rangle = \frac{\mu_c}{M}$$

$$\langle (\delta c_{i\alpha}^{gen})^2 \rangle = \frac{\sigma_c^2}{M}$$

La distribuzione Gamma garantisce che tutti i coefficienti siano positivi.

Il modello binario ha solo due possibili valori per ogni  $c_{i\alpha}$ : un livello di consumo basso  $\frac{c_0}{M}$  e uno alto  $\frac{c_0}{M} + c_1$ . Tale modello è parametrizzato da  $\mu_c$ ,  $q$  e dai due parametri  $c_0$  e  $c_1$ , che per una data media  $\mu_c$  determinano la varianza  $\sigma_c$ . I coefficienti  $c_{i\alpha}$  sono ottenuti da

$$c_{i\alpha}^A = \frac{c_0}{M} + c_1 X_{i\alpha},$$

dove  $X_{i\alpha}$  è una variabile aleatoria binaria che assume valore 1 con probabilità

$$p_{i\alpha}^A = \begin{cases} \frac{\mu_c}{Mc_1} \left[ 1 + \frac{M-M_A}{M_A} q \right], & \text{se } \alpha \in A \\ \frac{\mu_c}{Mc_1} (1 - q), & \text{altrimenti} \end{cases}$$

per le famiglie specializzate e

$$p_{i\alpha}^{gen} = \frac{\mu_c}{Mc_1}$$

per la famiglia generalista.

Infine, il modello uniforme è parametrizzato da  $\mu_c$ ,  $q$  e da un parametro  $b$ , che regola l'ampiezza dell'intervallo di campionamento della distribuzione. La media è parametrizzata nello stesso modo del modello Gaussiano e Gamma, mentre la varianza, dipendente da  $b$ , è la stessa in tutta la matrice.

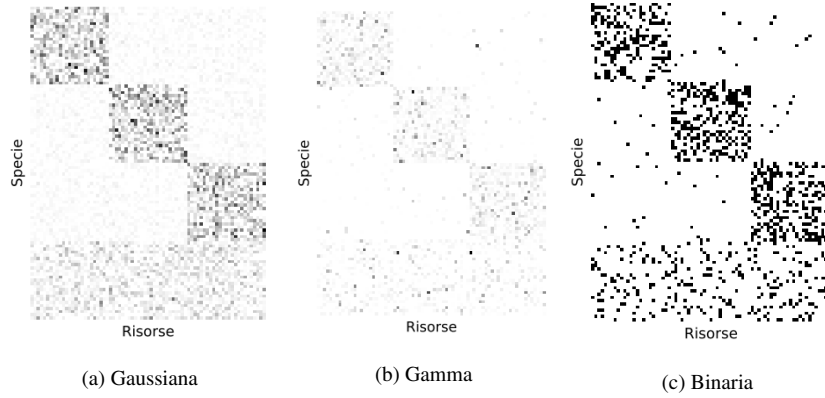


Figura 2.7: Heatmap delle matrici  $c_{i\alpha}$ , campionate in modo casuale da tre diverse distribuzioni. In questo esempio, le risorse sono suddivise in  $T = 3$  classi, le specie in  $F = 3$  famiglie specializzate e in una famiglia generalista. Il parametro  $q$  permette di regolare il grado di strutturazione delle matrici, in questo caso un  $q$  elevato ( $q=0.9$ ) porta a preferenze di consumo con una struttura ben marcata.

**Matrice metabolica  $D_{\beta\alpha}$ .** Il *MicroCRM* introduce una struttura metabolica grazie ad un modello di secrezione a tre livelli, che permette di implementare conoscenze minime sulla biochimica coinvolta nelle trasformazioni metaboliche (Figura 2.8). Il primo livello è una classe preferenziale di sottoprodotti metabolici, definita 'waste', ossia quella classe di sottoprodotti comunemente generati da molte vie metaboliche. Il secondo livello contiene i sottoprodotti metabolici della stessa classe delle risorse consumate e il terzo contiene tutti i restanti sottoprodotti. Questa struttura viene implementata nella matrice metabolica  $D_{\beta\alpha}$ , in modo che in media una frazione  $f_w$  del flusso di secrezione venga partizionata tra risorse del primo livello, mentre una frazione  $f_s$  tra risorse del secondo livello, e il resto vada al terzo livello. Tutto ciò si ottiene campionando ciascuna colonna  $\alpha$  da una distribuzione di Dirichelet con opportuni parametri  $d_{\beta\alpha}$ , dipendenti dalle classi  $A(\alpha)$  e  $A(\beta)$ , della risorsa consumata  $\alpha$  e di quella prodotta  $\beta$ . Indicando con  $w$  la classe 'waste' e con  $M_w$  la sua cardinalità, ogni colonna  $\alpha$  si ottiene da

$$Dir(d_{1\alpha}, d_{2\alpha}, \dots, d_{M\alpha})$$

$$d_{\beta\alpha} = \begin{cases} \frac{f_w}{sM_w}, & \text{se } A(\alpha) \neq w \text{ e } A(\beta) = w \\ \frac{f_s}{sM_{A(\alpha)}}, & \text{se } A(\alpha) \neq w \text{ e } A(\beta) = A(\alpha) \\ \frac{1-f_s-f_w}{s(M-M_{A(\alpha)}-M_w)}, & \text{se } A(\alpha), A(\beta) \neq w \text{ e } A(\beta) = A(\alpha) \\ \frac{f_w+f_s}{sM_w}, & \text{se } A(\alpha) = w \text{ e } A(\beta) = w \\ \frac{1-f_w-f_s}{s(M-M_w)}, & \text{se } A(\alpha) = w \text{ e } A(\beta) \neq w \end{cases}$$

dove  $s$  è un parametro che regola la struttura della matrice metabolica. Quando  $s \rightarrow 0$  si hanno reti di reazioni metaboliche complesse, in cui dal consumo di ciascuna risorsa si ottengono tutte i diversi tipi di risorse; mentre quando  $s \rightarrow 1$  si hanno reti di reazioni minime, dove da ciascuna risorsa in ingresso si ottiene un unico tipo di risorsa scelto casualmente. Quando vengono consumate risorse della classe 'waste', il primo e il secondo livello di secrezione coincidono, si ha quindi in media una frazione  $f_s + f_w$  allocata alla classe di sottoprodotti 'waste'. L'utilizzo di una distribuzione di Dirichelet, grazie alla sua intrinseca proprietà di campionare vettori la cui somma è pari ad 1, permette di allocare in modo casuale una fissata quantità, rappresentata in questo caso dal flusso in uscita  $l_\alpha J_{i\alpha}^{in}$ .

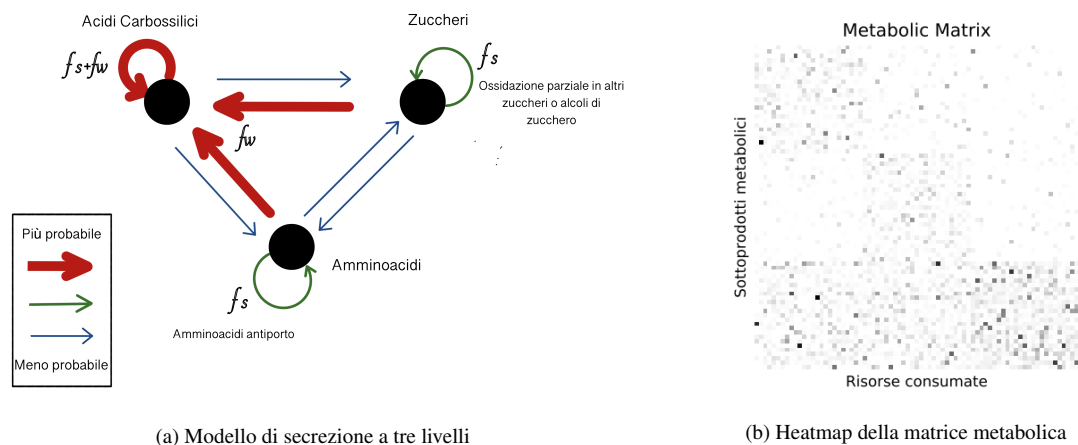


Figura 2.8: (a) Modello di secrezione che implementa una struttura metabolica basata su nozioni minime di biochimica. In questo esempio si considerano tre classi di risorse: zuccheri, amminoacidi e acidi carbossilici. Gli acidi carbossilici sono la classe 'waste' per i batteri fermentativi e respiro-fermentativi. In questo caso inoltre si ha  $f_w > f_s$ . (b) Heatmap della matrice metabolica campionata secondo il modello di secrezione a tre livelli. Ogni colonna contiene le frazioni di flusso in uscita (dovuto al consumo di un tipo di risorsa) allocate a ciascun tipo di risorsa.

Tutti i parametri delle equazioni dinamiche devono essere inclusi in un dizionario, da passare in ingresso al momento dell'inizializzazione di un'istanza della classe `Community`. Quando si vogliono avere parametri diversi tra i vari pozzetti, occorre una lista di dizionari. La funzione `MakeParams` genera automaticamente questi dizionari (sulla base di un dizionario di assunzioni e chiamando `MakeMatrices`), mantenendo gli stessi parametri in tutti i pozzetti, ad eccezione dei punti di equilibrio delle dinamiche intrinseche  $R_{\alpha}^0$ , che possono variare da pozzetto a pozzetto. In questo modo, coerentemente con `MakeInitialState`, permette di avere una fonte di carbonio variabile tra i diversi pozzetti. Quindi se si vogliono introdurre ulteriori differenze tra i parametri dei vari pozzetti, o se si vogliono fornire in ingresso più tipi di risorse, `MakeParams` non può essere utilizzato.

## 2.4 Simulazioni

### 2.4.1 Soluzione delle equazioni dinamiche

Il metodo di istanza `Propagate` invia lo stato e i parametri di ciascun pozzetto ad una diversa CPU (a tutte quelle disponibili); integrando le equazioni dinamiche dell'istanza, con la funzione `odeint` di SciPy, propaga tali stati per un tempo  $T$ , aggiornandoli alle dimensioni di popolazione e concentrazioni di risorse raggiunte.

Il metodo dispone di un'opzione `compress_species` (impostata di default a `True`), che elimina qualsiasi riferimento a specie estinte, riducendo le dimensioni dello stato e dei parametri dell'istanza. In questo modo permette di evitare che specie con dimensioni iniziali di popolazione nulle, ma con un tasso di crescita pro capite positivo, possano espandersi spontaneamente a causa dell'inevitabile imprecisione di qualsiasi risolutore numerico. Nel caso in cui si usi un modello diverso dal *MicroCRM*, al momento dell'inizializzazione di un'istanza, si deve fornire un ulteriore argomento, ossia un dizionario con le dimensioni dei parametri, così che i parametri riferiti alle specie possano essere individuati durante il processo di compressione.

### 2.4.2 Trasferimento su una nuova piastra

`Passage` è un metodo di istanza che simula il trasferimento (tramite pipette) di colture su una nuova piastra. Il metodo inizializza una nuova piastra aggiungendo una frazione  $f_{\mu\nu}$ , del contenuto di

ciascun vecchio pozzetto  $v$ , a ciascun nuovo pozzetto  $\mu$ . Tali frazioni sono contenute nella matrice  $f$  ( $n \times n$ ), passata come argomento al metodo. Le risultanti abbondanze di specie  $N_i$  sono convertite da unità arbitrarie di concentrazione a dimensioni assolute di popolazione, tramite un fattore di scala specificato. Le dimensioni intere di popolazione sono poi ottenute usando un campionamento multinomiale basato sul numero medio di cellule microbiche di specie  $i$  trasferite al pozzetto  $v$ , ossia  $\sum_v f_{\mu v}(N_i)_v$ .

Passage contiene anche un'opzione `refresh_resource`, impostata di default a `True` che, ispirandosi ai tipici esperimenti su piastre a 96 pozzetti, permette di impostare la nuova piastra con le stesse concentrazioni di risorse iniziali della piastra di partenza.

Questo metodo risulta un valido strumento per simulare le dinamiche di metacomunità in cui vi sia migrazione tra i pozzetti.

Inoltre, esso aiuta a stabilizzare l'integrazione nelle lunghe simulazioni, dove i valori esponenzialmente piccoli delle popolazioni di specie dirette verso l'estinzione, portano ad un'instabilità numerica. Il campionamento multinomiale, infatti, fissa a zero le dimensioni di popolazione significativamente inferiori ad 1. Quando si vuole eliminare l'instabilità numerica nelle simulazioni con un modello di rifornimento continuo, si può chiamare ripetutamente `Passage`, con assegnata ad  $f$  la matrice identità e `refresh_resource` impostato a `False`. La funzione `RunExperiment(f, T, np)` permette di eseguire  $np$  iterazioni identiche dei metodi `Passage(f)` e `Propagate(T)`, ossia applica la matrice di trasferimento  $f$  e propaga per un tempo  $T$ ,  $np$  volte. In questo modo si può eliminare l'instabilità numerica, senza che il valore di  $T$  influisca sui risultati della simulazione.

### 2.4.3 Identificazione efficiente dei punti di equilibrio

Il metodo `SteadyState` permette di trovare gli stati stazionari del sistema attraverso un nuovo algoritmo basato sull'ottimizzazione convessa e sull'approccio risolutivo di *Expectation Maximization*. Sotto determinate assunzioni di modello, che garantiscono la convessità del problema, quali, una risposta funzionale di Tipo I e l'assenza di regolazione metabolica, è possibile individuare gli stati stazionari in modo efficiente, senza dover ricorrere all'integrazione numerica delle equazioni differenziali.

L'algoritmo sfrutta la scoperta di una dualità tra i modelli consumatore-risorse e l'ottimizzazione vincolata nello spazio delle risorse [13, 14]. Nella regione dello spazio di risorse in cui il tasso di crescita  $dN_i/dt$ , di ciascuna specie  $i$ , è zero o negativo, lo stato stazionario  $R^*$  minimizza local-

mente una misura di dissimilarità  $d(R^0, R)$  tra il vettore delle concentrazioni correnti  $R$  e il punto di equilibrio delle dinamiche di risorsa intrinseche  $R^0$ . La misura di dissimilarità adottata dipende dal tipo di rifornimento delle risorse; nel caso di risorse fornite esternamente  $d$  è una divergenza di Kullback-Leibler pesata:

$$d^{external}(R^0, R) = \sum_{\beta} w_{\beta} \tau_{\beta}^{-1} \left[ R_{\beta}^0 \ln \frac{R_{\beta}^0}{R_{\beta}} - (R_{\beta}^0 - R_{\beta}) \right] \quad (2.9)$$

quando le risorse si auto-rigenerano  $d$  diventa una distanza euclidea pesata:

$$d^{self-renewing}(R^0, R) = \sum_{\beta} w_{\beta} r_{\beta} (R_{\beta}^0 - R_{\beta})^2$$

Questa dualità risulta valida quando le interazioni tra le coppie di specie consumatrici sono simmetriche, quindi, nel caso del *MicroCRM*, oltre alle assunzioni di assenza di regolazione metabolica e di tasso di consumo di Tipo I, occorrerebbe impostare  $l_{\alpha} = 0$  per ogni risorsa  $\alpha$ . Infatti, la produzione di sottoprodotti metabolici, regolata dai parametri  $l_{\alpha}$ , rende le interazioni tra consumatori asimmetriche. Per estendere l'applicabilità dell'approccio di ottimizzazione a modelli in cui si abbia  $l_{\alpha} > 0$ , *SteadyState* aggiorna il punto di equilibrio  $R^0$  delle dinamiche ambientali intrinseche ad un nuovo valore  $\tilde{R}^0(R^*)$ , il quale tiene conto delle risorse prodotte dalle specie quando il sistema è nel suo stato di equilibrio  $R^*$ . Il nuovo punto di equilibrio viene ricavato dalla seguente espressione:

$$\tilde{R}_{\alpha}^0(R^*) = R_{\alpha}^0 + \sum_{\beta \neq \alpha} \frac{Q_{\alpha\beta}^{-1}}{Q_{\beta\beta}^{-1}} (R_{\beta}^0 - R_{\beta}^*) \quad (2.10)$$

dove

$$Q_{\alpha\beta} = \delta_{\alpha\beta} - l_{\beta} D_{\alpha\beta} \frac{w_{\beta}}{w_{\alpha}}$$

e gli elementi della matrice inversa di  $Q_{\alpha\beta}$ , indicati con  $Q_{\alpha\beta}^{-1}$ , soddisfano  $\sum_{\beta} Q_{\alpha\beta}^{-1} Q_{\beta\gamma} = \delta_{\alpha\gamma}$ . Nel caso in cui il *MicroCRM* venga utilizzato con un consumo di Tipo I, senza regolazione metabolica e un rilascio di sottoprodotti metabolici da parte dei consumatori, la misura di dissimilarità si ottiene considerando nell'espressione (2.9) il nuovo punto di equilibrio (2.10):

$$d(\tilde{R}^0, R) = \sum_{\beta} \tilde{w}_{\beta} \tau_{\beta}^{-1} \left[ \tilde{R}_{\beta}^0 \ln \frac{\tilde{R}_{\beta}^0}{R_{\beta}} - (\tilde{R}_{\beta}^0 - R_{\beta}) \right]$$

con

$$\tilde{w}_{\alpha} = Q_{\alpha\alpha}^{-1} (1 - l_{\alpha}) \tau_{\alpha}^{-1} w_{\alpha}$$



Lo stato stazionario  $R^*$  viene ricavato risolvendo il problema di minimizzazione

$$R^* = \operatorname{argmin}_R d(\tilde{R}^0(R^*), R)$$

attraverso un approccio iterativo di *Expectation Maximization*.

### 3. Implementazione di un metodo per definire le interazioni tra specie

Al fine di ottenere una rete di interazioni 'ground truth', che determini la specifica composizione di una comunità microbica simulata, le interazioni devono essere ricavate tramite un approccio che sia indipendente dai diversi metodi di *Reverse Engineering* presenti in letteratura, in modo che nessuno di essi venga favorito in una successiva valutazione delle performance. Le interazioni negative, ossia interazioni che hanno un effetto negativo sulle due specie coinvolte, si instaurano quando le specie competono per uno stesso tipo di risorsa, quindi si possono individuare confrontando i profili di consumo di ciascuna di esse. Le interazioni positive che si instaurano tra due diverse specie microbiche, ossia interazioni che hanno un effetto positivo sulle specie coinvolte, si possono definire basandosi sui consumi che caratterizzano una specie e sui sottoprodotti metabolici rilasciati dall'altra nell'ambiente. Infatti, due specie si aiutano a vicenda, quando una produce tipi di risorsa che l'altra consuma e viceversa. In questo modo, a differenza dei metodi di *Reverse Engineering*, che estraggono le reti di interazioni analizzando i valori di abbondanza delle specie, le interazioni vengono ricavate sulla base delle risorse consumate e prodotte dalle specie; in particolare si analizzano i flussi energetici mediati da tali risorse, calcolati a partire da alcuni parametri del *MicroCRM*, modello matematico implementato nel *Community Simulator*.

In questa tesi è stata implementata una funzione, *InteractionNetworks*, la quale permette di assegnare un peso compreso tra 0 e 1 alle interazioni che si sviluppano tra le diverse coppie di specie, in un determinato istante temporale della simulazione. Tale funzione restituisce in uscita due matrici di adiacenza pesate, che rappresentano la rete delle interazioni positive e quella delle interazioni negative, nell'istante temporale analizzato. Per ogni coppia di specie ( $S_i, S_j$ ), che abbia consumi di risorse in comune, viene assegnato un peso  $w_{ij}$  alla competizione che risente  $S_j$  a causa della presenza di  $S_i$ :

$$w_{ij} = - \sum_{\alpha \in \{\text{Tipi di risorse consumate sia da } S_i \text{ che da } S_j\}} \frac{J_{i\alpha}^{in}}{J_{i\alpha}^{in} + J_{j\alpha}^{in}} \frac{J_{j\alpha}^{in}}{\sum_{\gamma} J_{j\gamma}^{in}}$$

dove  $\gamma$  indica i diversi tipi di risorsa consumati dalla specie  $S_j$ . Tale formula viene implementata partendo dalla matrice delle preferenze di consumo  $c_{i\alpha}$ , passando poi alla matrice dei flussi energetici in ingresso ai consumatori. Tali flussi sono dipendenti dalla concentrazione delle risorse e

quindi variabili nel tempo (in Figura 3.1 si riporta uno schema esemplificativo dei passi svolti per arrivare alla matrice di adiacenza pesata). Quando una risorsa  $\alpha$ , consumata da entrambe le specie, è presente nell'ambiente, essa contribuisce a determinare il peso complessivo. La competizione che risente la specie  $S_j$ , a causa del consumo della risorsa  $\alpha$  da parte di  $S_i$ , viene valutata tramite due fattori moltiplicativi: il primo mette a confronto il tasso di consumo di  $S_i$  rispetto a quello di  $S_j$ , mentre il secondo corregge tale peso considerando l'importanza per la specie  $S_j$  del flusso energetico proveniente dalla risorsa  $\alpha$ . Quest'ultimo termine, infatti, ad un determinato istante temporale, rappresenta la frazione del flusso totale in ingresso ad un microbo di specie  $S_j$ , mediata dal consumo della risorsa  $\alpha$ . In questo modo, la forza dell'interazione viene determinata considerando sia l'entità dei tassi di consumo delle diverse specie, sia la capacità delle specie di rifornirsi energeticamente da altre risorse.

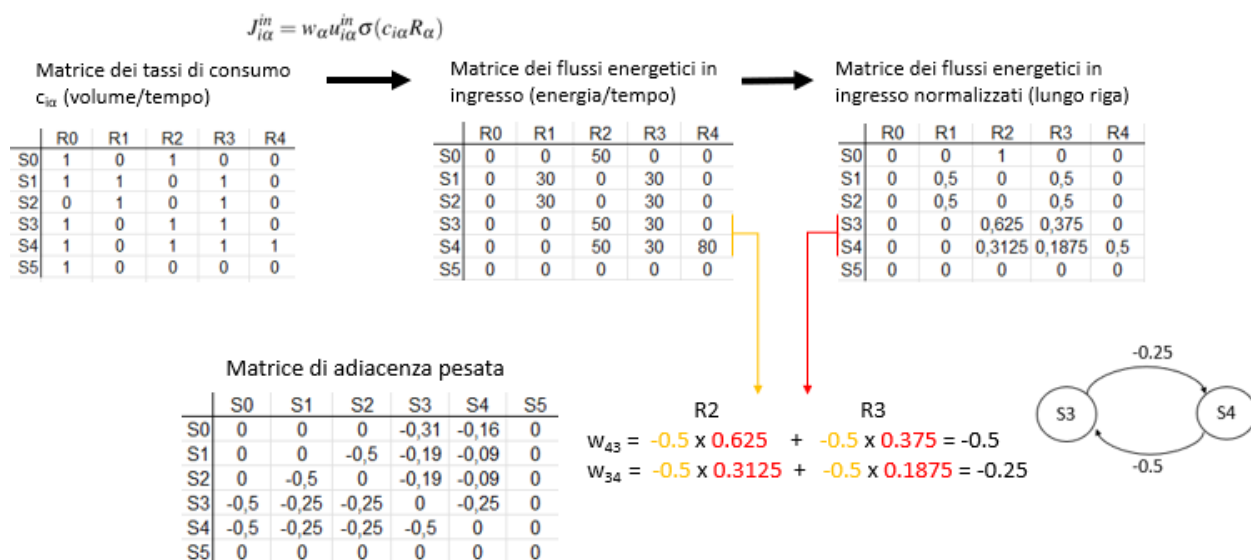


Figura 3.1: Schema esemplificativo dei passaggi implementati per ottenere una matrice di adiacenza con i pesi delle interazioni negative. Con  $S_i$  ( $i = 0, \dots, 5$ ), si indicano le diverse specie e con  $R_j$  ( $j = 0, \dots, 4$ ) i diversi tipi di risorsa. Dalla matrice dei consumatori, si passa alla matrice dei flussi energetici in ingresso (in questo esempio si scelgono  $w_\alpha = 1$ ,  $u_{i\alpha}^m = 1$  e una  $\sigma$  di Tipo-I) e da quest'ultima, attraverso una normalizzazione lungo le righe, ad una matrice di flussi energetici in ingresso normalizzati. Da tali matrici si analizzano i consumi in comune di ciascuna coppia di specie e si calcolano i pesi degli archi orientati.

Quando una specie  $S_i$  produce una risorsa consumata da  $S_j$ , il peso dell'interazione positiva viene calcolato considerando: quanto il consumo di quella risorsa è importante per la specie  $S_j$  e quanta risorsa produce  $S_i$ , in rapporto a quanta ne consuma  $S_j$ . Quindi il peso complessivo, che valuta tutti i diversi tipi di risorse coinvolte nell'interazione di coppia, viene ricavato nel modo

seguinte:

$$w_{ij} = \sum_{\alpha \in \{\text{Tipi di risorse consumate da } S_j \text{ e prodotte da } S_i\}} \min \left( 1, \frac{J_{i\alpha}^{out}}{J_{j\alpha}^{in}} \right) \frac{J_{j\alpha}^{in}}{\sum_{\gamma} J_{j\gamma}^{in}}$$

dove, quando il flusso prodotto dalla specie  $S_i$  eccede la richiesta energetica della specie  $S_j$ , il contributo del primo termine moltiplicativo è sempre pari a 1. Tale formula viene implementata partendo dalla matrice delle preferenze di consumo  $c_{i\alpha}$  e dalla matrice metabolica  $D_{\beta\alpha}$ , attraverso i passaggi schematizzati in figura.

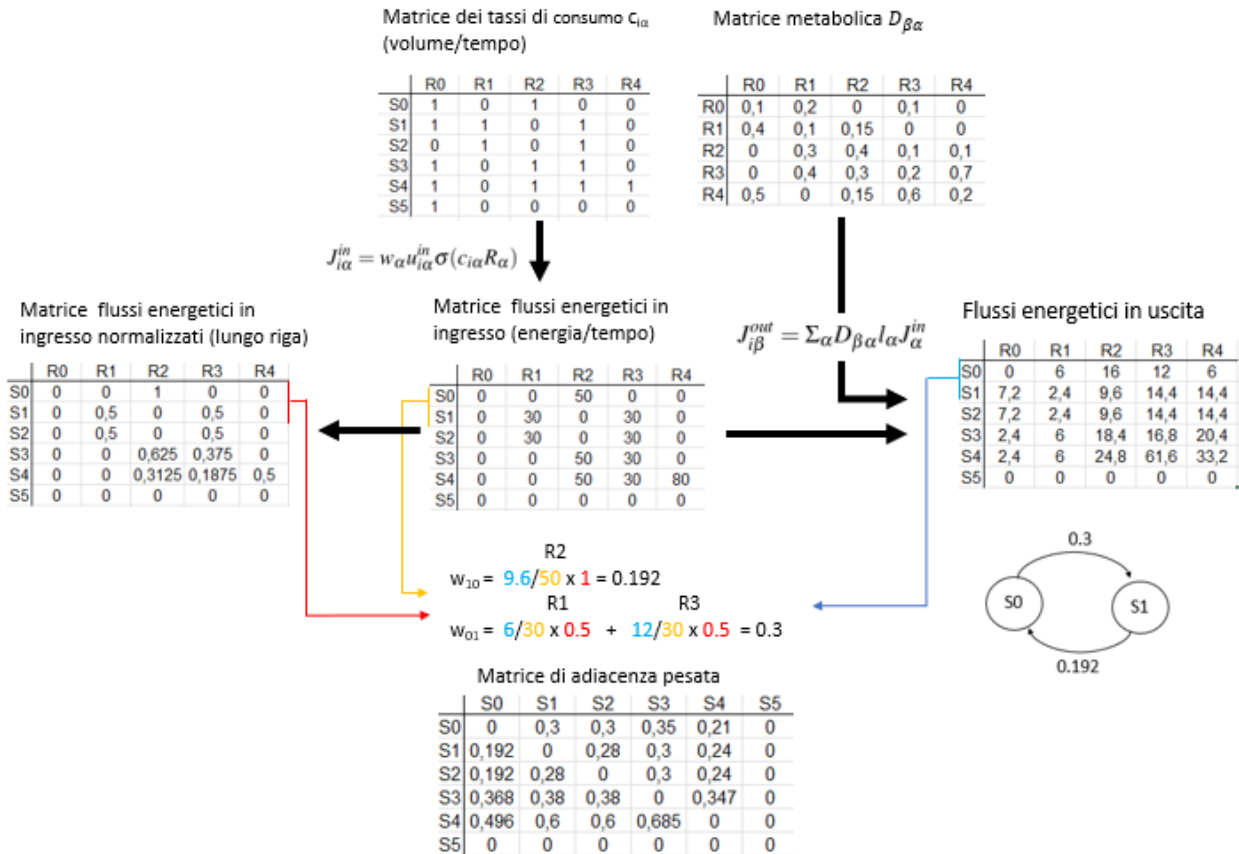


Figura 3.2: Schema esemplificativo dei passaggi implementati per ottenere una matrice di adiacenza con i pesi delle interazioni positive. Intrecciando i dati della matrice dei flussi energetici in ingresso ai consumatori (ottenuta dalla matrice dei consumatori) con i dati della matrice metabolica, si ricava la matrice dei flussi energetici in uscita dai consumatori. Quest'ultima matrice, insieme alle matrici dei flussi energetici in ingresso e di quelli normalizzati, viene utilizzata per ricavare i pesi positivi delle interazioni tra coppie di specie.

I pesi delle interazioni sono stati ottenuti attraverso un'analisi a coppie, in cui la presenza nel sistema di altre specie che competono e cooperano con i membri della coppia, viene presa in consi-

derazione per il fatto che esse influenzano le concentrazioni delle risorse coinvolte nell'interazione e di conseguenza anche i flussi energetici utilizzati nella determinazione dei pesi di coppia.

Le reti estratte dipendono dalle concentrazioni delle risorse e quindi, nella fase di transitorio fino allo stato stazionario, variano nel tempo; l'obiettivo è quello di ricavare da esse un'unica rete complessiva di tutte le interazioni, sia positive, che negative, la quale verrà poi confrontata con le reti ottenute dai diversi algoritmi di *Reverse Engineering*.

## 4. Simulazione di reti di interazioni microbiche

Tramite l'utilizzo del *Community Simulator*, integrato con la funzione *InteractionNetworks*, è stato possibile simulare le interazioni, positive e negative, che nel tempo si instaurano tra le diverse specie di una determinata nicchia ecologica. Tali interazioni determinano i profili di abbondanza ottenuti dalla simulazione. In questo capitolo vengono analizzate le simulazioni svolte e i principali problemi con esse affrontati. Inoltre, si mettono in evidenza le sfide future ancora aperte.

La comunità microbica che si è simulata è composta da  $S=60$  specie, suddivise in tre famiglie specializzate nel consumo di tre diverse classi di risorse. Nell'ambiente si considera un numero  $M=60$  di tipi diversi di risorse, uguale al numero di specie. La matrice dei consumatori viene campionata da una distribuzione di Bernoulli, con i seguenti parametri:  $c_0 = 0$ ,  $c_1 = 1$  (come di default),  $\mu_c = 7$  (impostato in modo proporzionale rispetto al numero di tipi di risorse, secondo il rapporto di proporzionalità usato nei valori di default) e un livello di specializzazione delle famiglie alto,  $q = 0.9$ . I parametri che determinano la struttura della matrice metabolica sono impostati come di default, in particolare  $s = 0.2$  comporta una matrice di densità elevata (Figura 4.1). Con  $l_\alpha = 0.8$ , per tutte le risorse  $\alpha$ , si impongono forti rapporti di simbiosi tra i consumatori. Per far evolvere il sistema ad uno stato stazionario non nullo, ossia per evitare l'estinzione di tutte le specie, la comunità viene rifornita con un flusso costante di specifiche risorse. Tale flusso viene fissato ad un valore elevato, in modo tale che, grazie anche ad un  $l_\alpha$  alto, la comunità abbia una grande disponibilità di risorse, e quindi l'estinzione delle specie sia determinata essenzialmente da rapporti di competizione. Inoltre le dinamiche sono state generate con una risposta di Tipo-I e senza regolazione metabolica (come di default).

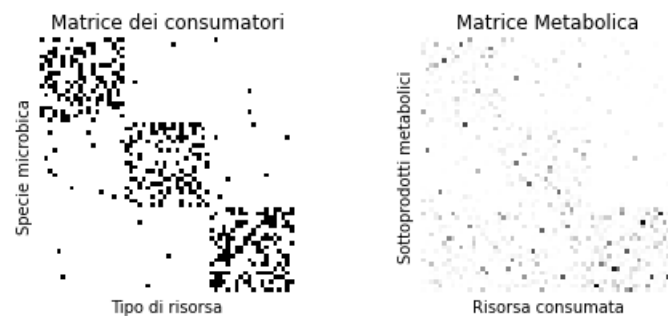


Figura 4.1: Struttura tassonomica e metabolica della comunità simulata.

## 4.1 Inizializzazione dello stato di una comunità microbica

Nel simulare una comunità microbica, in un determinato scenario ambientale, una prima considerazione riguarda l'istante iniziale  $t_0$  della simulazione. In tale istante infatti, lo stato del sistema che si otterrebbe con un'inizializzazione casuale, sarebbe caratterizzato da interazioni tra specie non realistiche; ossia, le relazioni di dipendenza tra le diverse variabili del sistema rifletterebbero una situazione creata dall'utente stesso. Per ottenere uno stato stazionario iniziale che sia realistico, si è ricorso a due simulazioni sequenziali. Una prima simulazione permette di arrivare ad uno stato stazionario finale, il quale successivamente viene utilizzato come stato iniziale di una seconda simulazione, in cui uno stimolo viene fornito al sistema. In quest'ultima è possibile ricavare, ad ogni istante temporale, le due reti di interazioni positive e negative, tramite la procedura descritta nel capitolo precedente e implementata nella funzione `InteractionNetworks`.

Nella prima simulazione, per inizializzare le abbondanze delle specie a valori biologicamente realistici, si è sfruttata una distribuzione di probabilità Gamma, che modella tali abbondanze considerando la variabilità riscontrabile tra repliche biologiche. Indicando con  $NO_{ij}$  l'abbondanza iniziale della specie  $i$  presente nella replica  $j$ , si ha:

$$NO_{ij} \sim \text{Gamma} \left( \frac{1}{\phi_i}, \phi_i \cdot \mu_i \right)$$

dove  $\mu_i$  rappresenta l'abbondanza media della specie  $i$  e  $\phi_i$  è un parametro che descrive la variabilità biologica nell'abbondanza della specie  $i$ ; per ciascuna delle  $S$  specie, tali parametri sono stati estratti campionando casualmente un vettore  $(\mu_0, \dots, \mu_{S-1})$  e un vettore  $(\phi_0, \dots, \phi_{S-1})$  da un insieme di valori preimpostati, inclusi nel simulatore *metaSPARSim*. In questa prima simulazione si è utilizzata una piastra virtuale con 100 pozzetti, ciascuno dei quali contenente la stessa comunità di specie microbiche, ma con abbondanze iniziali diverse, ottenute campionando cento volte dalla stessa distribuzione Gamma. Inoltre, tra i pozzetti si è mantenuto lo stesso ambiente iniziale e lo stesso rifornimento continuo di risorse, ossia sei tipi di risorse, corrispondenti ai primi due tipi di ciascuna classe e forniti con flussi uguali e costanti nel tempo. Selezionato un pozzetto, si utilizzano le abbondanze delle specie e delle risorse, nello stato stazionario finale, per inizializzare lo stato di una seconda simulazione, in cui si fornisce uno stimolo esterno che va a modificare l'ambiente nel pozzetto (Figura 4.2). Per simulare la stessa comunità in diversi scenari ambientali, si dovrà cambiare il tipo di stimolo fornito, agendo per esempio sulle risorse fornite in continuo.

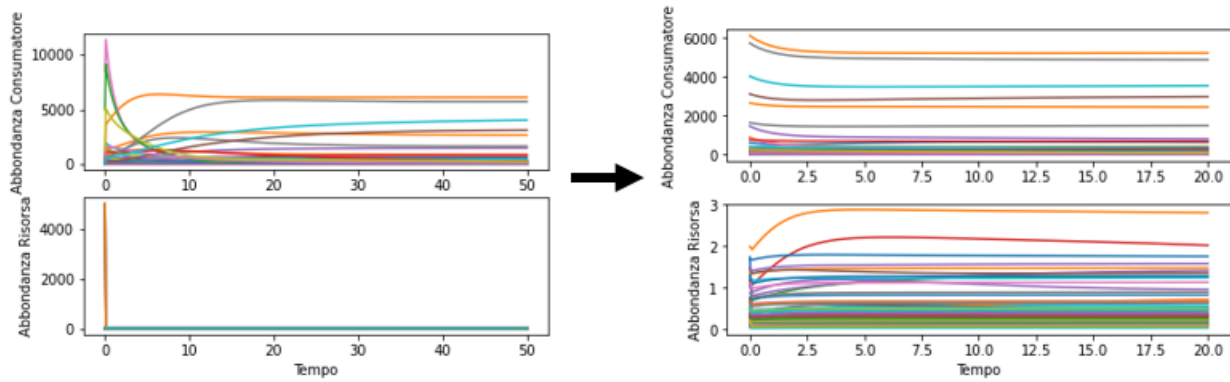


Figura 4.2: Utilizzo di due simulazioni sequenziali per ottenere all'istante  $t_0$  delle interazioni microbiche realistiche. Nella prima simulazione le dinamiche delle risorse sono rapide rispetto alla scala temporale dei cambiamenti nelle abbondanze delle specie. Rispetto alla prima simulazione, nella seconda si mantiene lo stesso rifornimento esterno di risorse, tranne per un tipo di risorsa che non viene più fornita.

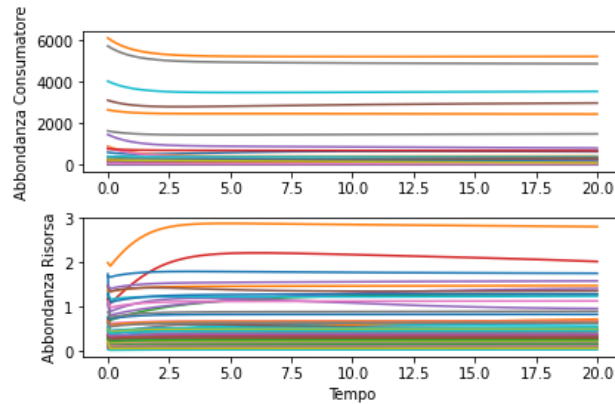
## 4.2 Analisi delle reti al variare del tempo

Nella seconda simulazione si lascia evolvere il sistema fino ad uno stato stazionario e ad ogni istante temporale si ricavano le due matrici di adiacenza pesate, corrispondenti alle reti di interazioni positive e negative.

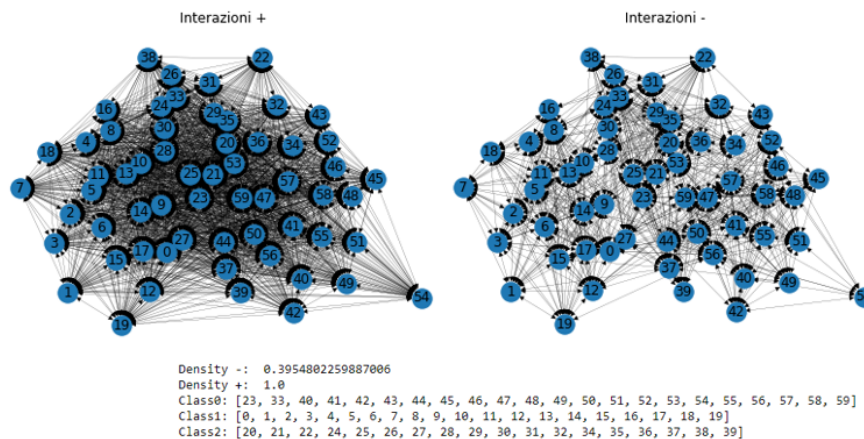
Per come sono state definite le interazioni, basandosi sulla matrice dei consumatori e su quella metabolica, la concentrazione delle risorse rappresenta l'unico termine che teoreticamente potrebbe portare a reti di interazioni con strutture variabili nel tempo, ossia ad avere archi mancanti quando determinati tipi di risorse sono assenti. Tuttavia, se la risorsa è inizialmente presente, l'integrazione del sistema nel tempo può portare a valori di concentrazione molto bassi, ma mai nulli. Quindi, nel tempo si osservano reti con pesi variabili, ma con una struttura topologica invariata. Analizzando tale struttura in reti ottenute dalla simulazione in figura 4.3 (a), la densità della rete di interazioni negative risulta essere approssimativamente pari a 0.395, mentre per quella delle interazioni positive pari a 1. Tutte le specie risultano, quindi, legate da un rapporto di aiuto reciproco; questo perché il parametro di modello  $s$ , che regola la densità della matrice metabolica, è stato impostato ad un valore molto basso ( $s=0.2$ , come di default). Regolando tale parametro a valori più prossimi ad 1, si possono ottenere reti di competizione meno dense. Inoltre, sulla rete negativa è stata effettuata un'analisi dei cluster presenti, per individuare i gruppi di nodi più interconnessi; come da aspettative, sono state individuate tre classi corrispondenti quasi del tutto alla suddivisione delle specie in



famiglie specialiste. Infatti, il parametro  $q$ , che regola il grado di specializzazione delle famiglie, è stato impostato ad un valore elevato ( $q = 0.9$ , ossia le specie consumano quasi esclusivamente risorse appartenenti alla classe preferenziale della famiglia a cui appartengono) e quindi, ci si aspetta che le interazioni di competizione si instaurino principalmente tra specie appartenenti alla stessa famiglia (Figura 4.3).



(a) Seconda simulazione



(b) Rete di interazioni positive e negative.

Figura 4.3: (a) Seconda simulazione (b) Reti di interazioni positive e negative, con archi non pesati. I nodi rappresentano le 60 specie e gli archi le interazioni tra di esse. Tale struttura rimane invariata nel tempo. Le tre classi rappresentano i cluster di nodi più interconnessi.

Oltre all'invarianza nel tempo della struttura delle reti, sarebbe desiderabile che anche i pesi delle interazioni non subiscano consistenti variazioni, ossia che le principali interazioni rimangano grossomodo le stesse nel tempo. Questo perché, alla fine, da tutte le reti ottenute ad ogni istante temporale, bisognerà estrarre un'unica rete complessiva. Una soluzione esplorabile potrebbe essere quella di mediare separatamente i pesi positivi e negativi ottenuti nei diversi istanti del transitorio e,

successivamente, sommare le due matrici di adiacenza risultanti. In quest'ottica, è stato effettuato un confronto tra le reti ricavate in cinque istanti temporali (quattro istanti per il transitorio e uno per lo stato stazionario), applicando diverse soglie alle loro matrici di adiacenza pesate. Tali soglie corrispondono al 5%, al 50% e al 90% del peso massimo presente in ciascuna matrice (per i pesi negativi si calcola il valore assoluto) e permettono di selezionare le interazioni più rilevanti ponendo a 1 tutti i pesi che superano la soglia considerata e a 0 tutti gli altri. In questo modo, tramite una semplice differenza tra matrici di adiacenza sogliate, si evidenziano gli archi non in comune. In figura 4.4 si riportano i diversi confronti effettuati, dove si sono sfruttate matrici delle distanze di Manhattan per ricavare il numero di archi non in comune tra le diverse coppie di reti confrontate. In generale, si può osservare che nei diversi istanti considerati le interazioni di maggior intensità si instaurano circa tra le stesse coppie di specie.

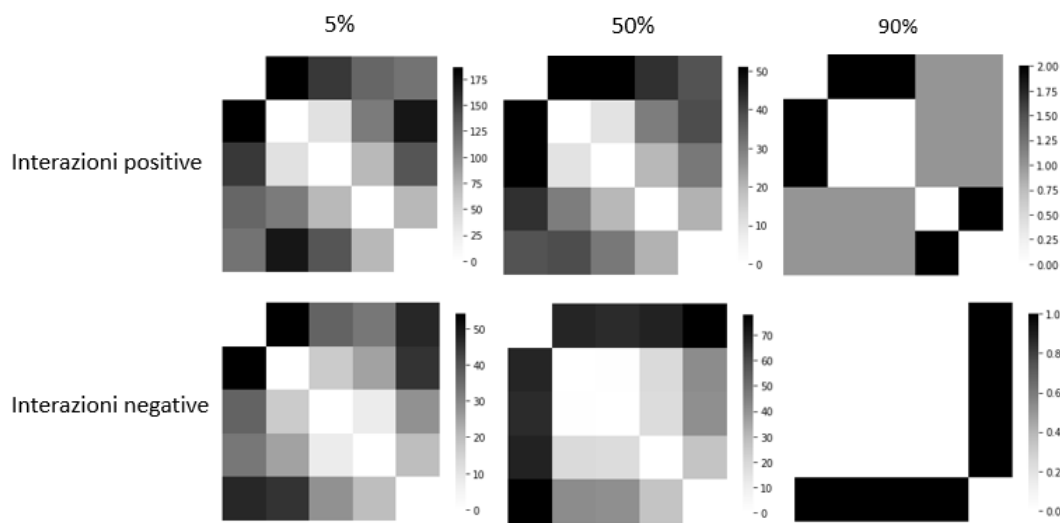


Figura 4.4: Heatmap per confrontare le reti di interazione in 5 istanti temporali. I valori delle matrici rappresentano il numero di archi non in comune tra la coppia di reti confrontate a diverse soglie.

### 4.3 Simulazione di repliche biologiche

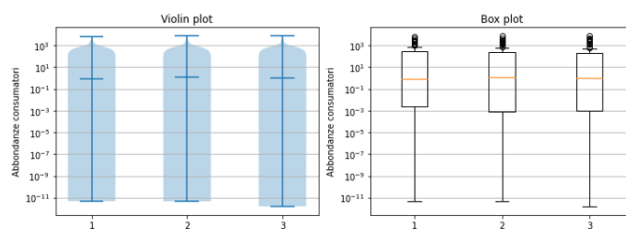
Un'altra importante sfida da affrontare, è quella di individuare una strategia per simulare una variabilità biologica nei profili di abbondanza delle specie presenti in una specifica comunità microbica. Infatti, un obiettivo è quello di simulare delle repliche biologiche, per poi ottenere una rete che sia stabile rispetto alla variabilità biologica riscontrabile in tali repliche. Per repliche biologiche si intendono delle comunità composte dalle stesse specie, nello stesso scenario ambientale, ma che, per fattori biologici intrinseci alle specie, presentano dei profili di abbondanze variabili. Tali comu-

nità sono state simulate sfruttando i diversi pozzetti di una piastra virtuale. Un possibile approccio potrebbe essere quello di agire sui parametri inerenti alle specie, ossia di assegnare ad alcuni di essi valori variabili da un pozzetto all'altro. Il campionamento dei parametri  $c_{i\alpha}$  e  $D_{\beta\alpha}$ , implementato nel *Community Simulator*, non può essere sfruttato per tali scopi, in quanto andrebbe a cambiare il tipo di risorse consumate e prodotte dalle stesse specie. Al contrario, ci si aspetta che le stesse specie presenti nelle diverse repliche biologiche abbiano un metabolismo identico, ossia consumino e producano gli stessi tipi di risorse con tassi leggermente variabili.

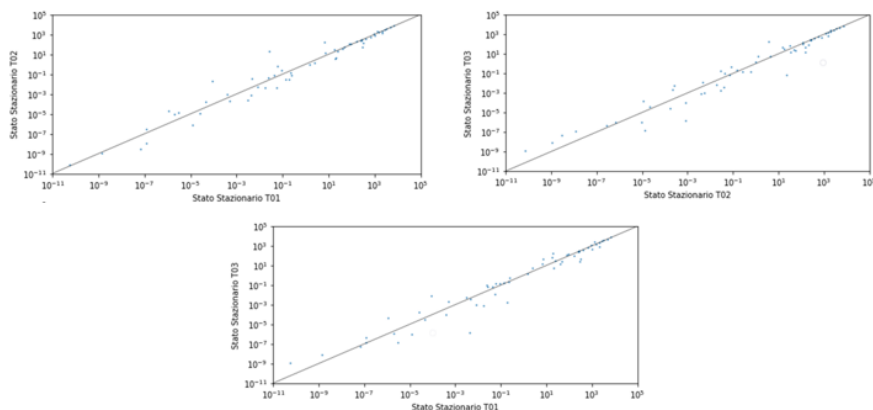
Un'idea alternativa potrebbe essere quella di inserire una variabilità biologica campionando le abbondanze iniziali delle specie da una distribuzione Gamma, la stessa per ogni pozzetto. Come accennato nella prima sezione di questo capitolo, nella prima simulazione si è utilizzato questo approccio. Un punto critico è rappresentato dal fatto che il simulatore, per come si è utilizzato, risulta essere deterministico; quindi, le differenze negli stati stazionari finali, ottenuti nei vari pozzetti, potrebbero non rappresentare la variabilità biologica, ma essere semplicemente stati stazionari diversi dovuti a condizioni iniziali differenti. Per questo motivo nella sezione successiva si riporta un confronto tra questi stati stazionari. Infine, si affronta un'analisi comparativa tra le reti di interazione simulate nei diversi pozzetti.

### 4.3.1 Confronto tra stati stazionari

Confrontando, tra i diversi pozzetti, le abbondanze di ciascuna specie negli stati stazionari finali, si vuole valutarne la similarità. In figura 4.5, per semplicità visiva, si riporta un'analisi riferita a tre pozzetti: attraverso un box-plot e un violin-plot si valutano le distribuzioni delle abbondanze, ma nel confronto si perde la corrispondenza tra le specie; attraverso un grafico di dispersione (scatter-plot), si riescono invece a comparare le abbondanze di ciascuna specie. In figura 4.6 si confrontano 100 pozzetti sfruttando una matrice delle distanze euclidee. Tali distanze vengono calcolate sui dati di abbondanza standardizzati, in modo da assegnare lo stesso peso alle abbondanze delle diverse specie. Da una valutazione complessiva i diversi stati stazionari sembrano simili e dalla heatmap della matrice delle distanze si possono notare dei cluster di maggior similarità.



(a) Violin-plot e box-plot



(b) Scatter-plot

Figura 4.5: (a) Confronto tra le distribuzioni dei valori di abbondanza in 3 stati stazionari corrispondenti a 3 pozzetti diversi (b) Confronto tra le abbondanze di ciascuna specie nei diversi stati stazionari.

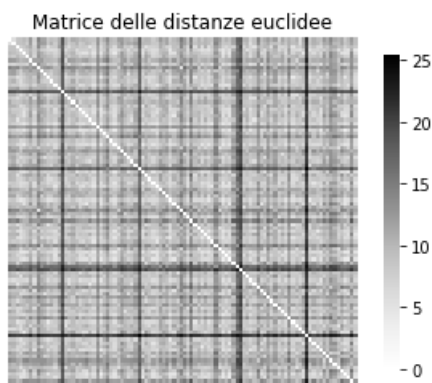


Figura 4.6: Heatmap della matrice delle distanze euclidee, valutata su dati standardizzati. Confronto tra 100 stati stazionari.

### 4.3.2 Analisi delle reti

A partire da tutti i diversi stati stazionari ottenuti nella prima simulazione, si è inizializzata una seconda simulazione. Assumendo che il campionamento iniziale delle abbondanze di specie permetta di simulare i profili di abbondanza di repliche biologiche, si vorrebbero riscontrare le stesse

interazioni microbiche in tutti i vari pozzetti. In un determinato istante temporale, la struttura delle reti, ricavate con la funzione `InteractionNetworks`, non varia tra i diversi pozzetti. Infatti, la matrice dei consumi e quella metabolica sono le stesse in tutte le repliche biologiche. I pesi degli archi, invece, cambiano a causa delle diverse concentrazioni di risorse. Applicando diverse soglie, si confrontano le reti ottenute nei 100 pozzetti, allo stato stazionario finale. Le interazioni principali risultano invariate, infatti le reti sogliate presentano un numero di archi non in comune relativamente basso (Figura 4.7).

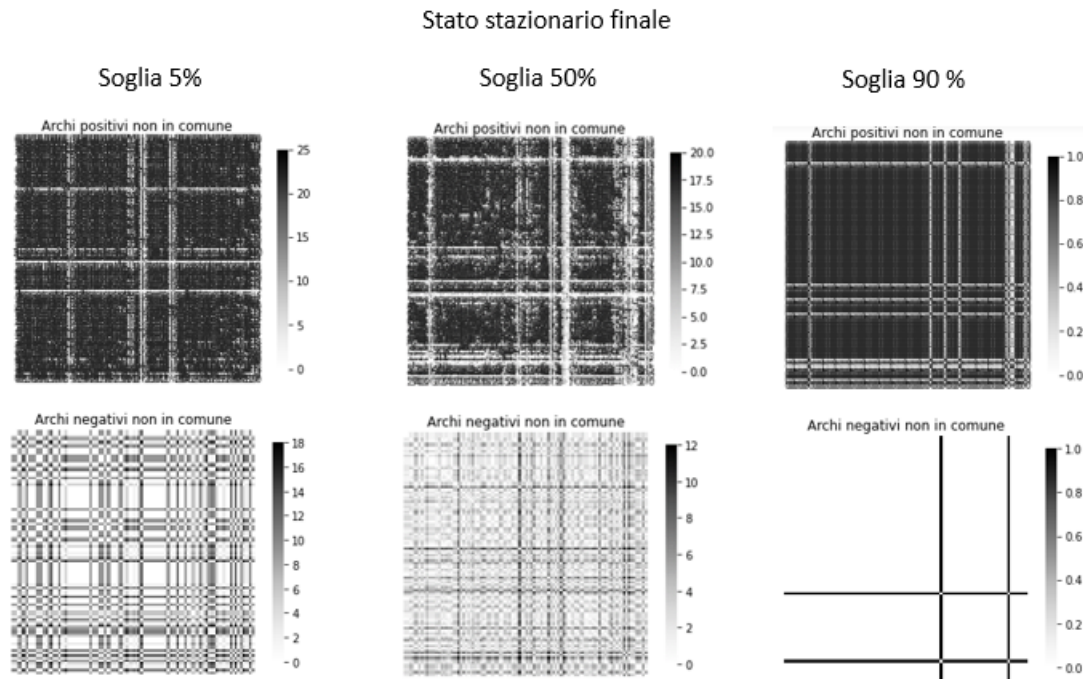


Figura 4.7: Confronto delle reti di interazione tra 100 pozzetti, allo stato stazionario finale.

# Conclusioni

In questa tesi si è affrontata l'applicazione di un simulatore di comunità microbiche, il *Community Simulator*, al fine di creare un contesto simulato in cui poter validare i metodi di *Reverse Engineering*. Dal simulatore si vuole ottenere una rete di interazioni microbiche che rappresenti relazioni realistiche tra le specie di una comunità. Essa rappresenterà la rete 'ground truth' per verificare la correttezza delle interazioni che i metodi di *Reverse Engineering* ricaveranno a partire da dati di sequenziamento sintetici. Tali dati, infatti, saranno ottenuti dalle abbondanze di specie simulate con il *Community Simulator*. Per ricavare le interazioni tra le diverse specie è stata implementata una funzione che, ad ogni istante temporale, restituisce le reti di interazioni positive e negative.

In questa applicazione, tale simulatore ha evidenziato molti aspetti positivi e altri punti critici da affrontare. Il simulatore permette di riprodurre l'evoluzione nel tempo di comunità microbiche di dimensioni realistiche, ossia costituite da un elevato numero di specie e risorse diverse. Inoltre, simulare processi, come l'alimentazione incrociata e la competizione per le risorse, fondamentali nel determinare il tipo e la forza delle interazioni che si instaurano tra le diverse specie della comunità simulata. Le due matrici che contengono la maggior parte della struttura tassonomica e metabolica, ossia la matrice dei consumatori e quella metabolica, consentono di ricavare le reti di interazioni positive e negative. Per di più, alcuni parametri permettono di regolare facilmente la struttura della rete simulata (ad esempio, densità e cluster di nodi), altri di avere un controllo sulla forza delle interazioni. Tuttavia, simulare profili di abbondanza e reti che siano il più possibile realistici, non è un obiettivo semplice. Inizializzare le abbondanze di specie e risorse a valori che riflettano una possibile situazione reale è di fondamentale importanza. Per questo, si è simulata la condizione iniziale partendo da uno stato iniziale casuale e portando il sistema ad uno stato stazionario finale. Successivamente, per simulare la dinamica, si è partiti da tale stato stazionario e si è perturbato il sistema. Anche le assunzioni di modello e la scelta di come impostare i parametri hanno un impatto sul risultato finale. Un altro punto critico risulta quello di riuscire a simulare una variabilità biologica nei profili di abbondanza delle specie. In una rete di interazioni la natura degli archi è intrinsecamente dinamica perché l'interazione dipende dalle risorse disponibili nell'ambiente. Si deve quindi definire quale rete di interazione si mira a ricostruire quando si intende applicare metodi di *Reverse Engineering*. Una scelta potrebbe essere quella di ricostruire ogni relazione che si

possa realizzare nell'ambiente osservato. In questa tesi, per come sono state definite le interazioni, le reti ottenute risultano dipendenti dal tempo, in quanto ricavate dai flussi energetici, che dipendono dalla concentrazione delle risorse. Un futuro proseguimento di questo lavoro sarà quello di dare in input i dati di abbondanza di specie simulati ad un altro simulatore, di nome *metaSPARSim*, che riproduce il processo di sequenziamento. Infine, ai dati così ottenuti verranno applicati alcuni metodi di *Reverse Engineering* e si effettuerà un confronto tra le reti estratte da tali metodi e quella di partenza.

# Bibliografia

- [1] Marsland R. et al. «The Community Simulator: A Python package for microbial ecology». In: *PLoS ONE* 15(3) (mar. 2020). DOI: 10.1371/journal.pone.0230430. URL: <http://doi.org/10.1371/journal.pone.0230430>.
- [2] I. Patuzzi et al. «metaSPARSim: a 16S rRNA gene sequencing count data simulator». In: *BMC Bioinformatics* (2019).
- [3] N.D. Gray e I.M. Head. «Microbial Ecology». In: *Encyclopedia of Ecology* (2008).
- [4] M. Cappellato et al. «Modeling Microbial Community Networks: Methods and Tools». In: *Current Genomics, Volume 21, No. 8* (2020).
- [5] Reed M. Stubbendieck, Carol Vargas-Bautista e Paul D. Straight. «Bacterial Communities: Interactions to Scale». In: *Front Microbiol* (2016). DOI: 10.3389/fmicb.2016.01234.
- [6] Ursell LK et al. «Defining the Human Microbiome». In: *Nutr Rev.* 2012;70 Suppl 1(Suppl 1):S38-S44 (2012). DOI: 10.1111/j.1753-4887.2012.00493.x.
- [7] Gruppo di lavoro "Microbiota". «IL Microbiota umano: dalla ricerca alle applicazioni cliniche. Raccomandazioni e Linee di indirizzo.» In: *Consiglio Superiore di Sanità - Sezione III* (2018).
- [8] Katharine Z. Coyte e Seth Rakoff-Nahoum. «Understanding Competition and Cooperation within the Mammalian Gut Microbiome». In: *Curr Biol.* 2019 Jun 3;29(11):R538-R544. (2019). DOI: 10.1016/j.cub.2019.04.017.
- [9] Wang Y.N. et al. «Effects of probiotics and prebiotics on intestinal microbiota in mice with acute colitis based on 16S rRNA gene sequencing». In: *Chin. Med. J. (Engl)* (2019).
- [10] Marsland R III, Cui W e Mehta P. «A minimal model for microbial biodiversity can reproduce experimentally observed ecological patterns». In: *Scientific Reports* 10:3308 (2020). DOI: 10.1038/s41598020-60130-2. URL: <http://doi.org/10.1038/s41598020-60130-2>.



- [11] Marsland R III et al. «Avaible energy fluxes drive a transition in the diversity, stability, and functional structure of microbial communities». In: *PLOS Computational Biology* 15 (2020). DOI: 10.1371/journal.pcbi.1006793. URL: <http://doi.org/10.1371/journal.pcbi.1006793>.
- [12] Goldford et al. «Emergent simplicity in microbial community assembly». In: *Science* 361 (6401) (2018). DOI: 10.1126/science.aat1168. URL: <http://www.science.org/doi/10.1126/science.aat1168>.
- [13] Mehta P et al. «Constrained optimization as ecological dynamics with application to random quadratic programming in high dimensions». In: *Physical Review E*. (2018). DOI: 99:052111. URL: <http://doi.org/10.1103/PhysRevE.99.052111>.
- [14] Marsland III R, Cui W e Mehta P. «The Minimum Enviromental Perturbation Principle: A new perspective on niche theory». In: *arXiv* (2019). DOI: 1901.09673.