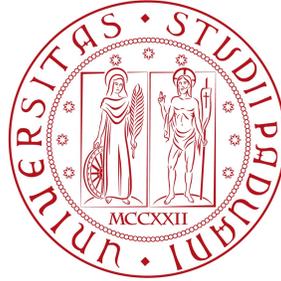


UNIVERSITÀ DEGLI STUDI DI PADOVA



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Tesi di Laurea Magistrale in Ingegneria dell'Automazione

**MODELLO FLUX BALANCE ANALYSIS E MODELLO  
BASATO SU EQUAZIONI DIFFERENZIALI DEL  
METABOLISMO DEL TRIPTOFANO**

RELATORE: Dott. Barbara Di Camillo

CORRELATORI: Prof. Mauro Bisiacco,  
Dott. Jole Costanza,  
Prof. Giuseppe Nicosia

LAUREANDA: Chiara Favaretto

Anno Accademico 2013/2014



# Contents

<b>1</b>	<b>Introduzione</b>	<b>9</b>
1.1	Il triptofano . . . . .	9
1.1.1	Obiettivi della tesi . . . . .	10
1.1.2	Uso del triptofano nella dieta . . . . .	11
1.1.3	Struttura della tesi . . . . .	11
1.2	Metabolismo e produzione di biomassa nelle cellule . . . . .	12
1.3	Processi metabolici del triptofano . . . . .	15
1.3.1	Via metossindolica . . . . .	15
1.3.2	Via della chinurenina . . . . .	16
<b>2</b>	<b>Modellizzazione del pathway del metabolismo del triptofano</b>	<b>20</b>
2.1	Rete (o pathway) metabolica . . . . .	20
2.1.1	Pro e contro del modello cinetico . . . . .	21
2.1.2	Come interpretare la rappresentazione di una reazione in un pathway . . . . .	21
<b>3</b>	<b>Modello cinetico del metabolismo del triptofano</b>	<b>22</b>
3.1	Descrizione del problema . . . . .	22
3.2	Metodi utilizzati . . . . .	24
3.2.1	Modellizzazione dell'attività specifica nei diversi tessuti . . . . .	25
3.3	Analisi effettuate tramite il Modello Cinetico . . . . .	26
3.3.1	Reazioni di particolare interesse nell'analisi . . . . .	27
3.3.2	Dati utilizzati . . . . .	28
3.4	Risultati . . . . .	28
3.4.1	Analisi dei flussi relativi . . . . .	28
3.4.1.1	Flussi relativi nei diversi tessuti (fegato e cervello). . . . .	28
3.4.1.2	Flussi relativi nel muscolo scheletrico: confronto tra soggetti sani e diabetici di tipo due . . . . .	31
3.4.2	Analisi di sensitività . . . . .	32
3.4.2.1	Analisi di sensitività delle concentrazioni rispetto a variazioni di espressione genica o concentrazione enzimatica . . . . .	32
3.4.2.2	Analisi di sensitività dei flussi di reazione rispetto a variazioni di espressione genica o concentrazione enzimatica . . . . .	36
3.4.2.3	Considerazioni finali sull'analisi di sensitività . . . . .	41
<b>4</b>	<b>Modelli basati su vincoli (constraint-based)</b>	<b>42</b>
4.1	Flux Balance Analysis (FBA) . . . . .	43
4.1.1	Introduzione . . . . .	43
4.1.1.1	Definizione del sistema . . . . .	43
4.1.1.2	Acquisizione della matrice stechiometrica . . . . .	44
4.1.1.3	Definizione della funzione obiettivo . . . . .	45
4.1.1.4	Ottimizzazione . . . . .	46
4.2	Varianti della FBA . . . . .	47
4.2.1	Geometric FBA . . . . .	48
4.2.1.1	Metodi . . . . .	48
4.2.1.2	Osservazioni conclusive . . . . .	51

4.2.2	Parsimonious Flux Balance Analysis (pFBA)	51
4.2.3	Flux Variability Analysis (FVA)	54
4.3	Sampling (metodo Monte Carlo per reti metaboliche)	55
4.3.0.1	Sampling di uno spazio di soluzioni privo di loop	56
4.3.1	Algoritmo ACHR	57
4.3.1.1	Hit and run Sampling	57
4.4	Analisi del modello (lunghezza degli extreme pathways e Reaction Participation)	58
4.4.1	Extreme pathways	58
4.4.1.1	Lunghezza del pathway e matrice della Reaction Participation	58
4.4.1.2	Correlazione tra i valori delle Reaction Participation e definizione di Sottosistemi di Reazioni	61
4.5	Gap filling	61
4.5.1	Ricerca dei gap in un modello metabolico	61
4.5.1.1	Algoritmo di identificazione dei no-production metabolites	61
4.5.2	Ricerca dei candidati per riempire i gap	63
4.5.2.1	Algoritmo per il riempimento dei Gap	65
4.6	Robustness Analysis	66
4.7	Metabolic engineering	66
4.7.1	OptKnock	66
4.7.2	OptGene	68
4.7.2.1	Algoritmo	69
4.7.3	Genetic design local search (GDLS)	71
4.7.3.1	Algoritmo di ottimizzazione MILP per la ricerca locale	73
<b>5</b>	<b>COBRA Toolbox</b>	<b>74</b>
5.1	Struttura di COBRA Toolbox	74
5.2	Nozioni basilari per l'utilizzo di COBRA Toolbox	75
5.3	Funzionalità di COBRA Toolbox	75
<b>6</b>	<b>Modelli FBA del metabolismo del triptofano</b>	<b>80</b>
6.1	Considerazioni generali	80
6.2	Costruzione dei modelli FBA in formato Excel e visualizzazione	81
6.2.1	Struttura del file contenente il modello	81
6.2.1.1	Cartella 'reactions'	81
6.2.1.2	Cartella 'metabolites'	82
6.2.2	Ripartimento delle informazioni e assunzioni	82
6.2.3	Visualizzazione del modello completo	83
6.3	Analisi effettuate	83
6.3.1	Reazioni di interesse	85
6.3.2	Simulazione della sottoproduzione di melatonina	85
6.4	Risultati ottenuti dal modello FBA completo	86
6.4.1	Analisi pFBA	86
6.4.2	Analisi FBA ed FVA	86
6.4.3	Analisi dei KnockOut	88
6.5	Risultati ottenuti dal modello FBA ridotto	90
6.5.1	Analisi pFBA	90
6.5.2	Analisi FBA ed FVA	90
6.5.3	Analisi dei KnockOut	92
6.6	Integrazione dei risultati ottenuti tramite modello cinetico e modello FBA ridotto	92
<b>7</b>	<b>Conclusioni</b>	<b>94</b>

<b>A</b>	<b>Descrizione dei tasks del software COPASI</b>	<b>97</b>
A.1	Steady-State Analysis . . . . .	97
A.2	Stoichiometric State Analysis . . . . .	98
A.2.1	Elementary Flux Modes . . . . .	98
A.2.2	Mass Conservations . . . . .	98
A.3	Time Course Analysis . . . . .	98
A.3.1	Deterministic Simulation (LSODA) . . . . .	99
A.3.2	Stochastic Simulation: The Next-Reaction-Method . . . . .	99
A.3.3	Hybrid Simulation: Runge-Kutta . . . . .	100
A.4	Lyapunov Exponents . . . . .	100
A.5	Time Scale Separation Analysis . . . . .	101
A.5.1	Metodo ILDM (Deuffhard) . . . . .	101
A.5.2	Metodo ILDM Modificato . . . . .	102
A.6	Cross Section . . . . .	102
A.7	Parameter Scan . . . . .	102
A.8	Parameter Estimation . . . . .	102
A.8.1	Dati sperimentali . . . . .	102
A.9	Ottimizzazione . . . . .	102
A.9.1	Evolutionary Programming (EP) . . . . .	103
A.9.2	Evolutionary Strategy con Ranking Stocastico (SRES) . . . . .	103
A.9.3	Genetic Algorithm (GA) . . . . .	103
A.9.4	Genetic Algorithm SR . . . . .	104
A.9.5	Hooke & Jeeves . . . . .	104
A.9.6	Levenberg - Marquardt . . . . .	105
A.9.7	Nelder - Mead . . . . .	105
A.9.8	Particle Swarm . . . . .	105
A.9.9	Praxis . . . . .	106
A.9.10	Random Search . . . . .	106
A.9.11	Simulated Annealing . . . . .	106
A.9.12	Steepest Descent . . . . .	107
A.9.13	Truncated Newton . . . . .	107
A.10	Metabolic Control Analysis (MCA) . . . . .	108
A.10.1	Control Coefficients . . . . .	108
A.10.2	Summation Theorem . . . . .	108
A.10.3	Enzyme Kinetics ed Elasticity Coefficients . . . . .	108
A.10.4	Relazioni di connettività . . . . .	109
A.10.5	Scaling . . . . .	109
A.10.6	Sensitivity Analysis . . . . .	109
<b>B</b>	<b>Utilizzo e funzionalità di COBRA Toolbox</b>	<b>111</b>
B.1	Materiali . . . . .	111
B.1.1	Equipment . . . . .	111
B.1.2	Equipment setup . . . . .	112
B.2	Descrizione delle procedure e dei metodi utilizzati nell'analisi . . . . .	112
B.2.1	Procedure preliminari . . . . .	112
B.2.1.1	Inizializzazione del Toolbox . . . . .	112
B.2.1.2	Cambiare il risolutore in uso . . . . .	113
B.2.1.3	Test delle funzionalità del Toolbox . . . . .	113
B.2.2	Caricare un modello in Matlab . . . . .	113
B.2.3	Salvare un modello in un file . . . . .	114
B.2.4	Apportare delle modifiche ai modelli COBRA Toolbox . . . . .	114
B.2.4.1	Modificare i limiti delle reazioni . . . . .	114
B.2.4.2	Aggiungere nuove reazioni . . . . .	114
B.2.4.3	Rimuovere alcune reazioni . . . . .	115
B.2.4.4	Modificare il flusso obiettivo . . . . .	115
B.2.4.5	Stampare la formula di una reazione . . . . .	116
B.2.5	Visualizzazione del modello . . . . .	116

B.3	Gap filling . . . . .	116
B.3.1	Ricerca di metaboliti che partecipano ad un'unica reazione: . . . . .	117
B.3.2	Individuazione di tutti i gap del modello: . . . . .	117
B.3.3	Identificazione di candidati per colmare i gap: . . . . .	117
B.4	Ottimizzazione di flussi obiettivo tramite Flux Balance Analysis (FBA) . . . . .	118
B.4.1	FBA standard . . . . .	118
B.4.2	Geometric FBA . . . . .	119
B.4.3	Visualizzazione dei risultati . . . . .	119
B.5	Classificazione di geni e reazioni mediante pFBA . . . . .	120
B.5.1	Visualizzazione dei risultati . . . . .	120
B.6	Robustness Analysis . . . . .	121
B.7	Flux Variability Analysis (FVA) . . . . .	121
B.7.1	Visualizzazione dei risultati . . . . .	122
B.8	Analisi delle caratteristiche topologiche della matrice stechiometrica S: metabolite connectivity e reaction participation . . . . .	122
B.8.1	Visualizzazione della matrice S . . . . .	122
B.8.2	Determinazione del numero di reazioni a cui partecipa ciascun metabolita . . . . .	122
B.8.3	Reaction Participation . . . . .	123
B.8.4	Correlazione tra essenzialità delle reazioni e metabolite connectivity . . . . .	123
B.9	Sampling delle distruzione di flusso . . . . .	124
B.9.1	Riconoscere moduli di reazioni nelle reti metaboliche . . . . .	125
B.10	Simulazione di delezioni di geni . . . . .	125
B.10.1	Essential gene study . . . . .	126
B.10.2	Synthetic lethal study . . . . .	126
B.11	Metabolic engineering . . . . .	127
B.11.1	OptKnock . . . . .	127
B.11.2	OptGene . . . . .	127
B.11.3	GDLS . . . . .	128
B.12	Descrizione della struttura di un file SBML compilabile in COBRA . . . . .	129
<b>C</b>	<b>Condizioni iniziali del modello cinetico</b>	<b>131</b>
<b>D</b>	<b>File relativi alla costruzione dei modelli FBA utilizzati</b>	<b>132</b>

## Abstract

La tesi sviluppa la modellizzazione del pathway del metabolismo del triptofano nell'essere umano. Esso è un amminoacido essenziale, precursore degli ormoni serotonina e melatonina e di due acidi di interesse (acido quinolinico e xanturenico) ed è consigliato nella dieta agli individui soggetti a depressione, insonnia, difficoltà nella regolazione dell'insulina e agli anziani. Tramite l'uso di un modello cinetico (basato su equazioni differenziali) e la costruzione di due modelli FBA (basati su vincoli strutturali) si vuole osservare in che misura il triptofano contribuisca ad alterare i livelli di melatonina, serotonina, acido quinolinico ed acido xanturenico prodotti nell'organismo in soggetti sani e affetti da diabete e in che modo si possa eventualmente intervenire a livello genico per alterare i risultati ottenuti.

Nell'elaborato viene analizzata approfonditamente la Flux Balance Analysis (FBA), uno strumento di analisi che utilizza un'ottimizzazione lineare per determinare la distribuzione dei flussi di una rete metabolica in stato stazionario, attraverso la massimizzazione di una determinata funzione obiettivo e la definizione di vincoli dettati dalle equazioni di bilanciamento di massa e da limitazioni sui valori possibili dei flussi. Dalle equazioni di bilanciamento di massa che descrivono la rete viene creata una matrice (matrice stechiometrica), che descrive la dinamica dei flussi delle reazioni presenti nel pathway. Imponendo che, in stato stazionario, tali variazioni di flusso siano nulle, si ottengono quindi i vincoli strutturali utilizzati durante l'ottimizzazione. L'output di tale analisi è quindi una (non necessariamente unica) distribuzione di flussi nella rete, che permettono il raggiungimento del valore massimo (o minimo) della funzione obiettivo determinata in precedenza.

Tramite l'utilizzo di un modello cinetico del metabolismo del triptofano, su cui sono state eseguite analisi di sensibilità e di distribuzione dei flussi, e dei due modelli FBA, costruiti appositamente per questo studio e analizzati mediante gli strumenti messi a disposizione dalla FBA e dalle tecniche da essa derivate, è stato possibile analizzare il pathway di interesse in condizione di normalità e di limitata produzione di melatonina. Dall'integrazione dei risultati ottenuti con i diversi modelli è risultato che nei pazienti diabetici si verifica effettivamente una sottoproduzione di melatonina rispetto ai soggetti sani e che una situazione di limitata produzione di melatonina può portare ad una sovrapproduzione degli acidi quinolinico e xanturenico. Tramite un'analisi dei KnockOut (KO) è stato verificato che, in caso di malattia del diabete, il KO del gene TDO2, causando un aumento in termini di produzione di melatonina e una diminuzione di produzione degli acidi, può portare verso una situazione di normalità.

**Key words:** triptofano, rete metabolica, modello ODE, Flux Balance Analysis, melatonina, diabete.



# Chapter 1

## Introduzione

### 1.1 Il triptofano

L'**L-triptofano** (ovvero il triptofano) è uno degli otto amminoacidi essenziali del corpo umano: non può quindi essere sintetizzato a partire da carbonio e azoto e dev'essere assunto tramite la dieta. In natura il triptofano si ritrova nelle proteine alimentari (essenzialmente quelle di origine animale) e nella Griffonia (pianta tropicale africana della famiglia delle leguminose); ad esempio è contenuto nel bianco dell'uovo, nel merluzzo, nei semi di soia, sesamo e girasole, nel parmigiano, nel Cheddar, nel maiale, nelle carni bianche, nel manzo, ma anche nella farina di frumento e d'avena, nel cioccolato fondente, nel latte, nel riso, nella patata e nella banana. Nonostante esso si trovi nel nostro corpo in concentrazione minore rispetto a tutti gli altri amminoacidi (la quantità giornaliera raccomandata per una persona adulta è stimata tra i 250 mg/giorno e i 425 mg/giorno), esso è legato ad alcune funzioni importanti del nostro metabolismo.

I pazienti diabetici sono spesso soggetti alla depressione, motivo per cui talvolta viene somministrato loro triptofano, il quale dovrebbe aumentare la produzione di serotonina e melatonina. Un aumento di tale amminoacido però potrebbe causare anche un aumento della produzione di acido quinolinico ed acido xanturenico. La nostra attenzione in particolare si vuole focalizzare sul ruolo del triptofano nella sintesi della serotonina, della melatonina, del quinolinato e dell'acido xanturenico. Per capire l'importanza di tali metaboliti, riportiamo qui di seguito una breve descrizione, che può aiutarci nel comprendere il motivo per il quale siano stati scelti come nostro oggetto di studio.

#### Serotonina

Anche se è solo il 3% del triptofano ingerito che viene utilizzato per la sintesi della serotonina, questa è una delle più importanti pathways del triptofano. Nonostante la maggiorparte della serotonina si trovi nel tratto intestinale e ci sia una bassa concentrazione relativa di tale prodotto nel cervello (si stima infatti che solamente l'1% della quantità totale di triptofano ingerito venga utilizzato per la sintesi di serotonina proprio nel cervello), essa ha un ampio impatto come neurotrasmettitore e neuromodulatore ed è implicata in numerose patologie psichiatriche e processi psicologici.

La serotonina infatti (nota anche come "ormone del buonumore", 5-idrossitriptamina o 5-HT) è un neurotrasmettitore <sup>1</sup> sintetizzato nel cervello e in altri tessuti. Essa è coinvolta in importanti e numerosi funzioni biologiche, non ancora tutte chiare, poichè, interagendo con vari e specifici recettori, produce effetti diversi in base alla parte del corpo considerata. La serotonina agisce nelle cellule enterocromaffini, nelle piastrine, nel sistema cardiovascolare, nel tratto gastrointestinale e nel sistema nervoso centrale.

La deplezione del triptofano, ottenuta con la riduzione o l'eliminazione dell'amminoacido dalla dieta, può ridurre la biosintesi di serotonina. Un'abbondante assunzione di triptofano può portare invece ad una sovrapproduzione dell'ormone.

#### Melatonina

La melatonina (N-Acetil-5-Methoxytriptofano) è un ormone lipo-idrosolubile prodotto nel pathway triptofano/serotonina, che regola i ritmi diurni e influenza sia i sistemi riproduttivo e immunitario, sia i processi

---

<sup>1</sup>**neurotrasmettitore**: sostanza che media la propagazione dell'impulso nervoso tra due strutture collegate da una sinapsi chimica.

digestivi e la motilità gastrointestinale. I livelli di circolo di questo ormone variano nell'arco della giornata: si raggiunge il culmine durante la notte, al mattino scende e rimane basso tutto il giorno. Grazie alla secrezione di questo ormone, sulla base dell'alternarsi della luce e del buio, l'organismo si organizza sui ritmi di sonno e di veglia. La produzione di melatonina diminuisce drasticamente con l'età, fino a diventare quasi insignificante nelle persone di età superiore ai sessanta anni.

Alcuni recenti studi<sup>2</sup> hanno dimostrato un ruolo importante di questo ormone (in particolare del suo recettore MT1) nella regolazione del metabolismo glucidico: pare che l'attivazione recettoriale possa minimizzare il rischio di diabete mellito di tipo II, migliorando la sensibilità all'insulina. Al contrario, a quanto pare, un livello notturno basso di secrezione della melatonina può essere associato ad un rischio maggiore dell'insorgere del diabete di tipo due. La presenza dei recettori dell'ormone nelle isole pancreatiche, suggerisce che la melatonina possa anche regolare il rilascio dell'insulina e, di conseguenza, il livello di glucosio nel sangue; l'insulina infatti viene regolata anche dal ritmo giorno-notte.

### **Acido quinolinico**

L'acido quinolinico è un prodotto a valle della via della chinurenina, che metabolizza il triptofano (questa via metabolica sarà descritta più nel dettaglio nel seguito). In condizioni fisiologiche normali, tale acido può modulare alcuni eventi locali nel sistema nervoso centrale. Tuttavia, se presente in quantità eccessiva, esso ha un effetto neurotossico potente: può essere coinvolto in molti disturbi, tra cui disturbi psichiatrici o neurodegenerativi nel cervello.

L'acido quinolinico agisce come un agonista del recettore NMDA, che consiste in un recettore di glutammato, dispositivo molecolare predominante per controllare la plasticità sinaptica e la funzione di memoria.

### **Acido xanturenico**

L'acido xanturenico (acido 4-8 diidrossichinolin-carbonico) è un metabolita secondario del triptofano, derivante da una transaminazione ossidativa dell'acido 3-idrossifenil-piruvico, che spontaneamente si trasforma in acido xanturenico.

Formatosi nell'organismo oppure fornito dall'esterno, tale acido non subisce ulteriori trasformazioni e viene eliminato come tale attraverso le urine o in forma di acido libero oppure in forma glicuronide coniugato alla serina.

Per quanto riguarda gli essere umani, l'acido xanturenico è sospettato essere un agonista endogeno per il gruppo II, costituito dai recettori metabotropici del glutammato. Se si osserva un'abbondanza di xanturenico nelle urine, può essere indice di carenza di vitamina B6 e di alcuni disturbi del catabolismo del triptofano.

#### **1.1.1 Obiettivi della tesi**

Il nostro studio quindi sarà un'analisi riguardante il metabolismo del triptofano, allo scopo di capire in che misura esso contribuisca ad alterare i livelli dei metaboliti appena descritti nell'organismo umano. In particolare ci occuperemo di modellizzare tramite diverse tecniche il pathway del metabolismo di questo amminoacido, per poter osservare:

- se una diminuzione di produzione di serotonina e melatonina può portare ad un aumento della produzione degli acidi quinolinico e xanturenico;
- se nei pazienti diabetici si osserva effettivamente una sottoproduzione di melatonina e una sovrapproduzione degli acidi quinolinico e xanturenico;
- nell'eventualità del verificarsi dei casi precedenti, se e come si può agire a livello genico per evitare tali eventuali sovrapproduzioni e cercare di aumentare il livello di melatonina nel sangue, così da riportarsi ad una situazione di normalità.

Per cercare di dare una risposta alle nostre domande, analizzeremo un modello cinetico del metabolismo del triptofano nell'essere umano, descritto da equazioni differenziali. Poiché i parametri cinetici e i dati di espressione genica richiesti per poter effettuare tale analisi non sono facilmente reperibili nel caso dell'uomo, i risultati ottenuti da questa analisi non potranno essere completi e totalmente soddisfacenti. Ci serviremo

---

<sup>2</sup>ricerca condotta dal Brigham and Women's Hospital (<http://www.italiasalute.it/551/Poca-melatonina-aumenta-rischio-di-diabete.html>)

perciò anche di un altro strumento di analisi: la Flux Balance Analysis (FBA), che verrà descritta nelle sezioni successive e che richiede un minor numero di informazioni per essere eseguita. Come vedremo in seguito, essa richiede solamente una conoscenza del modello in esame basato sulle equazioni di bilanciamento di massa che descrivono le reazioni della rete e utilizza una semplice ottimizzazione lineare per rispondere a diverse domande di nostro interesse.

Poichè questo strumento di analisi si sta rivelando uno strumento molto utile in ambito della biologia sintetica, scopo della nostra tesi è anche quello di approfondire questa metodologia di studio, andando a fondo nella descrizione di quali siano le sue funzionalità e le possibilità che essa offre. In particolare, poichè la FBA solitamente viene utilizzata per lo studio del funzionamento di organismi unicellulari, si vuole esplorare in che modo essa possa anche risultare utile per lo studio del metabolismo di un amminoacido in un organismo più complesso, quale è l'essere umano, per cui è più complicato reperire informazioni inerenti ai parametri cinetici e per il quale risulterebbe molto interessante poter utilizzare questo nuovo approccio.

### 1.1.2 Uso del triptofano nella dieta

Una dieta ricca di triptofano viene attualmente consigliata ed utilizzata in diverse situazioni. In quanto precursore degli ormoni della serotonina e della melatonina, la supplementazione di triptofano aumenta le concentrazioni di ambedue le sostanze ed è un importante sussidio terapeutico negli stati depressivi e nei disturbi del sonno.

Poiché dopo i trenta/quaranta anni la produzione di melatonina cala (intorno ai sessant'anni è circa la metà di un ventenne), le persone anziane tendono a dormire meno: è questa una delle ragioni per cui anche agli anziani viene spesso suggerita una dieta ricca di triptofano.

Inoltre pare che la disponibilità di triptofano alimenti influenzi le funzioni cognitive.

Gli effetti più interessanti del triptofano stanno nel controllo del tono d'umore: deplezioni acute sono collegate alla comparsa di sintomi depressivi sia nell'adulto sano che nel paziente; viceversa la supplementazione porta giovamento ai pazienti affetti da sintomi depressivi medi o lievi, potenziando invece l'azione dei farmaci antidepressivi, nei pazienti affetti da depressione maggiore.

Una dieta ricca di triptofano infatti aumenta il senso di benessere e migliora lo stato d'animo; essa comporta una migliore disposizione al sonno e la modulazione dell'appetito, grazie alla produzione di serotonina. Tale amminoacido infatti sembra avere degli effetti benefici anche in pazienti soggetti ad obesità [42]. Il triptofano entra nel cervello secondo un rapporto determinato dalla sua concentrazione relativa agli altri amminoacidi neutri: questo rapporto TRP/LNAA (amminoacidi neutri) indica quindi la disponibilità di tale amminoacido nel cervello e determina la secrezione di serotonina. L'insulina resistenza di una persona obesa, che causa una diminuzione della caduta nel plasma della concentrazione degli amminoacidi neutri (l'obeso risponde di meno ai normali livelli di insulina, perciò l'insulina secreta dopo un pasto non è sufficiente a regolare le necessarie quantità di glucosio e amminoacidi nelle cellule muscolari) bloccherebbe così il normale aumento del rapporto TRP/LNAA, causato da una ingestione di carboidrati, causando una minore disponibilità di triptofano nel cervello e una conseguente diminuzione di serotonina rilasciata dopo un pasto di carboidrati. La fame di carboidrati, descritta in alcune persone obese, potrebbe essere interpretata come l'incapacità del cervello di produrre abbastanza serotonina per sopprimere l'apporto di carboidrati.

Nonostante non ci sia quindi ancora una chiara e definita terapia basata sull'utilizzo del triptofano, è ormai assodato che una dieta ricca di tale amminoacido può avere effetti benefici in diverse situazioni. Essa dunque viene suggerita in tutti quei casi in cui compaiono sintomi quali depressione, insonnia, bassa sensibilità all'insulina. Anche ai soggetti diabetici dunque viene spesso suggerita una dieta ricca di triptofano.

### 1.1.3 Struttura della tesi

Al fine di accompagnare il lettore nella comprensione dello studio effettuato durante la stesura della tesi, abbiamo scelto di strutturare l'elaborato nel seguente modo:

1. Un capitolo introduttivo comprende le nozioni necessarie per conoscere ed acquisire le conoscenze biologiche inerenti ai concetti di metabolismo e di biomassa in senso generale e al caso specifico del metabolismo del triptofano nelle cellule dell'organismo umano.
2. Nel secondo capitolo verranno descritti i concetti di rete e pathway metabolico, al fine di comprendere la modellizzazione necessaria per effettuare le nostre analisi. Verranno in tal sede discussi vantaggi e svantaggi dei diversi tipi di modellizzazione possibili.

3. Il terzo capitolo è relativa al modello cinetico del metabolismo del triptofano. In particolare verrà descritta la tecnica con la quale è stato realizzato e che condizioni iniziali sono state scelte. Successivamente saranno descritte le tipologie di analisi effettuate, i dati utilizzati e i risultati ottenuti tramite questo tipo di modello.
4. Il capitolo numero quattro è invece relativa ai modelli basati su vincoli, utilizzati anch'essi per modellare il metabolismo del triptofano. Poichè è nostro scopo anche introdurre questo nuovo metodo di analisi da un punto di vista teorico, in questa sezione verranno descritte le varie tipologie di analisi realizzabili e gli strumenti utilizzabili tramite questo tipo di modellizzazione, con un'attenzione particolare agli algoritmi che ne permettono l'implementazione e alla loro comprensione.
5. Il quinto capitolo introduce il software COBRA Toolbox, utilizzato per l'analisi dei modelli basati su vincoli e ne evidenzia le sue funzionalità. Per una sua descrizione più completa si farà invece riferimento all'Appendice B.
6. Nella sesta parte verranno invece analizzati due diversi modelli basati su vincoli, costruiti appositamente per il nostro studio. Ne verrà descritta la costruzione, le analisi effettuate e i risultati ottenuti. Verranno inoltre confrontati tali risultati con quelli ottenuti precedentemente tramite il modello cinetico, al fine di poter arrivare a dare una risposta alle nostre domande iniziali, che hanno motivato la nostra tesi.
7. Le conclusioni della tesi sono riportate nel settimo capitolo dell'elaborato. In questa sezione viene riportato tutto il percorso che ha portato alla stesura della tesi.
8. In conclusione sono presenti diverse Appendici, che contengono informazioni utili per quanti vogliono cimentarsi nell'utilizzo di questi strumenti. In particolare:
  9. Descrizione dei tasks del software COPASI, utilizzato per analizzare il modello cinetico;
  10. Utilizzo e funzionalità di COBRA Toolbox, utilizzato per effettuare le analisi dei modelli basati su vincoli;
  11. Condizioni iniziali utilizzate nel modello cinetico;
  12. File relativi alla costruzione dei modelli FBA utilizzati.

## 1.2 Metabolismo e produzione di biomassa nelle cellule

Il metabolismo è l'insieme delle trasformazioni chimiche che avvengono all'interno di un organismo e che sostengono i processi vitali. L'insieme delle reazioni, catalizzate da enzimi, permette all'organismo di crescere, riprodursi, autoripararsi e rispondere alle necessità ambientali.

Una reazione chimica è un processo nel quale uno o più composti (Reagenti) si trasformano in nuovi composti (Prodotti). Tale trasformazione deve obbedire alla legge della conservazione di massa, ovvero la massa di ciascun elemento presente nelle molecole dei reagenti deve coincidere con quella dello stesso elemento nelle molecole dei prodotti. Questo bilanciamento viene effettuato definendo i coefficienti stechiometrici (ossia il numero di molecole di ogni composto (reagente e/o prodotto) che partecipa alla reazione) in modo che il numero di atomi di ogni specie atomica coinvolta sia conservato. Così viene garantito il bilancio di massa.

Il metabolismo di un individuo quindi riflette le trasformazioni di materia ed energia, attuate sia per il mantenimento di strutture esistenti, sia per la produzione di nuova biomassa.

Le due classi di energia biologicamente importanti su cui si basa il metabolismo sono le seguenti:

- energia chimica-potenziale: liberata dall'ossidazione di composti a base di carbonio o idrogeno;
- energia cinetica a livello molecolare.

Negli esseri animali questi tipi di energia sono correlati in modo abbastanza complesso: tutto il lavoro ottenuto dalla conversione dell'energia chimica-potenziale è inefficiente e produce calore metabolico (tutte le reazioni chimiche, comprese quelle biochimiche cataboliche e di biosintesi, producono variazioni nel contenuto calorico di un sistema). A sua volta, l'efficienza metabolica è proporzionale alla temperatura dell'organismo. È quindi impossibile separare i problemi di bilancio termico con quelli legati all'acquisizione e all'allocazione di energia chimica potenziale.

Solitamente il metabolismo viene suddiviso in due tipi di processi (Figura 1.1 [28]):

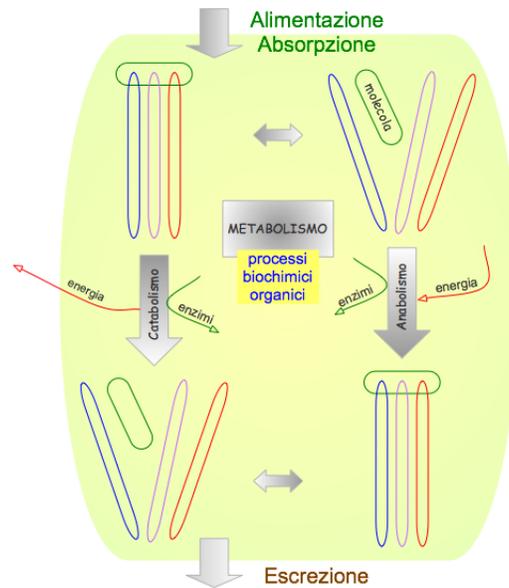


Figure 1.1: [28] *Processi del metabolismo*

- **anabolismo**: insieme di reazioni che portano alla formazione di materia cellulare (crescita e riproduzione delle cellule). In pratica l'anabolismo permette di produrre molecole complesse a partire da molecole più semplici. Tale processo consuma energia e dunque necessita di essere accoppiato con reazioni che forniscono energia libera;
- **catabolismo**: insieme delle reazioni di demolizione (ossidazione) di nutrienti forniti dall'esterno. Si occupa quindi della degradazione di molecole complesse in molecole più semplici. Tale processo libera energia tramite la rottura di legami molecolari ad alta energia.

Più del 70% del consumo energetico totale umano è dovuto al metabolismo basale<sup>3</sup>, circa il 20% è consumato nell'attività fisica e infine solo il rimanente 10% per la termogenesi o la digestione.

Ci sono però numerosi meccanismi che possono far variare i tassi metabolici, alcuni dei quali possono avere effetto immediato e altri che intervengono in periodi di tempo molto più lunghi. Alcuni esempi:

- l'**acclimatamento**: gli organismi regolano il proprio metabolismo in reazione alle variazioni ambientali, quali temperature, umidità, ecc., per mantenere costante la performance;
- la **plasticità fenotipica**: un organismo può essere più o meno capace di modificare il proprio fenotipo (a livello morfologico, fisiologico, comportamentale o biochimico) in seguito a variazioni ambientali;
- l'**evoluzione**: un'intera specie può avere la capacità di adattarsi a variazioni dell'ambiente di periodo molto lungo.

I fattori che influenzano il metabolismo sono sia endogeni (taglia corporea, attività, stato di salute e stress), sia esogeni (temperatura, concentrazione di ossigeno, disponibilità di nutrimento e luce).

In particolare, la temperatura influenza tutte le reazioni chimiche, anche quelle che avvengono nei sistemi biologici e che sono catalizzate da enzimi, perciò sia le singole reazioni biochimiche, sia il metabolismo dell'intero organismo, sono funzioni della temperatura.

Il trasporto delle specie chimiche attraverso le membrane cellulari può avvenire secondo tre diversi meccanismi (Figura 1.2):

1. **diffusione passiva**, secondo cui la specie si muove dalla zona ad alta concentrazione verso quella a bassa concentrazione attraverso la membrana. Questa è una diffusione spontanea;

<sup>3</sup>**metabolismo basale**: quantità di energia impiegata in condizioni di neutralità termica, dal soggetto sveglio, ma in uno stato di totale rilassamento fisico e psichico, a digiuno da almeno dodici ore. Consiste quindi nel minimo dispendio energetico necessario a mantenere le funzioni vitali e lo stato di veglia.

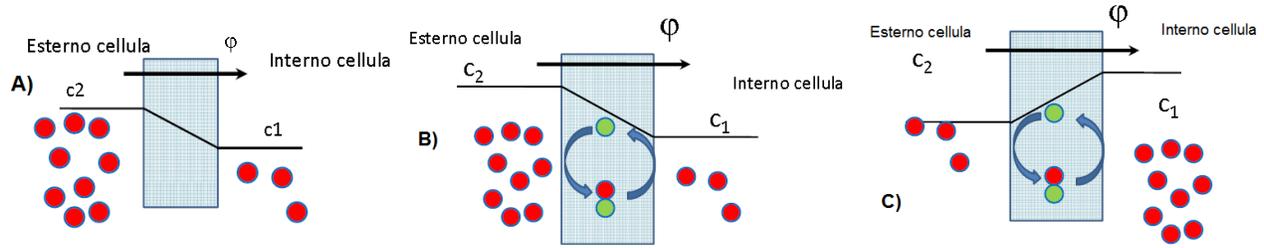


Figure 1.2: A) *Diffusione passiva*; B) *Diffusione facilitata*; C) *Trasporto attivo*

2. **diffusione facilitata**, secondo cui, a causa della struttura della membrana plasmatica (che circonda tutte le cellule e gli organuli degli eucarioti), non tutte le specie chimiche possono penetrare nella membrana con la stessa facilità;
3. **trasporto attivo**, secondo cui la specie si muove contro il suo gradiente chimico, sfruttando una molecola di trasporto e necessita quindi di energia libera.

Il metabolismo permette la sintesi di **biomassa**: la crescita cellulare comporta consumo di substrato<sup>4</sup>, che fornisce energia e materiale di partenza per la sintesi di biomassa. Si può definire perciò un

Fattore di resa in biomassa = rapporto tra la quantità di biomassa formata e la quantità di substrato utilizzato:

$$Y_{X/S} = \frac{\Delta X}{\Delta S}.$$

Lo studio della formazione dei prodotti del metabolismo è complicato dal fatto che tali prodotti, oltre a poter essere numerosi e molto vari, possono essere liberati all'esterno della cellula o accumulati nella cellula stessa. In alcuni casi i prodotti sono intermedi, altre volte si tratta di prodotti finali del catabolismo. Inoltre si è soliti differenziare tra:

- metaboliti primari: i prodotti essenziali alla vita stessa della cellula;
- metaboliti secondari: metaboliti costituiti da molecole più complesse che svolgono speciali funzioni per la cellula (es. antibiotici per autodifesa).

I metaboliti primari sono i prodotti di partenza per la sintesi dei metaboliti secondari.

Se il prodotto in esame è il risultato del metabolismo energetico primario, esiste una relazione pressoché costante tra il consumo di substrato, l'aumento di biomassa e la formazione di prodotto:

$$Y_{P/S} = \frac{\Delta P}{\Delta S};$$

$$Y_{P/X} = \frac{\Delta P}{\Delta X}.$$

Nonostante le cellule utilizzino molto efficientemente l'energia, una parte dell'energia del substrato viene rilasciata come calore; si definisce quindi un fattore di resa:

$$Y_{\Delta} = \frac{\text{Biomassa prodotta [g]}}{\text{calore rilasciato [Kcal]}}.$$

Per misurare il metabolismo dell'intero organismo si possono utilizzare diverse tecniche:

- calorimetria (diretta e indiretta);
- respirometria (in stato stazionario e in flusso continuo);

<sup>4</sup>**substrato**: molecola sulla quale agisce un enzima.

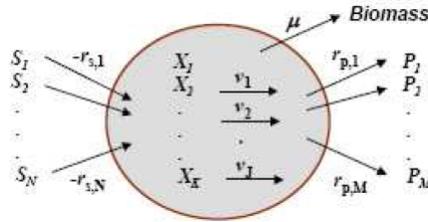


Figure 1.3: **Generica rappresentazione di reazioni in una rete metabolica.**  $N$  substrati ( $S_1, \dots, S_N$ ) entrano nella cellula ( $r_{s,i}$ : flussi entranti) e vengono trasformati in  $M$  prodotti metabolici ( $P_1, \dots, P_M$ ) ( $r_{p,i}$ : flussi uscenti) attraverso  $K$  metaboliti intracellulari ( $X_1, \dots, X_K$ ). La conversione avviene attraverso  $J$  reazioni intracellulari, i cui flussi sono  $v_1, \dots, v_J$ .  $\mu$  rappresenta il tasso di crescita di biomassa.

- sviluppo di  $\text{CO}_2$ ;
- traccianti adioisotopici (doppia marcatura e acqua a doppia marcatura);
- metabolomica ambientale.

### 1.3 Processi metabolici del triptofano

In Figura 1.4 ([41]: database di Kegg) è rappresentato il metabolismo del triptofano. Le reazioni colorate di verde sono relative al metabolismo di tale amminoacido nell'essere umano.

L'assorbimento del triptofano avviene nell'intestino tenue; viene poi attivamente trasportato nel cervello da una proteina trasportatrice<sup>5</sup>, comune per il trasporto di acidi neutri e amminoacidi a catena ramificata di grandi dimensioni. Perciò il livello di triptofano nel cervello è influenzato non solo dalla sua concentrazione plasmatica, bensì anche dalle concentrazioni plasmatiche degli altri amminoacidi che competono per il carrier di assorbimento cerebrale. Ad esempio, è noto che alte concentrazioni di leucina bloccano l'assorbimento del triptofano.

Nell'uomo, il triptofano viene metabolizzato mediante due vie non proteiche:

1. la via dei metossindoli;
2. la via della chinurenina.

#### 1.3.1 Via metossindolica

I due fattori limitanti la via del metossindolo sono la disponibilità di triptofano (substrato<sup>6</sup>) (infatti meno del 5% del metabolismo del triptofano segue questa via) e l'idrossilazione del triptofano, che avviene mediante l'enzima triptofano-idrossilasi<sup>7</sup> (TPH) con formazione di 5-idrossitriptofano (fase 1) in Figura 1.5). Questo derivato amminoacidico (presente negli alimenti in quantità irrisorie), viene a volte integrato artificialmente nell'organismo come antidepressivo, aiuto contro l'insonnia ed inibitore dell'appetito. Negli esseri umani, l'enzima TPH è codificato sui cromosomi 11 e 12, poichè esistono due diversi enzimi omologhi TPH1 e TPH2:

- TPH1: espressa principalmente nelle cellule enterocromaffili del tratto intestinale, nella pelle, nella ghiandola pineale e nel sistema nervoso centrale;
- TPH2 (isoforma predominante): espressa unicamente in cellule neuronali.

Il composto 5-idrossitriptofano viene poi decarbossilato a 5-HT (substrato per la sintesi di melatonina) (fase 2)). La sintesi di melatonina è dipendente dalla N-acetilazione<sup>8</sup> della 5-HT (fase 3)), che porta alla formazione di N-acil-serotonina (NAS) e dalla successiva O-metilazione<sup>9</sup> in melatonina (fase 4)).

<sup>5</sup>**proteina trasportatrice**: proteina che permette il trasporto di ioni o di piccole molecole organiche attraverso le membrane cellulari.

<sup>6</sup>**substrato**: molecola sulla quale agisce un enzima.

<sup>7</sup>**triptofano-idrossilasi**: enzima appartenente alla classe delle ossidoreduttasi.

<sup>8</sup>**acetilazione**: reazione chimica in cui viene legato un gruppo acetile ad una molecola.

<sup>9</sup>**metilazione**: introduzione di uno o più gruppi metilici; è uno dei principali meccanismi di modificazione post-trascrizionale delle proteine.



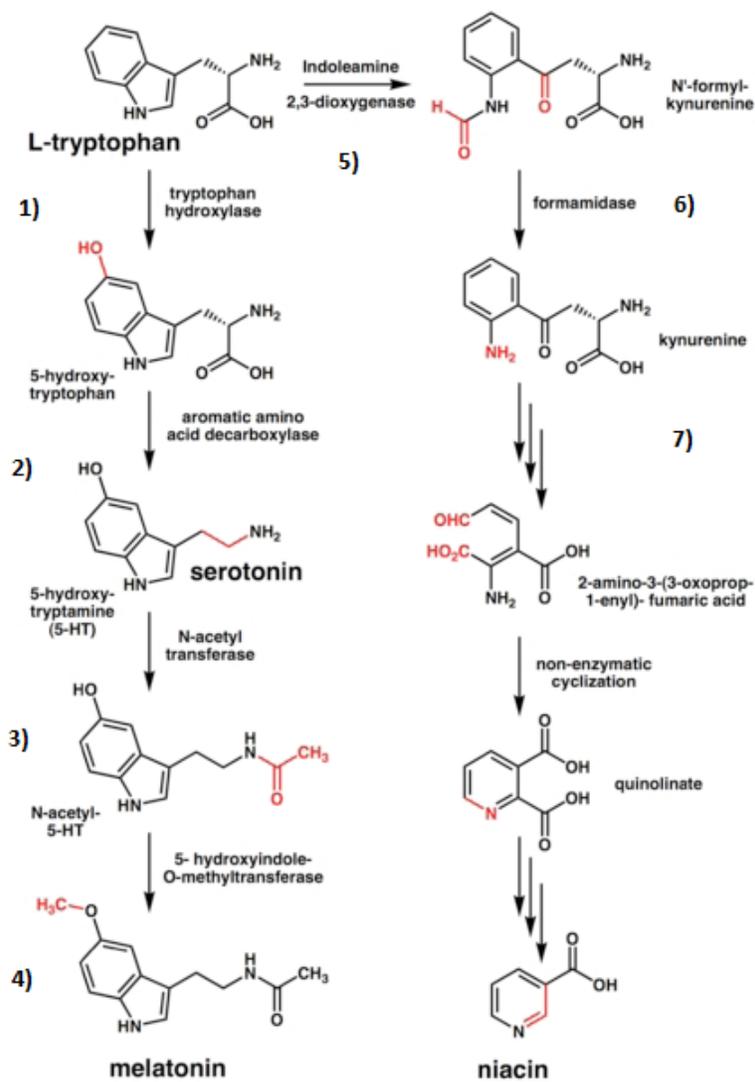


Figure 1.5: **Metabolismo del triptofano:** 1) idrossilazione del triptofano (mediante l'enzima triptofano-idrossilasi), con formazione di 5-idrossitriptofano; 2) il 5-idrossitriptofano viene decarbossilato a 5-HT; 3) N-acetilazione della 5-HT, con formazione di N-acil-serotonina; 4) O-metilazione in melatonina; 5) clivaggio dell'anello indolico del triptofano, con formazione di N-formilchinerina; 6) formazione di chinurenina; 7) metabolismo della chinurenina.

Gli enzimi dose-limitanti in questa fase sono le IDO (indolamina 2,3-diossigenasi) e le TDO (triptofano 2,3-diossigenasi).

La chinurenina inibisce il trasporto di triptofano attraverso la barriera ematoencefalica<sup>11</sup>, stimola l'attività delle IDO e si è dimostrato che induce ansia in modelli animali.

**Metabolismo della chinurenina** La chinurenina prodotta nella prima tappa viene poi metabolizzata attraverso due ulteriori vie distinte, che competono per la chinurenina come substrato:

1. la via chinurenina-acido chinurenico (fase b) Figura 1.6);
2. la via chinurenina-nicotinamide adenina dinucleotide (fasi c) e d) Figura 1.6).

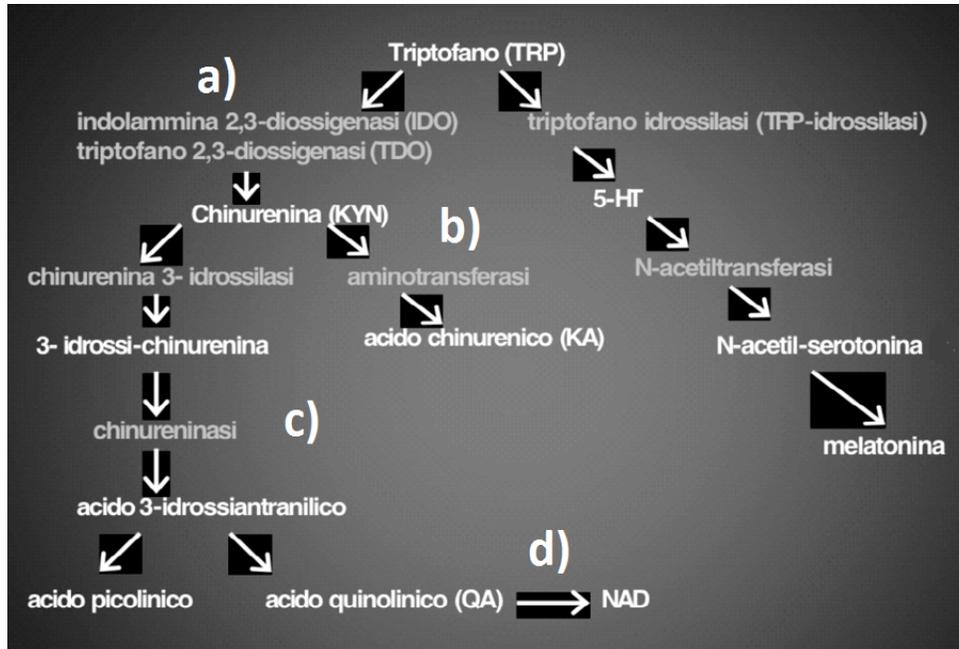


Figure 1.6: [27] **Via metossindolica e della chinurenina.** a) conversione del triptofano in chinurenina; b) via chinurenina-acido chinurenico; c) produzione di acido quinolinico e picolinico; d) produzione di acido nicotinico.

La prima è regolata dalla chinurenina-aminotransferasi, il principale enzima coinvolto nella sintesi di acido chinurenico nel cervello. Tale acido è l'unico antagonista endogeno conosciuto dei recettori della N-metil-D-aspartato, quindi potrebbe avere un effetto antidepressivo e psicotomimetico (imitazione di malattie mentali).

Inoltre l'acido chinurenico può agire a livello della via della melatonina e della chinurenina, inibendo la 5-HT-N-acetilazione (tappa limitante della biosintesi di melatonina).

A causa della sua affinità per certi recettori piuttosto che per altri, può inoltre contribuire alla comparsa di deficit cognitivi, legati alla depressione, alla schizofrenia e alla demenza.

La seconda via invece produce agonisti dei recettori NDMA (acido quinolinico e picolinico) e generatori di radicali liberi (3-idrossichinurenina e acido-3-idrossiantranilico). Un aumento di questi agonisti può attivare il sistema glutammatergico, attualmente associato alla depressione. È in questa seconda fase che viene prodotto l'acido nicotinico (NAD) (fase d)), che può essere amminato a nicotinammide e quindi risultare utile per la formazione di coenzimi<sup>12</sup> contenenti nicotinammide.

<sup>11</sup>**barriera ematoencefalica:** regola il passaggio sanguigno di sostanze chimiche da e verso il cervello, proteggendo il sistema nervoso da avvelenamenti ed intossicazioni. È costituita da due meccanismi cellulari: in primo luogo le cellule delle pareti dei capillari sanguigni del cervello formano giunzioni strette, compattando le pareti, in secondo luogo gli astrociti (cellule speciali gliali) avvolgono con i loro filamenti gli stessi capillari cerebrali, esercitando una funzione di filtraggio.

<sup>12</sup>**coenzima:** sostanza organica termostabile, non proteica, di composizione chimica relativamente semplice, necessaria per l'azione biologica di numerose proteine.

Gli acidi quinolico e picolinico incrementano l'ansia, stimolando la sintesi di ossidonitricosintasi-inducibile. Inoltre, assieme alla 3-idrossichinurenina e all'acido 3-idrossiantranilico, possono aumentare la perossidazione dei lipidi e attivare la cascata dell'acido arachidonico, causando un aumento della sintesi dei fattori dell'infiammazione: le prostaglandine<sup>13</sup> (mediante le ciclossigenasi, l'inibizione delle quali comporta il blocco della produzione di acido chinurenico e quindi ha un effetto antidepressivo e antipsicotico) e i leucotrieni<sup>14</sup> (mediante la arachinodato 5-lipossigenasi, coinvolta nella depressione e nell'aterosclerosi).

Anche l'attivazione delle IDO, con produzione di cataboliti<sup>15</sup> del triptofano (inclusi chinurenina, acido chinurenico e acido quinolico) è un meccanismo mediante il quale l'infiammazione può indurre processi neurodegenerativi: l'acido chinurenico agisce sul sistema glutammatergico, riducendo il rilascio di glutammato e quindi di dopamina (precursore della noradrenalina, con funzione di neurotrasmettitore nel sistema nervoso centrale); l'acido quinolinico invece causa la distruzione dei neuroni post-sinaptici e altera il metabolismo di dopamina, colina, acido  $\gamma$ -amminobutirrico ed encefalina.

---

<sup>13</sup>**prostaglandine:** gruppo di acidi grassi a 20 atomi di carbonio, la cui struttura di base è quella dell'acido prostanico, con un ciclopentano a due catene laterali idrocarburiche.

<sup>14</sup>leucotriene: sostanza a reazione lenta dell'anafilassi, che provoca contrazioni potenti della muscolatura dei bronchioli.

<sup>15</sup>**catabolita:** prodotto dal catabolismo.

## Chapter 2

# Modellizzazione del pathway del metabolismo del triptofano

Per eseguire le analisi descritte nella parte introduttiva e capire quindi come funzioni il metabolismo del triptofano nell'essere umano, per studiarne la sua influenza nella produzione di serotonina e melatonina oltre che di acido quinolinico e acido xanturenico, è opportuno quindi scegliere una metodologia e degli strumenti adatti. Per analizzare i sistemi biologici, uno degli strumenti più utili e diffusi sono le reti metaboliche, chiamate anche pathway metabolici; prima di proseguire è quindi necessario introdurre brevemente tali reti, per poter meglio capire il significato dei risultati delle analisi che verranno in seguito descritte.

### 2.1 Rete (o pathway) metabolica

Un pathway metabolico è rappresentato da una sequenza di reazioni biochimiche sperimentalmente osservabili e/o misurabili che connettono uno specifico set di metaboliti in entrata e in uscita.

La ricostruzione di un pathway metabolico consiste quindi in un processo durante il quale i vari componenti del pathway di un sistema biologico (id est geni, proteine, reazioni e metaboliti), che intervengono nell'attività metabolica, vengono identificati, catalogati e interconnessi tra loro per formare una rete. La vasta disponibilità di sequenze proteiche di diversi organismi, ottenute tramite l'analisi bioinformatica, è un punto di partenza molto utile per eseguire tale ricostruzione.

L'analisi dei flussi metabolici è uno strumento molto utilizzato per riuscire ad associare il genotipo di una cellula con il suo fenotipo: in particolare la constraint-based analysis di reti metaboliche ricostruite si è rivelata molto utile in varie applicazioni, come ad esempio l'ingegneria metabolica, la previsione dei risultati in seguito all'eliminazione di geni, l'identificazione delle dosi di farmaci e il raffinamento di reti metaboliche contenenti dei 'gap'.

Per studiare un metabolismo è possibile considerare due diversi punti di vista: dinamico e strutturale, il primo dei quali necessita di equazioni differenziali, mentre il secondo utilizza modelli basati sui grafi o su vincoli (constraint based).

Con un modello di tipo cinetico, basato quindi su equazioni differenziali (ODE), le analisi che si possono effettuare sono più complete e permettono di ottenere dei risultati più significativi. Infatti, dopo aver definito delle condizioni iniziali opportune, si possono eseguire delle analisi nel tempo, per osservare come evolve il sistema al trascorrere del tempo, oltre che ad analisi in stato stazionario (in una situazione quindi di equilibrio) e ad analisi di sensibilità per osservare come i diversi parametri e/o componenti della rete si influenzino tra di loro. È per questi motivi che inizialmente ci serviremo di un modello cinetico del metabolismo del triptofano nell'essere umano, basato su equazioni differenziali.

Purtroppo, per poter costruire un modello ODE è necessaria la conoscenza di molti parametri cinetici ed espressioni geniche inerenti al funzionamento del sistema nell'organismo di interesse. Per quanto riguarda l'essere umano, questi dati non sono totalmente disponibili: i risultati che otterremo perciò saranno solamente parziali e, in seguito, ci serviremo anche di modelli basati su vincoli, per completare l'analisi da un punto di vista strutturale.

### 2.1.1 Pro e contro del modello cinetico

Il pathway del metabolismo del triptofano è attivo in tutte le cellule (e quindi in tutti i tessuti del nostro organismo): in ogni tessuto però l'amminoacido viene metabolizzato in modo differente.

Inoltre tale pathway è molto complesso e per poterlo modellizzare occorre utilizzare un numero significativo di variabili di stato. In aggiunta, per quanto appena evidenziato, in ogni cellula relativa a diversi tessuti occorre utilizzare dei parametri diversi e delle concentrazioni diverse.

Per questi motivi il modello cinetico del metabolismo del triptofano è caratterizzato da alcuni aspetti positivi e negativi, che è bene porre in evidenza fin da subito, al fine di comprendere al meglio i risultati che si otterranno.

- Vantaggi dell'uso di un modello cinetico (basato su equazioni differenziali):

1. possibilità di effettuare analisi nel tempo;
2. possibilità di eseguire analisi di sensitività.

- Svantaggi dell'uso di un modello cinetico (basato su equazioni differenziali):

1. il modello così costruito è tessuto-dipendente, perciò tessuti diversi necessitano di modelli diversi;
2. è necessario essere a conoscenza dei parametri cinetici caratteristici delle diverse reazioni nei diversi tessuti;
3. occorre definire delle condizioni iniziali del sistema, che non sempre sono chiare nel caso dell'essere umano;
4. a causa della grande quantità di informazioni richieste, è spesso necessario limitarsi all'analisi di modelli di piccole dimensioni.

### 2.1.2 Come interpretare la rappresentazione di una reazione in un pathway

Prima di procedere con la descrizione dei modelli, è bene capire come vanno interpretate le reazioni in modelli metabolici di questo tipo.



Figure 2.1: *Rappresentazione delle reazioni in un pathway metabolico*

La Figura 2.1 può esserci d'aiuto nella comprensione: il flusso di reazione viene rappresentato da una freccia che lega i due compartimenti caratterizzanti i metaboliti. La freccia è uscente dal metabolita che viene utilizzato nella reazione ed entrante nel metabolita prodotto. In questa rappresentazione vengono visualizzati solamente i metaboliti prodotti direttamente dal pathway (non sono visibili ad esempio altri componenti, quali ossigeno, acqua, ecc.) che non sono specifici del pathway in esame.

Ciascun flusso, se non è un flusso spontaneo, viene regolato da un enzima, frutto della traduzione di un tratto di RNA trascritto.

## Chapter 3

# Modello cinetico del metabolismo del triptofano

Per quanto riguarda il modello cinetico che verrà utilizzato nelle analisi seguenti, faremo riferimento ad un paper [2], pubblicato online il 15 Ottobre 2013, che contiene la descrizione del modello di interesse, che verrà descritto qui di seguito.

Per eseguire le analisi di interesse, è stato utilizzato il software COPASI, disponibile online<sup>1</sup> e descritto in Appendice A.

### 3.1 Descrizione del problema

Nel paper di riferimento, viene presentato un modello cinetico della rete del metabolismo del triptofano dell'essere umano, basato sui dati cinetici esistenti per tutte le reazioni enzimatiche e di trasporto e sull'espressione genetica, inerenti, in particolare ai tessuti del fegato e del cervello. In Figura 3.1 possiamo osservare il modello compartimentale descritto, la cui costruzione risulterà più chiara nel seguito.

#### COME INTERPRETARE UNA REAZIONE:

$$GENE_{REGOLATORE}metabolita_{input}$$

- frecce continue ( $\longrightarrow$ ): reazione modellizzata;
- frecce tratteggiate ( $\dashrightarrow$ ): reazione NON modellizzata.

Come si può osservare in Figura 3.1, il triptofano può essere utilizzato come substrato per sei reazioni differenti, che permettono all'amminoacido di entrare nei diversi rami del suo metabolismo. Abbiamo già avuto modo di osservare precedentemente che le due parti principali di tale sistema metabolico sono i pathway della chinurenina ('Kynurenine Pathway' in figura) e della serotonina ('Serotonin Pathway'), nei quali vengono prodotti diversi metaboliti di interesse. Poiché l'utilizzazione del triptofano è differente in base al tessuto di riferimento, nasce l'importanza di definire dei modelli tissue-specific, che ripecchino le finalità del triptofano nelle diverse parti del corpo. La maggiorparte dell'amminoacido viene metabolizzato nel fegato e una grossa parte di serotonina viene sintetizzata nell'intestino, mentre molti altri prodotti intermedi vengono utilizzati nel cervello.

Un fatto da non dimenticare nella costruzione del modello, che porta a delle complicazioni non indifferenti consiste nel fatto che molti enzimi della rete hanno una selettività piuttosto bassa del substrato e perciò catalizzano reazioni in rami diversi della rete. Ciò comporta che metaboliti che compaiono in parti differenti competono per il sito di attivazione dello stesso enzima. Ad esempio l'enzima IDO usa quattro substrati differenti (Trp, 5hTrp, serotonina e melatonina). Inoltre possiamo osservare che la L-kynurenine (un prodotto intermedio della via delle chinurenine) può essere trasportata attraverso la membrana cellulare utilizzando gli stessi trasportatori del triptofano: questo implica una limitazione nel rate di assorbimento del triptofano, che deve competere con tale prodotto, per attraversare la membrana.

<sup>1</sup><http://www.copasi.org/tiki-index.php?page=downloadnoncommercial>

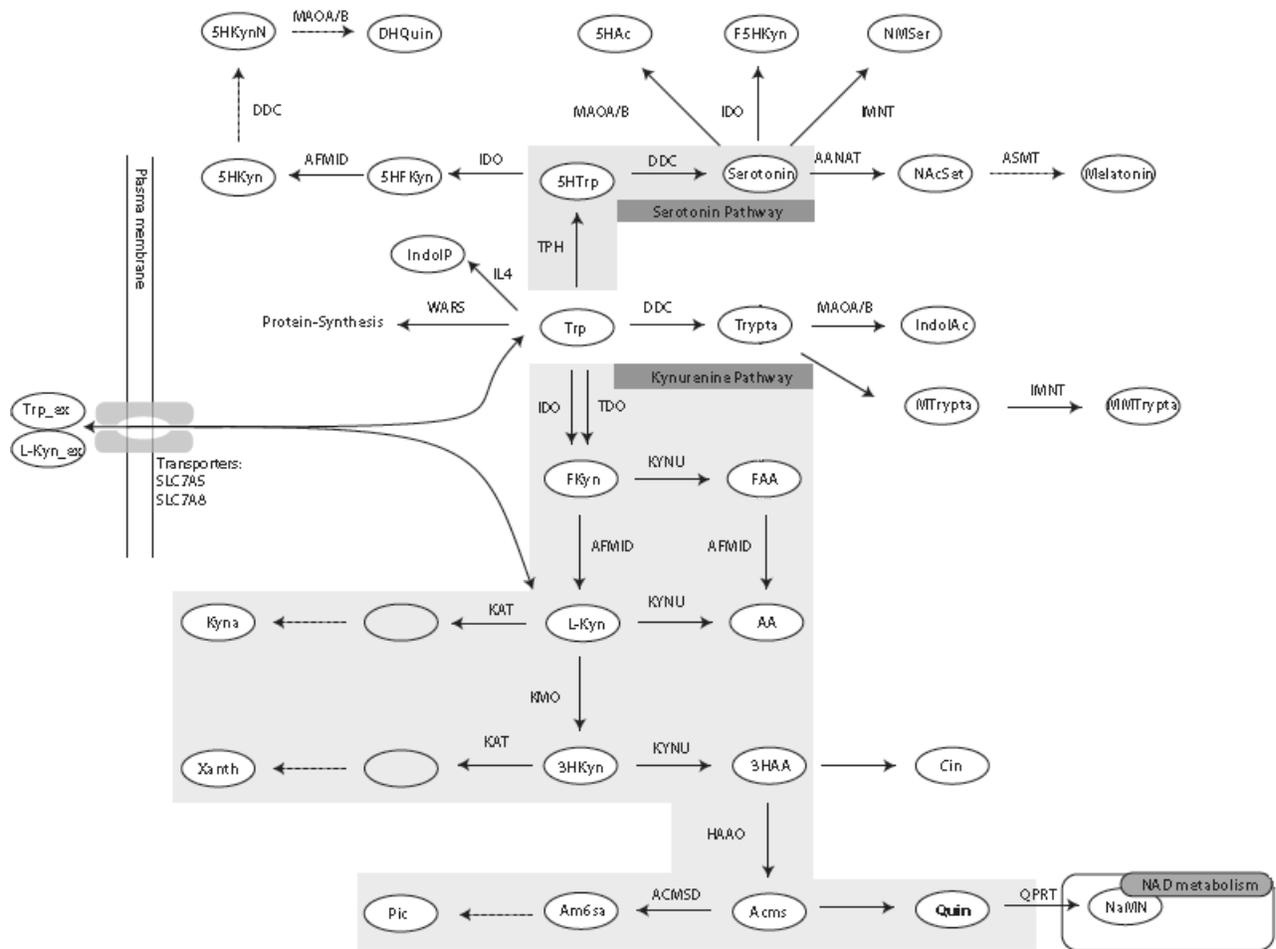


Figure 3.1: *Modello cinetico (compartimentale) del metabolismo del triptofano nell'essere umano. I due pathway principali sono evidenziati dallo sfondo grigio. I metaboliti sono rappresentati dalle ellissi, mentre le direzioni delle reazioni sono rappresentate dalle frecce e dall'enzima regolatore (le frecce prive di enzima corrispondono a reazioni NON-enzimatiche): le frecce tratteggiate corrispondono a reazioni che non sono state modellizzate a causa della mancata conoscenza dei relativi parametri cinetici. Le due linee verticali sulla sinistra rappresentano la membrana cellulare: le due frecce bidirezionali identificano il trasporto attraverso la membrana di triptofano ed L-kynurenine.*

Tramite la Metabolic Control Analysis (MCA), la cui descrizione sarà chiarita nel seguito, gli autori hanno potuto calcolare le distribuzioni del flusso di controllo fra tutti gli enzimi, assumendo che tale flusso di controllo sia comune a tutti gli enzimi nella rete. I coefficienti del flusso di controllo (FCCs) sommano a uno, in modo che i cambiamenti in termini di concentrazione (attività) di un enzima hanno sempre degli effetti sul controllo esercitato dagli altri enzimi del pathway.

Uno degli aspetti fondamentali per la costruzione di un modello cinetico quindi è la definizione dei parametri relativi agli enzimi della rete. Nonostante non siano noti tutti i meccanismi cinetici relativi agli enzimi di interesse, gli autori hanno utilizzato le relazioni 'Michaelis-Mentel kinetics' come buona approssimazione per valutare i rate delle reazioni di interesse. Tali reazioni necessitano di due diversi parametri:

- $K_m$ : costante specifica di half-saturation;
- $V_{max}$ : velocità massima.

La velocità massima  $V_m$  può essere misurata direttamente come l'attività specifica del tessuto, oppure può essere determinata indirettamente tramite:

- $k_{cat}$ : turnover number;
- $E_T$ : concentrazione enzimatica.

Tutti questi parametri cinetici, ad eccezione di  $E_T$  sono stati ricavati dalla letteratura o da appositi database.

Oltre alle informazioni cinetiche, la costruzione di un modello cinetico necessita della conoscenza delle attività specifiche di ciascun enzima nei diversi tessuti: i flussi enzimatici e il loro controllo nei vari pathway variano tra i diversi tessuti, a causa di una differente espressione dei geni ad essi relativi. A tale scopo è necessario introdurre dei fattori di scala per differenziare le diverse attività enzimatiche nei tessuti. Poiché è noto che molti metaboliti intermedi prodotti in tale rete metabolica vengono trasportati nel sangue, le diverse simulazioni utilizzate per costruire il modello hanno verificato che i metaboliti generati in un tessuto specifico possono influenzare il metabolismo del triptofano in altri tessuti. Per tener conto di tale inter-dipendenza tra tessuti, il modello costruito considera il sangue come "metabolita trasportatore".

## 3.2 Metodi utilizzati

Gli autori del paper hanno raccolto le diverse informazioni da database consultabili in rete:

- KEGG (<http://www.genome.jp/kegg/pathway/map/map00380.html>): per l'insieme delle reazioni coinvolte nel metabolismo del triptofano;
- Brenda (<http://www.brenda-enzymes.org/>): per le informazioni inerenti ai parametri cinetici degli enzimi (dove disponibili);
- Uniprot (<http://www.uniprot.org/>): per le affinità tra trasportatori e substrati.

Tutte le informazioni ricavate dai database, sono state rivisitate, per verificare che le condizioni misurate fossero appropriate.

Purtroppo alcune reazioni non si sono potute includere nel modello, a causa di mancanza di dati. Fortunatamente tali reazioni si trovano nelle parti finali delle diverse vie del metabolismo e quindi non hanno effetti nella parte rimanente del modello. Purtroppo però, per quanto ci riguarda, manca la descrizione della trasformazione della serotonina in melatonina.

Per quanto riguarda la modellizzazione delle reazioni, per le reazioni enzimatiche e di trasporto sono state utilizzate le relazioni di 'Michaelis-Mentel kinetics' (o delle versioni leggermente modificate), mentre le reazioni non enzimatiche sono state modellate tramite la legge di azione di massa.

Per poter sopperire ad alcune informazioni mancanti, gli autori hanno messo in atto alcune semplificazioni necessarie:

- Eccetto che per le reazioni di trasporto, tutte le altre reazioni sono modellate come IRREVERSIBILI, per due ragioni principali:

1. le reazioni che coinvolgono ossigenazione, acetilazione, formazione o rottura di anelli sono irreversibili per la loro natura chimica;

2. le reazioni che potenzialmente potrebbero essere reversibili, sono seguite da veloci reazioni non enzimatiche, che guidano la reazione precedente nel verso 'in avanti'.
- Nel modello non viene inclusa nessuna azione di inibizione da parte di alcun metabolita, nonostante si sia osservato (in vitro) che metaboliti presenti nella rete, quali Pic, Quin, AA e Kyna, possono inibire diverse azioni enzimatiche. In ogni caso la metà delle concentrazioni inibitorie sono 1000 volte superiori rispetto a quelle fisiologiche riportate, perciò si è preferito non tenerne conto.
  - Tutti i co-substrati presenti nel modello (inclusi PRPP, NADPH, acetyl-CoA e ossigeno) sono stati settati come metaboliti esterni. le loro concentrazioni sono state fissate a 1  $mM$  (ad eccezione di NADPH: 0.03  $mM$ ) per non limitare il flusso. Ciò significa anche che le leggi che regolano i rate cinetici possono essere modellati come reazioni monomolecolari.
  - La concentrazione di triptofano esterno alla membrana cellulare è stato settato a 5  $\mu M$ .

### 3.2.1 Modellizzazione dell'attività specifica nei diversi tessuti

Per ottenere i modelli specifici per il tessuto di interesse, gli autori hanno utilizzato due diversi approcci:

1. il primo è basato sulla misura dell'attività enzimatica nel tessuto specifico;
2. il secondo è basato sulla combinazione della misura dell'attività enzimatica 'purificata' e i dati di espressione genica specifici del tessuto di interesse.

Anche se per completezza qui di seguito riporteremo le equazioni inerenti ad entrambi gli approcci, va sottolineato il fatto che gli autori del paper, in seguito ai primi risultati ottenuti utilizzando il primo approccio, hanno poi preferito seguire l'approccio numero due che sembrava dare dei risultati più simili a quanto osservato sperimentalmente.

Seguendo il primo approccio, che porta alla conoscenza del parametro  $V_{max}$ , le attività nel tessuto specifico possono essere calcolate tramite l'equazione di Michaelis-Mentel:

$$(1) \quad v = \frac{V_{max} \times S}{K_m + S} ;$$

dove:

- $v$  è il rate della reazione;
- $K_m$  è la costante di half-saturation;
- $S$  è la concentrazione del substrato.

Utilizzando il secondo approccio, il rate  $v$ , può essere calcolato tramite una delle due equazioni seguenti. Le due equazioni sono equivalenti, in quanto la concentrazione enzimatica  $E_T$  può essere calcolata a partire dai dati di espressione genica come  $E_T = F \times mRNA$ :

$$(2) \quad v = \frac{E_T \times k_{cat} \times S}{K_m + S}$$

$$(3) \quad v = \frac{F \times mRNA \times k_{cat} \times S}{K_m + S}$$

dove:

- $E_T$  è la concentrazione totale dell'enzima. Poichè spesso la concentrazione enzimatica di un tessuto specifico non è nota, essa viene calcolata come il prodotto tra la misura del segnale di espressione genica  $mRNA$  e un fattore  $F$  (in  $mM$ ), utilizzato per convertire segnali ottenuti da microarray in concentrazioni enzimatiche. In caso  $F$  non fosse noto è stato posto uguale ad 1  $mM$ : ciò ha delle ripercussioni inerenti ai risultati ottenuti con le simulazioni relativi ai flussi delle reazioni non enzimatiche (i risultati restituiscono dei flussi arbitrari);
- $k_{cat}$  è il turnover number dell'enzima.

Quando gli enzimi catalizzano più di una reazione, i rate di flusso sono stati calcolati utilizzando l'equazione di Michaelis-Mentel relativa all'inibizione competitiva di reazioni irreversibili: a seconda di quale approccio si voglia utilizzare, tale reazione è espressa in uno dei due modi seguenti:

$$(4) \quad v = \frac{V_{max} \times S}{K_m + S + \frac{K_m \times I}{K_i}}$$

$$(5) \quad v = \frac{F \times mRN A \times k_{cat} \times S}{K_m + S + \frac{K_m \times I}{K_i}};$$

dove:

- $I$  è la concentrazione del substrato in concorrenza;
- $K_i$  è la costante di Michaelis-Mentel del substrato in esame.

Tale equazione può essere estesa a qualsiasi numero finito di substrati in competizione.

Per quanto riguarda le reazioni di trasporto di triptofano e L-chinurenina, gli autori hanno determinato il rate  $v$  utilizzando la legge inerente a substrati in competizione basata sulle assunzioni di stato stazionario.

Le informazioni inerenti ai data set delle espressioni geniche sono state reperite da Array Express (<http://www.ebi.ac.uk/arrayexpress/>) o da Gene Expression Omnibus (GEO) (<http://www.ncbi.nlm.nih.gov/geo/>).

Il file sbml del modello del metabolismo del triptofano nel fegato è scaricabile dal database BioModels (<http://www.ebi.ac.uk/biomodels-main/>).

### 3.3 Analisi effettuate tramite il Modello Cinetico

Il software Copasi, che verrà descritto dettagliatamente nel seguito, permette di eseguire molteplici tipologie di analisi. In particolare la nostra attenzione sarà rivolta a due tasks particolari (anch'essi saranno descritti nel dettaglio nella sezione relativa al software Copasi; qui ci limiteremo a spiegarne in breve l'utilità per la nostra analisi):

1. analisi della distribuzione dei flussi relativi;
2. analisi di sensitività.

Nell'analisi della distribuzione dei flussi osserveremo quale percentuale di tutto il triptofano entrante nella cellula viene metabolizzato in ciascuna reazione del modello (ci aspettiamo che tali flussi relativi siano diversi nei differenti tessuti analizzati).

L'analisi di sensitività ci permette di osservare come varia una determinata variabile in relazione alla variazione di un parametro di interesse:

$$S_P^A = \frac{\partial A}{\partial p} \cdot \frac{p}{A};$$

In particolare ci interessa osservare in che modo l'espressione genica influenza:

- le concentrazioni:

$$S_{gET}^C = \frac{\partial C}{\partial gET} \cdot \frac{gET}{C};$$

- i flussi:

$$S_{SET}^F = \frac{\partial F}{\partial SET} \cdot \frac{SET}{F};$$

↗ FLUSSO di una REAZIONE  
↘ ESPRESSIONE GENICA (in t=0)

Come abbiamo avuto modo di constatare nella descrizione delle equazioni utilizzate per costruire il modello, esiste un legame tra la concentrazione enzimatica  $E_T$  e l'espressione genica  $mRNA$ :

$$E_T = F \times mRNA.$$

Perciò, se consideriamo il fattore  $F = costante$ , nel caso in cui non siamo a conoscenza della concentrazione enzimatica, possiamo utilizzare l'espressione genica come valore per indicare tale concentrazione e viceversa. A seconda della disponibilità dei dati quindi, nell'analisi di sensitività verrà fatta variare la concentrazione enzimatica  $E_T$  oppure l'espressione genica  $mRNA$ .

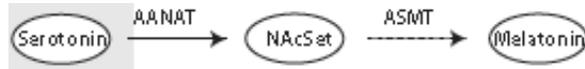
### 3.3.1 Reazioni di particolare interesse nell'analisi

Per raggiungere lo scopo di questa tesi, durante l'analisi concentreremo la nostra attenzione su alcune reazioni in particolare:

- **DDC\_5HTrp**: produzione di serotonina (Serotonin)



- **AANAT\_Serotonin**: produzione di melatonina (Melatonin);



- **Quinolinic Formation (spontanea)**: produzione di acido quinolinico (Quin);



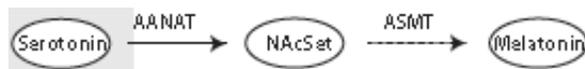
- **KAT\_3HKyn**: produzione di acido xanturenico (Xanth).



Per quanto riguarda la serotonina e la melatonina, siamo maggiormente interessati alla produzione di melatonina a partire dalla serotonina, più che alla produzione della serotonina stessa. Perciò nel proseguimento delle analisi, porremo attenzione solamente alla reazione AANAT\_Serotonin e non a DDC\_5HTrp.

Nell'analisi di sensitività, in cui osserveremo come variano le concentrazioni e i flussi di reazione in seguito alla variazione dell'espressione genica (o analogamente alla concentrazione enzimatica), purtroppo il modello non ci permette di osservare direttamente le grandezze di interesse, a causa della mancata modellizzazione di parti del modello. Tale difficoltà viene riscontrata in particolare in due casi, in cui si procederà come descritto qui di seguito:

- produzione di melatonina (non modellizzata):



metabolita osservato: Serotonin;

flusso osservato: AANAT\_Serotonin.

- produzione di acido xanturenico (non modellizzato):



metabolita osservato: 3HKyn;

flusso osservato: KAT\_3HKyn, regolato dai geni KAT1, KAT2 e KAT3.

### 3.3.2 Dati utilizzati

Come è stato più volte chiarito, le analisi effettuate con l'utilizzo del modello cinetico necessitano della conoscenza di diversi parametri relativi ai soggetti e ai tessuti di interesse. Il paper contenente la descrizione del modello, contiene tutte le informazioni inerenti ai tessuti del fegato e del cervello nell'essere umano. È stato perciò possibile studiare questi due tessuti differenti.

Inoltre sono stati utilizzati dei dati reali del muscolo scheletrico relativi a soggetti sani, diabetici di tipo due e intolleranti al diabete: tali dati sono stati reperiti dal database GEO<sup>2</sup> (<http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE18732>).

In conclusione quindi le analisi sono state effettuate in diversi tessuti e situazioni:

1. fegato (soggetti sani);
2. cervello (soggetti sani);
3. muscolo scheletrico (soggetti sani e diabetici).

Al fine di non fraintendere i risultati che verranno esposti, è necessario sottolineare che i dati reali utilizzati (relativi al muscolo scheletrico) sono misure di espressione di RNA. Per raggiungere i nostri obiettivi di studio, sarebbe stato più utile avere a disposizione dei dati relativi alle concentrazioni nei soggetti sani e diabetici. Queste misure però non sono disponibili, perciò abbiamo utilizzato tali informazioni geniche per risalire ai parametri utili nella costruzione del modello.

Va evidenziato però che, nel caso del muscolo scheletrico, non è stato sempre possibile decidere in modo univoco quale diversa probe associare ad ogni particolare gene: più probe infatti possono essere relative allo stesso gene, in sezioni diverse del DNA. Perciò, sono state analizzate diverse situazioni possibili, derivanti da scelte diverse di probe considerate. Nei grafici che verranno utilizzati per esporre i risultati, tali combinazioni diverse saranno rappresentate da diverse lettere maiuscole.

Inoltre, in base al tipo di analisi effettuata, si è scelto di utilizzare la media o la mediana come statistica.

## 3.4 Risultati

### 3.4.1 Analisi dei flussi relativi

#### 3.4.1.1 Flussi relativi nei diversi tessuti (fegato e cervello).

Grazie alla costruzione di modelli specifici per i diversi tessuti, possiamo osservare subito un primo risultato inerente ai tessuti del fegato e del cervello descritti nel paper: il triptofano viene metabolizzato in percentuali diverse attraverso le reazioni nei due modelli relativi ai due tessuti. I risultati riportati fanno riferimento ad un'analisi in stato stazionario a partire da condizioni iniziali che rispecchiano le assunzioni riportate precedentemente e che ripetiamo qui di seguito:

- Tutti i co-substrati presenti nel modello (inclusi PRPP, NADPH, acetyl-CoA e ossigeno) sono stati settati con concentrazioni fissate a  $1\text{ mM}$  (ad eccezione di NADPH:  $0.03\text{ mM}$ ) per non limitare il flusso. Ciò significa anche che le leggi che regolano i rate cinetici possono essere modellati come reazioni monomolecolari.
- La concentrazione di triptofano esterno alla membrana cellulare è stato settato a  $5\ \mu\text{M}$ .

---

<sup>2</sup>Series GSE18732; PMID: 20353613.

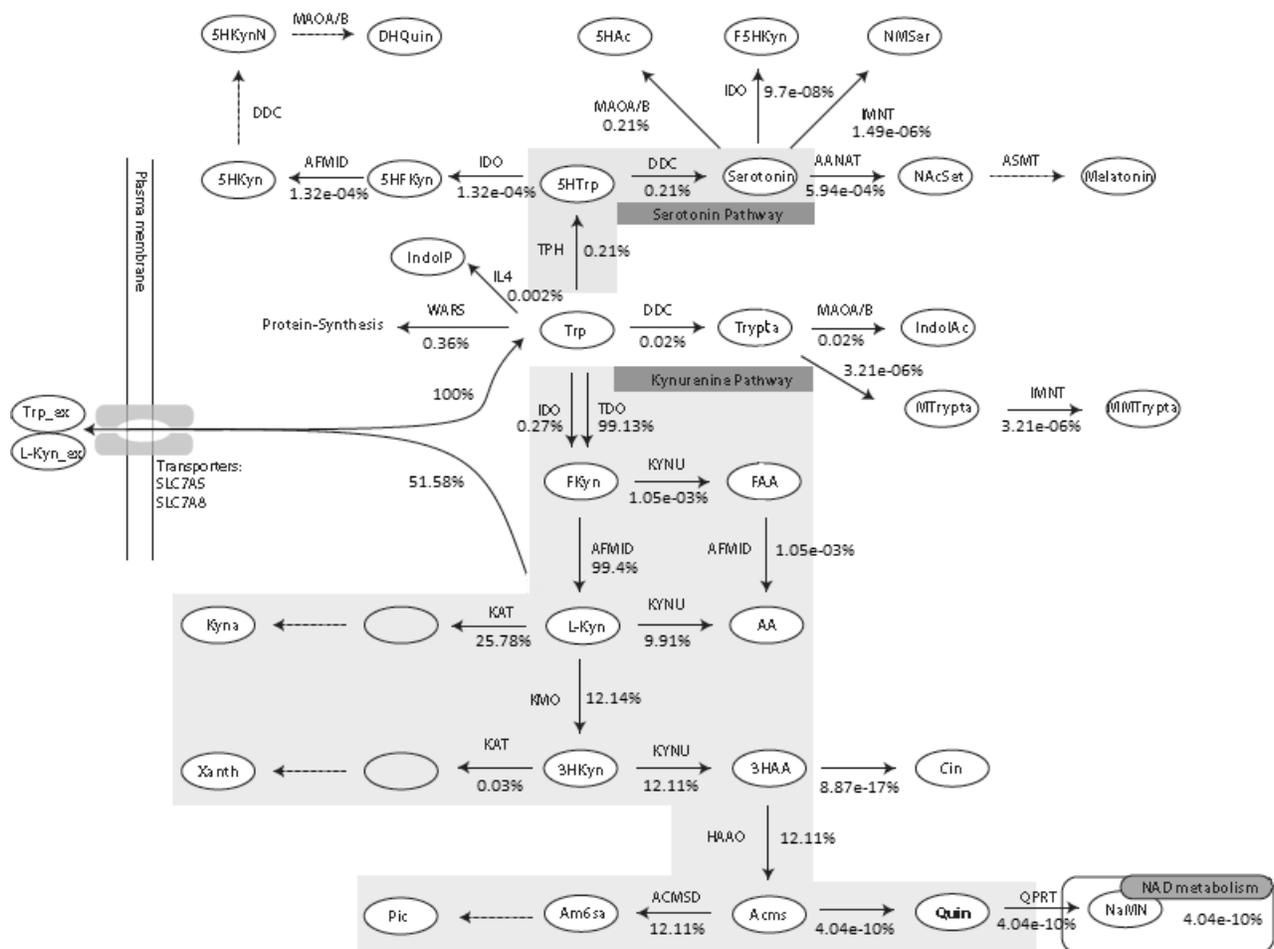


Figure 3.2: Modello del metabolismo del triptofano nel *fegato*: percentuali di triptofano metabolizzato in ciascuna reazione della rete.

Per quanto riguarda tutte le altre condizioni iniziali, fare riferimento all'Appendice C, in cui sono si possono visualizzare i valori settati per ciascun metabolita presente nella rete.

In Figura 3.2 e 3.3 possiamo osservare quale percentuale del triptofano entrante nella rete (ossia nella cellula del tessuto in esame) venga utilizzato in ciascun flusso di reazione nel fegato (Figura 3.2) e nel cervello (Figura 3.3).

Dalle Figure è già possibile osservare degli aspetti interessanti per quanto riguarda il nostro scopo di analisi, ricordando che il nostro obiettivo primario è quello di capire in che misura il metabolismo del triptofano influenza la produzione di serotonina, melatonina e acidi quinolinico e xanturenico. Come osservato precedentemente, purtroppo il modello cinetico analizzato non contiene la modellizzazione della reazione che porta alla produzione di melatonina, regolata dal gene ASMT. In ogni caso, per le osservazioni future, possiamo fare riferimento alla reazione subito precedente regolata dal gene AANAT. Lo stesso problema si verifica nel caso dell'acido xanturenico, per il quale faremo riferimento al flusso uscente dal compartimento di 3HKyn, regolato dai geni KAT1, KAT2 e KAT3.

Possiamo quindi osservare una differenza tra i tessuti del fegato e del cervello inerenti alla percentuale di triptofano entrante nella cellula metabolizzato dalle reazioni di nostro interesse, ovvero:

- **DDC\_5HTrp**: produzione di serotonina (Serotonin)
- **AANAT\_Serotonin**: produzione di melatonina (Melatonin);
- **Quinolinic Formation (spontanea)**: produzione di acido quinolinico (Quin);
- **KAT\_3HKyn**: produzione di acido xanturenico (Xanth).

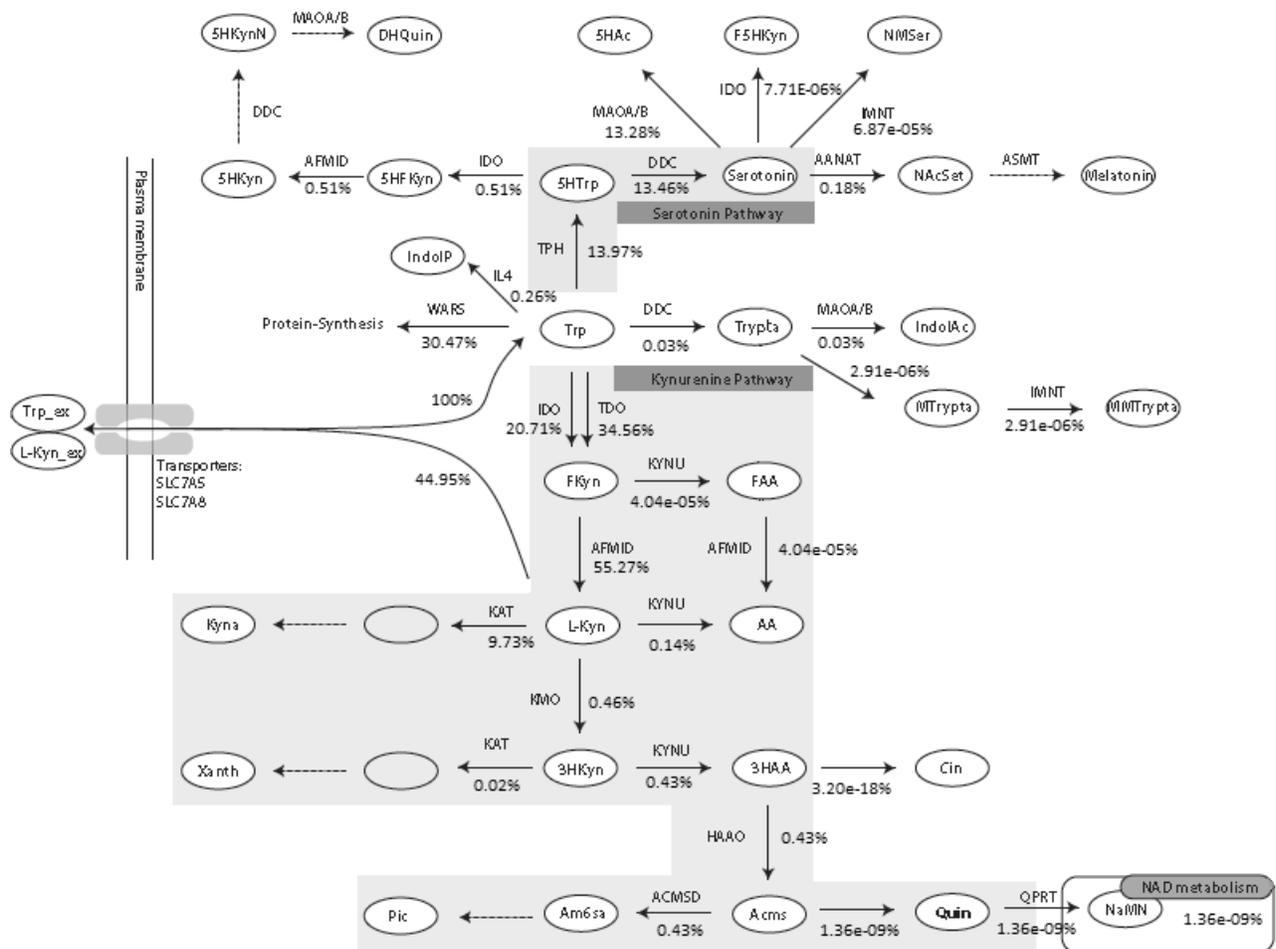


Figure 3.3: *Modello del metabolismo del triptofano nel cervello: percentuali di triptofano metabolizzato in ciascuna reazione della rete.*

In Tabella possiamo osservare nel dettaglio tali differenze.

REAZIONE	FEGATO	CERVELLO
DDC_5HTrp (serotonina)	0.21%	15.46%
AANAT_Serotonin (melatonina)	$5.94 \cdot 10^{-4}\%$	0.18%
Quinolinic formation (quinolinato)	$4.04 \cdot 10^{-10}\%$	$1.36 \cdot 10^{-9}\%$
KAT_3HKYN (acido xanturenico)	0.03%	0.02%

Anche se sappiamo che la maggiorparte della serotonina viene secreta nell'intestino, dalla Tabella risulta chiaro che il cervello, rispetto al fegato, ha una rilevanza maggiore per quanto riguarda la produzione di serotonina e melatonina. Seppur in percentuali molto ridotte, anche l'acido quinolinico è prodotto maggiormente nel cervello. Il fegato invece è predominante rispetto al cervello per quanto riguarda l'acido xanturenico.

### 3.4.1.2 Flussi relativi nel muscolo scheletrico: confronto tra soggetti sani e diabetici di tipo due

Per quanto riguarda l'analisi relativa ai flussi di reazione nel muscolo scheletrico, è possibile effettuare un confronto tra la distribuzione dei flussi nei soggetti sani e nei soggetti diabetici. (Tutte le analisi nel muscolo sono state effettuate nelle stesse condizioni descritte nel caso del fegato e del cervello). In questo caso è stata utilizzata la statistica della mediana, ritenuta più robusta.

Come osservato in precedenza, sono state studiate diverse situazioni possibili, caratterizzate da diverse scelte di dati (probe) utilizzati. Nelle Figure seguenti si possono osservare dei confronti tra sani e diabetici relativi alle tre reazioni di particolare interesse.

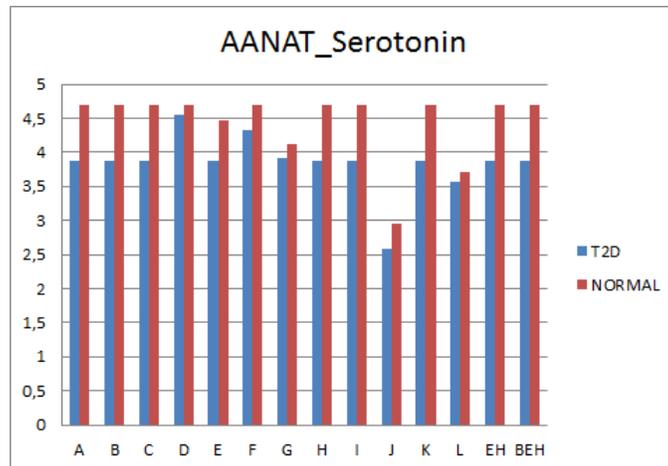


Figure 3.4: Percentuale di triptofano metabolizzato nella reazione AANAT\_Serotonin (produzione di melatonina): confronto tra soggetti sani (NORMAL) e diabetici (T2D) in diverse situazioni (A, B, ...) possibili nel **muscolo scheletrico**.

In Figura 3.4 in particolare appare evidente come in tutte le diverse situazioni osservate, nei pazienti diabetici si osserva una diminuzione del flusso relativo legato alla produzione di melatonina rispetto ai soggetti sani. Al contrario, in Figura 3.5 si nota un leggero aumento di flusso attraverso la reazione che produce acido quinolinico nei soggetti diabetici: tale risultato non è presente in tutte le situazioni analizzate, poiché nelle situazioni D, E ed EH si osserva un comportamento opposto. Per quanto riguarda il flusso relativo alla produzione di acido xanturenico, in Figura 3.6 possiamo osservare anche in questo caso un aumento nei pazienti diabetici rispetto ai sani nella maggiorparte dei casi.

Queste osservazioni sembrano confermare le ipotesi, secondo le quali, nei pazienti diabetici si osserva:

- una sottoproduzione di melatonina,
- una sovrapproduzione di acidi quinolinico e xanturenico

rispetto ai soggetti sani.

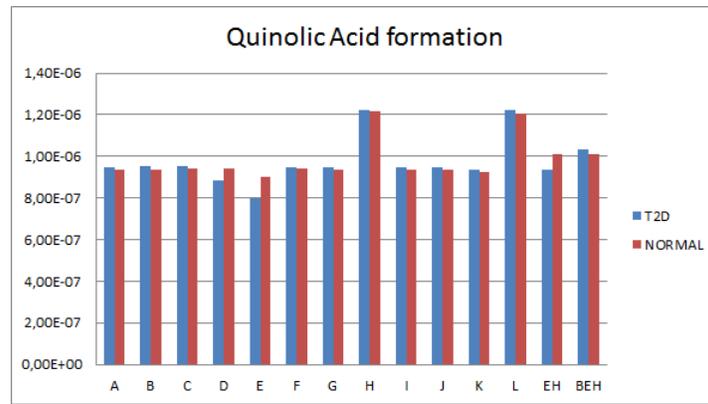


Figure 3.5: Percentuale di triptofano metabolizzato nella reazione spontanea *QuinolinicAcid formation* (produzione di acido quinolinico): confronto tra soggetti sani (*NORMAL*) e diabetici (*T2D*) in diverse situazioni (*A, B, ...*) possibili nel *muscolo scheletrico*.

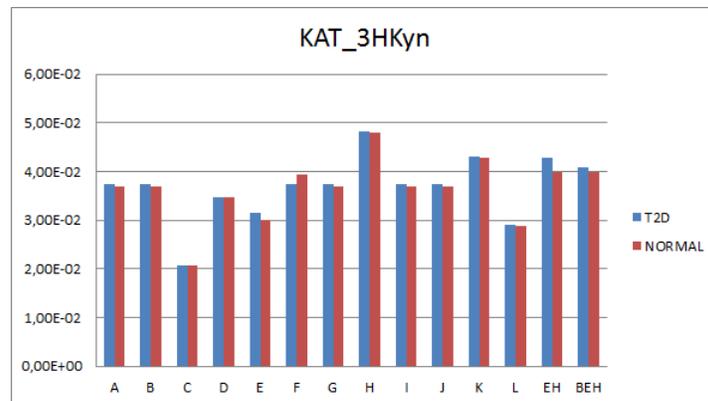


Figure 3.6: Percentuale di triptofano metabolizzato nella reazione *KAT\_3HKyn* (produzione di acido xanturenico): confronto tra soggetti sani (*NORMAL*) e diabetici (*T2D*) in diverse situazioni (*A, B, ...*) possibili nel *muscolo scheletrico*.

### 3.4.2 Analisi di sensitività

#### 3.4.2.1 Analisi di sensitività delle concentrazioni rispetto a variazioni di espressione genica o concentrazione enzimatica

In questa analisi è stato possibile osservare come reagiscano le diverse concentrazioni dei metaboliti di interesse in seguito a variazioni nella disponibilità dei geni che regolano le diverse reazioni del modello.

Per quanto riguarda i tessuti di fegato e cervello, il parametro utilizzato è la concentrazione enzimatica  $E_T$ , mentre nel caso del muscolo scheletrico è stata considerata l'espressione genica *mRNA*. Inoltre in questo caso, la statistica ritenuta più opportuna (sia nel caso di pazienti sani, sia diabetici) è stata la media.

Poiché ogni enzima è il risultato di una traduzione di RNA, tramite questa analisi è possibile individuare quali geni siano particolarmente influenti per le concentrazioni di interesse.

In Figura 3.7, 3.8, 3.9 e 3.10 possiamo osservare rispettivamente i valori ottenuti come sensitività delle concentrazioni di serotonina, acido quinolinico ed acido xanturenico nel fegato, nel cervello, nel muscolo scheletrico (pazienti sani in Figura 3.9 e malati in Figura 3.10). Nell'asse delle ascisse si leggono i nomi dei geni caratteristici degli enzimi, le cui concentrazioni sono state variate (gli eventuali geni non rappresentati hanno portato ad un valore di sensitività nulla per tutte le concentrazioni osservate).

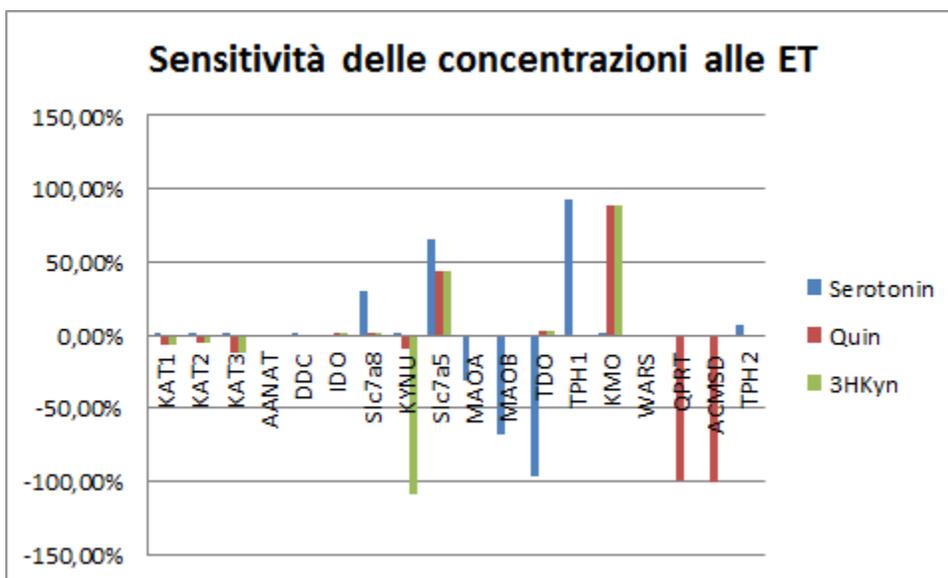


Figure 3.7: *Analisi di sensitività delle concentrazioni di serotonina (blu), acido quinolinico (rosso) ed acido xanturenico (verde) rispetto a variazioni di concentrazioni enzimatiche nel **fegato**. Nell'asse delle ascisse si leggono i nomi dei geni caratteristici degli enzimi, le cui concentrazioni sono state variate.*

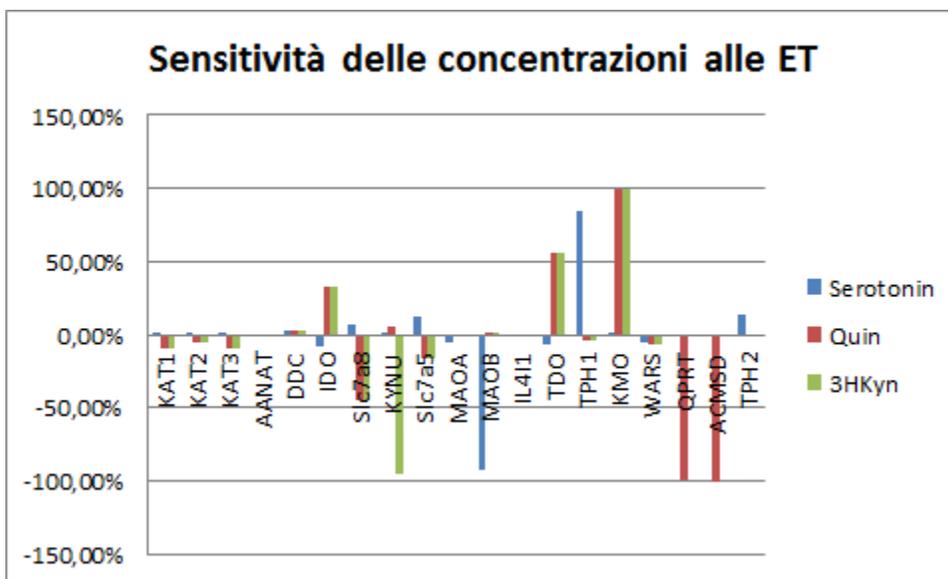


Figure 3.8: *Analisi di sensitività delle concentrazioni di serotonina (blu), acido quinolinico (rosso) ed acido xanturenico (verde) rispetto a variazioni di concentrazioni enzimatiche nel **cervello**. Nell'asse delle ascisse si leggono i nomi dei geni caratteristici degli enzimi, le cui concentrazioni sono state variate.*

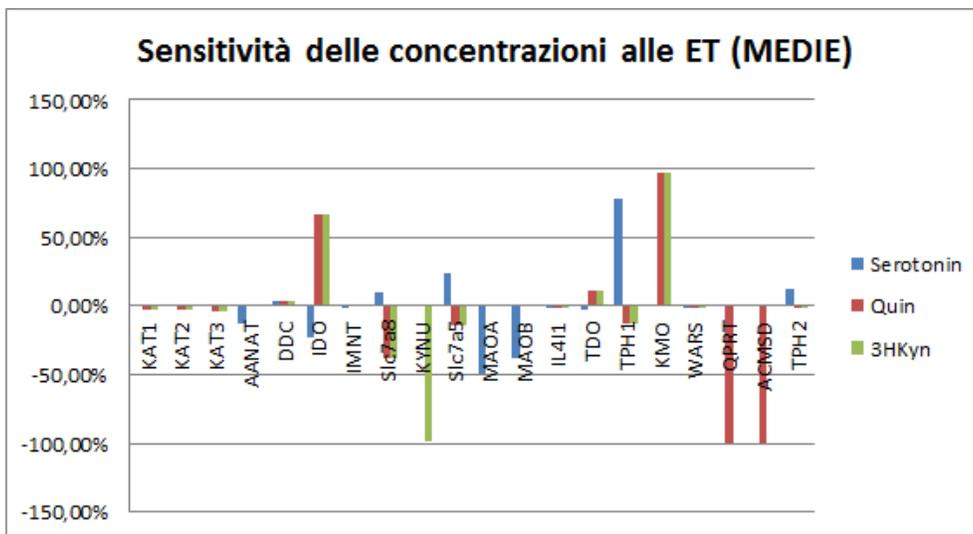


Figure 3.9: *Analisi di sensitività delle concentrazioni di serotonina (blu), acido quinolinico (rosso) ed acido xanturenico (verde) rispetto a variazioni di concentrazioni enzimatiche nel muscolo (paziente sano). Nell'asse delle ascisse si leggono i nomi dei geni caratteristici degli enzimi, le cui concentrazioni sono state variate.*

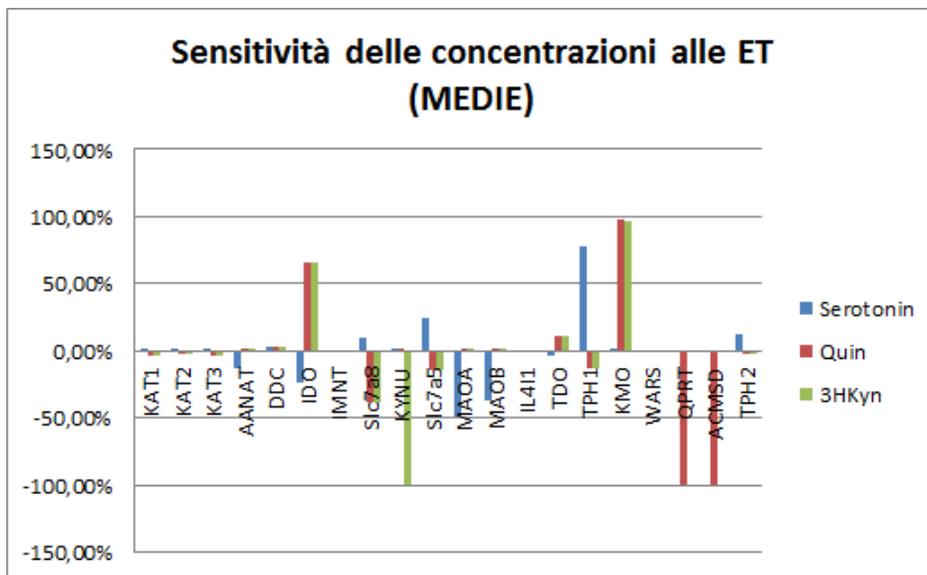


Figure 3.10: *Analisi di sensitività delle concentrazioni di serotonina (blu), acido quinolinico (rosso) ed acido xanturenico (verde) rispetto a variazioni di concentrazioni enzimatiche nel muscolo (paziente diabetico). Nell'asse delle ascisse si leggono i nomi dei geni caratteristici degli enzimi, le cui concentrazioni sono state variate.*

Per effettuare una valutazione più completa dell'analisi eseguita, è interessante visualizzare un confronto delle diverse sensitività relative a ciascuna concentrazione nei diversi tessuti.

In Figura 3.11 possiamo osservare quali geni siano più influenti nella concentrazione della serotonina nei diversi tessuti.

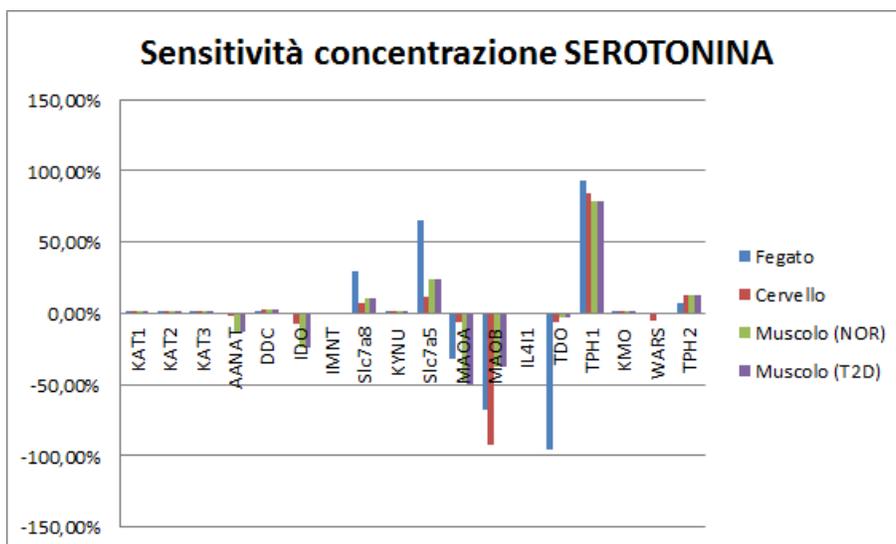


Figure 3.11: Confronto dei valori di sensitività della concentrazione di serotonina rispetto a variazioni dell'espressione genica nei diversi tessuti e nei soggetti sani e diabetici.

In particolare si può notare come i geni AANAT e TPH1 abbiano un'influenza positiva in tutti i tessuti (un aumento della disponibilità di tale gene produce un aumento di serotonina in tutti i tessuti), anche se questo si verifica maggiormente nel fegato e nel cervello. È molto interessante osservare come nel caso di altri geni si verifichino invece delle nette differenze tra tessuti diversi: in particolare evidenziamo:

- il gene TDO2 produce una forte sensitività nel fegato (in verso negativo), mentre è quasi ininfluenza negli altri tessuti;
- i geni relativi alle reazioni di trasporto attraverso la membrana cellulare (Slc7a8 e Slc7a5) sono particolarmente influenti nel fegato, in misura minore nel muscolo e in misura ancora più piccola nel cervello;
- solamente nel caso del gene Slc7a8 si nota una differenza sensibile tra sensitività in pazienti sani e malati nel muscolo.

In Figura 3.12 è possibile osservare un confronto delle sensitività della concentrazione di acido quinolinico.

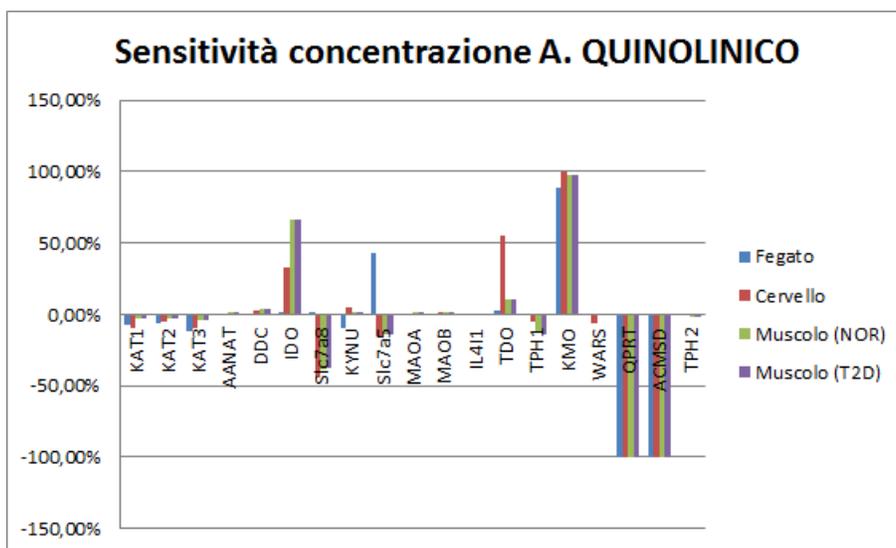


Figure 3.12: Confronto dei valori di sensitività della concentrazione di acido quinolinico rispetto a variazioni dell'espressione genica nei diversi tessuti e nei soggetti sani e diabetici.

In questo caso si notano diverse situazioni interessanti:

- il gene IDO, praticamente ininfluenza nel fegato, è invece molto legato alla concentrazione dell'acido (in senso positivo) nel cervello e, soprattutto, nel muscolo;
- l'acido quinolinico è particolarmente sensibile ai geni KMO (in senso positivo) e QPRT e ACMSD (in senso negativo) in tutti i tessuti osservati;
- il gene TDO2 è particolarmente influente nel cervello, ma lo è in misura molto minore negli altri tessuti;
- un aumento del gene Slc7a8 fa diminuire la concentrazione di quinolinato in tutti i tessuti, mentre il gene Slc7a5 ha un effetto negativo in tutti eccetto che nel fegato, nel quale un suo aumento produce un aumento dell'acido.

Per quanto riguarda la concentrazione di 3HKyn (legata all'acido xanturenico), osserviamo la Figura 3.13, in cui possiamo osservare alcuni aspetti simili al caso dell'acido quinolinico e altri differenti:

- anche in questo caso un aumento del gene Slc7a8 fa diminuire la concentrazione di acido xanturenico in tutti i tessuti, mentre il gene Slc7a5 ha un effetto negativo in tutti eccetto che nel fegato, dove suo aumento produce un aumento dell'acido.
- il gene KMO influenza in senso positivo anche la concentrazione di xanturenico, che invece non è affatto sensibile a variazioni di QPRT e ACMSD;
- ancora una volta il gene IDO è praticamente ininfluenza nel fegato ed è invece molto legato alla concentrazione dell'acido (in senso positivo) nel cervello e, soprattutto, nel muscolo. In questo caso però si osserva una lieve differenza nella sensibilità tra soggetto sano e malato nel muscolo;
- come nel caso del quinolinato, il gene TDO2 è particolarmente influente nel cervello, ma lo è in misura molto minore negli altri tessuti.

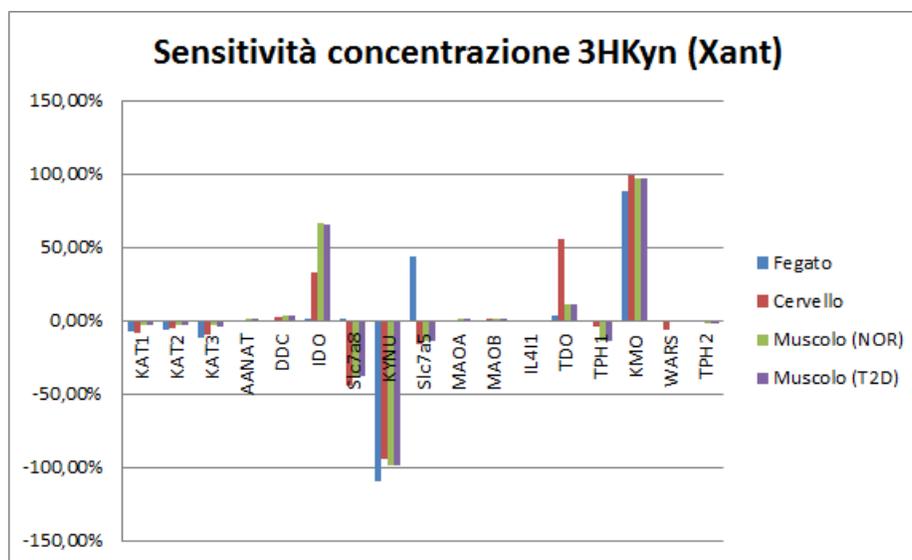


Figure 3.13: Confronto dei valori di sensibilità della concentrazione di acido xanturenico rispetto a variazioni dell'espressione genica nei diversi tessuti e nei soggetti sani e diabetici.

### 3.4.2.2 Analisi di sensibilità dei flussi di reazione rispetto a variazioni di espressione genica o concentrazione enzimatica

In questa seconda analisi di sensibilità abbiamo osservato invece in che modo la variazione genetica può far variare i flussi delle reazioni di interesse, al fine, ancora una volta, di capire quali geni siano più o meno caratterizzanti il nostro studio.

Anche in questo caso, osserviamo innanzitutto le sensibilità nei diversi tessuti, per poi poter eseguire un confronto osservando le singole reazioni di interesse.

In Figura 3.14 e 3.15 possiamo visualizzare le sensibilità dei flussi a variazioni di concentrazioni enzimatiche rispettivamente nel fegato e nel cervello, mentre in Figura 3.16 e 3.17 osserviamo le sensibilità dei flussi alla variazione dell'espressione genica nel muscolo scheletrico di soggetti sani e diabetici.

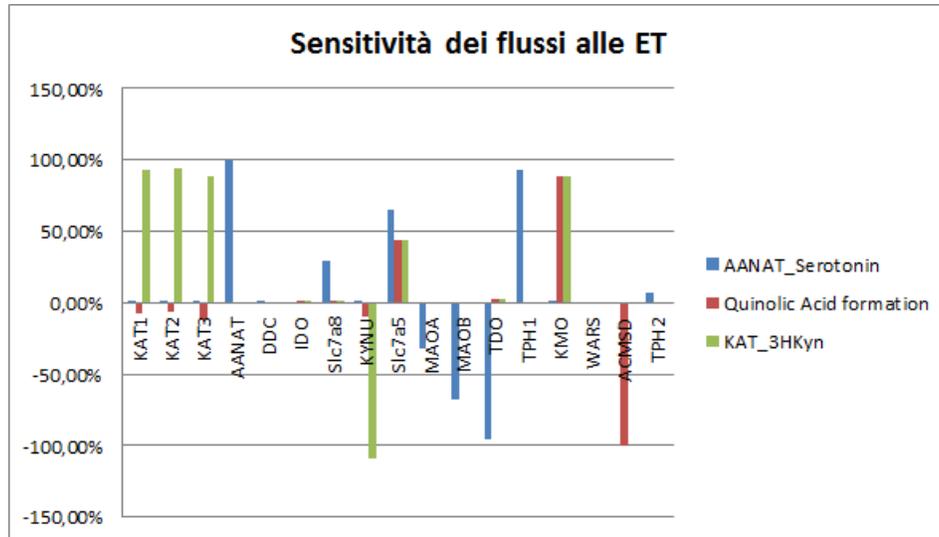


Figure 3.14: Analisi di sensibilità dei flussi di reazione relativi alla produzione di melatonina (blu), acido quinolinico (rosso) e acido xanturenico (verde) rispetto a variazioni di concentrazioni enzimatiche nel **fegato**. Nell'asse delle ascisse si leggono i nomi dei geni caratteristici degli enzimi, le cui concentrazioni sono state variate.

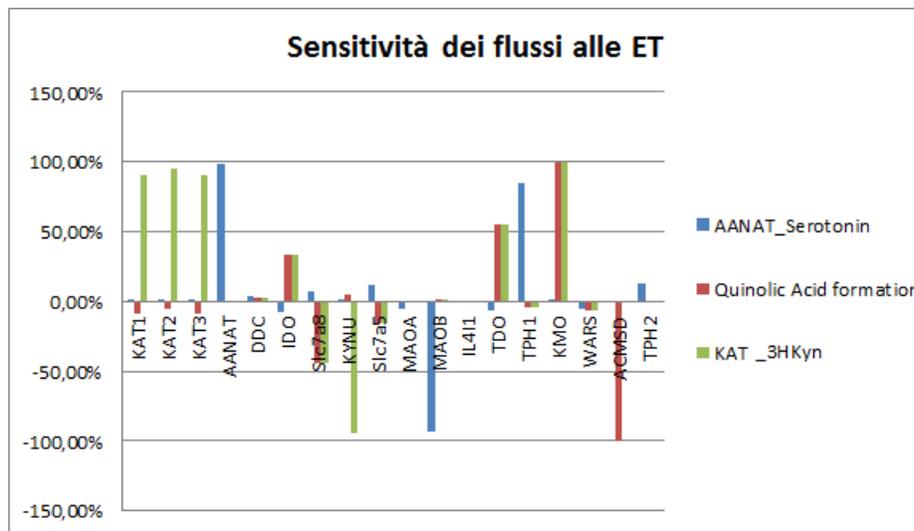


Figure 3.15: Analisi di sensibilità dei flussi di reazione relativi alla produzione di melatonina (blu), acido quinolinico (rosso) e acido xanturenico (verde) rispetto a variazioni di concentrazioni enzimatiche nel **cervello**. Nell'asse delle ascisse si leggono i nomi dei geni caratteristici degli enzimi, le cui concentrazioni sono state variate.

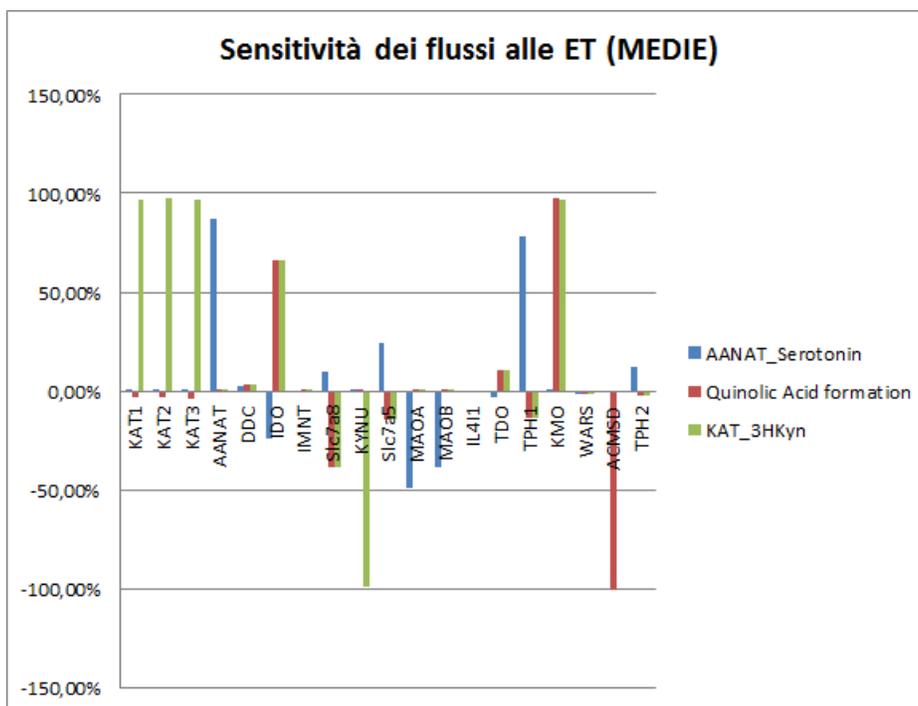


Figure 3.16: *Analisi di sensitività dei flussi di reazione relativi alla produzione di melatonina (blu), acido quinolinico (rosso) e acido xanturenico (verde) rispetto a variazioni di concentrazioni enzimatiche nel muscolo (soggetti sani). Nell'asse delle ascisse si leggono i nomi dei geni caratteristici degli enzimi, le cui concentrazioni sono state variate.*

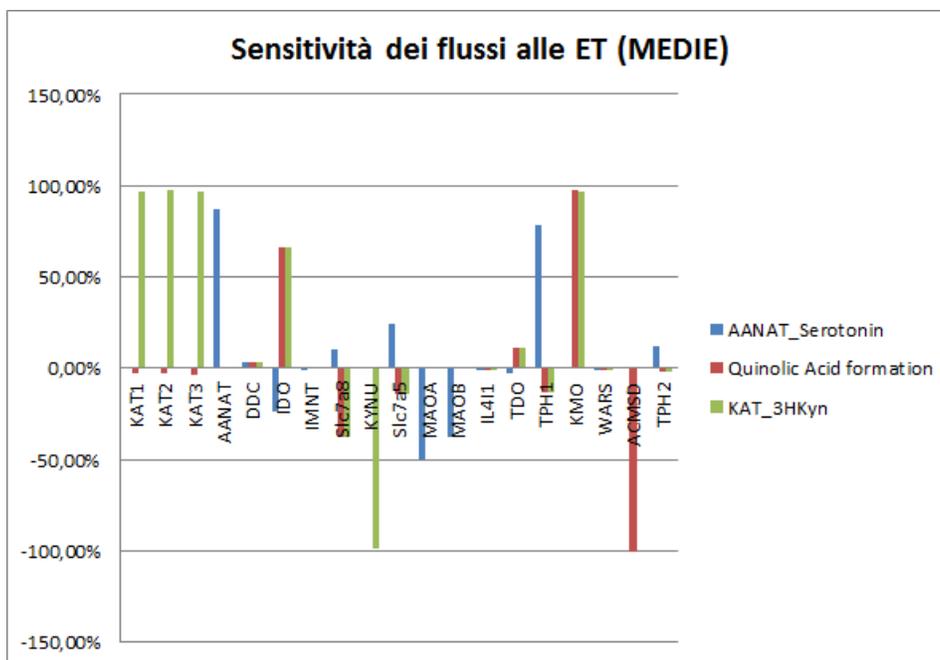


Figure 3.17: *Analisi di sensitività dei flussi di reazione relativi alla produzione di melatonina (blu), acido quinolinico (rosso) e acido xanturenico (verde) rispetto a variazioni di concentrazioni enzimatiche nel muscolo (soggetti malati). Nell'asse delle ascisse si leggono i nomi dei geni caratteristici degli enzimi, le cui concentrazioni sono state variate.*

A questo punto possiamo quindi effettuare un confronto delle sensitività dei flussi delle specifiche reazioni

nei diversi tessuti. In Figura 3.18 è possibile visualizzare tale confronto in relazione alla reazione AANAT\_Serotonin, che rappresenta il flusso che produce melatonina a partire dalla serotonina.

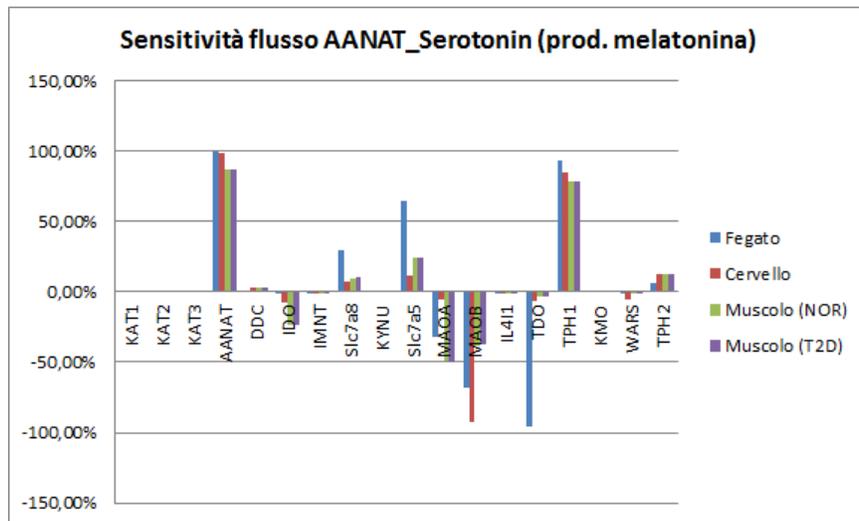


Figure 3.18: *Confronto dei valori di sensitività del flusso di reazione AANAT\_Serotonin (produzione di melatonina) rispetto a variazioni dell'espressione genica nei diversi tessuti e nei soggetti sani e diabetici.*

In questo caso si osserva immediatamente che tale flusso ha valori di sensitività molto diversi nel fegato rispetto agli altri tessuti, in particolare relativamente al gene TDO2, il cui aumento provoca un'importante diminuzione nel flusso solamente nel fegato e ai geni relativi alle reazioni di trasporto attraverso la membrana (Slc7a8 ed Slc7a5), che, ancora una volta, hanno un'influenza molto maggiore in questo tessuto rispetto agli altri.

Anche i geni MAOA e MAOB sembrano avere influenze diverse nei diversi tessuti: in tutti i casi un loro aumento produce una diminuzione del flusso, ma il comportamento nei tessuti (in particolare nel cervello) è molto diverso.

Infine osserviamo che soggetti sani e diabetici rispondono in maniera leggermente differente a variazioni del gene Slc7a8 nel muscolo.

In Figura 3.19 è visibile il confronto delle sensitività relative alla reazione spontanea che porta alla produzione di acido quinolinico.

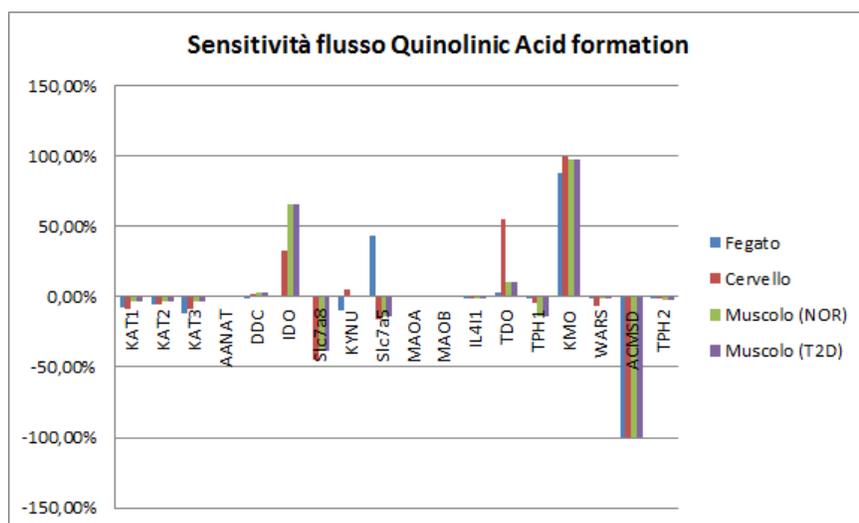


Figure 3.19: *Confronto dei valori di sensitività del flusso di reazione Quinolinic Acid formation (produzione di acido quinolinico) rispetto a variazioni dell'espressione genica nei diversi tessuti e nei soggetti sani e diabetici.*

Anche in questo caso si notano molte differenze tra tessuti diversi, in particolare poniamo in evidenza alcuni aspetti rilevanti:

- nel fegato si osservano dei valori di sensitività molto differenti rispetto agli altri tessuti in relazione al gene **IDO**, che nel fegato sembra essere ininfluente, mentre negli altri svolge un ruolo significativo (soprattutto nel muscolo), al gene **KYNU**, al cui aumento il flusso diminuisce nel fegato, aumenta nel cervello e rimane immutato nel muscolo e ai geni **TDO2** e **TPH1**, praticamente ininfluenti nel fegato rispetto agli altri tessuti (**TDO2** è particolarmente importante per questa reazione nel cervello);
- i geni **KMO** e **ACSMD** sono molto influenti in tutti i tessuti: il primo in verso positivo e il secondo in verso negativo;
- per quanto riguarda la produzione di acido quinolinico, soggetti sani e malati sembrano essere sensibili in egual misura a tutte le possibili variazioni geniche.

Infine, in Figura 3.20, osserviamo un analogo confronto relativamente ai flussi di reazione **KAT1\_3HKyn**, **KAT2\_3HKyn** e **KAT3\_3HKyn**, che sono state inglobate sotto il nome di **KAT\_3HKyn**.

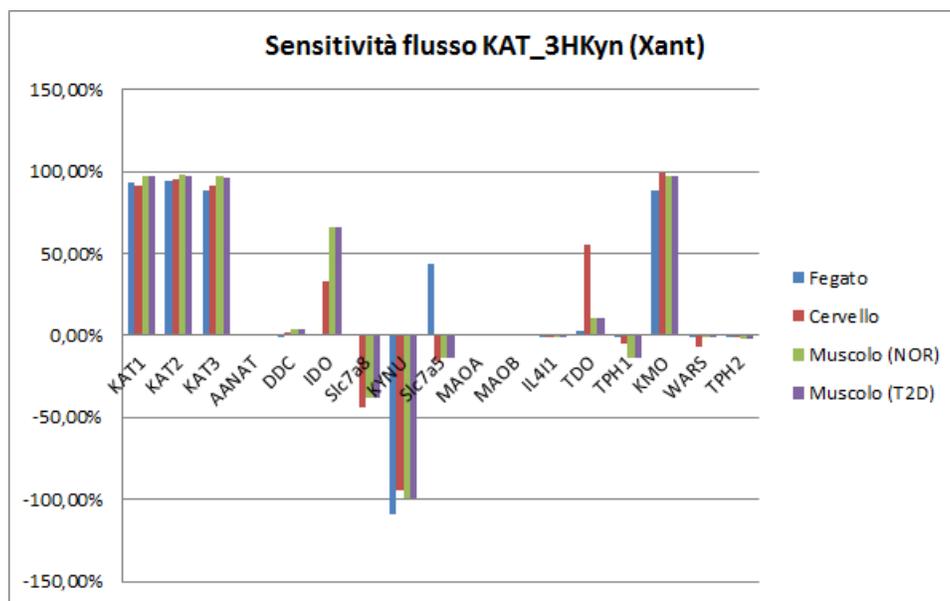


Figure 3.20: Confronto dei valori di sensitività del flusso di reazione **KAT\_3HKyn** (produzione di acido xanturenico) rispetto a variazioni dell'espressione genica nei diversi tessuti e nei soggetti sani e diabetici.

In questo caso osserviamo che:

- la produzione di acido xanturenico, in tutti i tessuti, è fortemente sensibile (in senso positivo) a variazioni dei geni **KAT1**, **KAT2**, **KAT3**. Questo risultato era aspettato, poichè tali sono i geni che regolano questa reazione;
- un aumento del gene **IDO** non ha alcuna influenza in questa reazione nel fegato, mentre provoca un aumento sensibile di produzione di xanturenico nel cervello e, soprattutto, nel muscolo scheletrico;
- viceversa, un aumento dei geni **Slc7a8** ed **Slc7a5** causa una diminuzione del flusso in cervello e muscolo, mentre nel fegato si osservano comportamenti differenti: il primo sembra non avere alcuna influenza, mentre il secondo porta ad un aumento significativo del flusso;
- il gene **TDO2** sembra essere particolarmente influente nel cervello, a differenza degli altri tessuti;
- entrambi i geni **KYNU** e **KMO** hanno un ruolo importante nella produzione di questo acido in tutti i tessuti: il primo in senso negativo e il secondo in senso positivo;
- anche se in misura non molto rilevante, anche le variazioni rispetto al gene **DDC** sono in verso opposto nel fegato rispetto agli altri;

- per quanto riguarda i geni KAT2 e KAT3, si osserva una leggera differenza nella sensibilità nei soggetti sani e malati nel muscolo scheletrico.

### 3.4.2.3 Considerazioni finali sull'analisi di sensibilità

A conclusione di queste diverse analisi di sensibilità, è possibile riassumere alcuni importanti risultati ottenuti, indicativi di quali geni siano più o meno influenti nelle reazioni e nelle concentrazioni di nostro interesse. In questo modo è possibile individuare quali possibili manipolazioni geniche possano essere utili al fine di modificare, se necessario, alcuni flussi di interesse.

Innanzitutto osserviamo che ci sono alcuni geni che hanno un'influenza significativa in tutte le reazioni e concentrazioni analizzate, in particolare:

- un aumento dell'espressione dei geni Slc7a5 e Slc7a5 porta ad un aumento della disponibilità di serotonina (soprattutto nel fegato) e una diminuzione di acidi quinolinico e xanturenico (ad eccezione del fegato in cui, in entrambi i casi, un aumento di Slc7a5 provoca un aumento anche degli acidi).
- Un aumento nella disponibilità del gene TDO2 provoca una diminuzione nella produzione di melatonina (soprattutto nel fegato) e un aumento degli acidi (soprattutto nel cervello). Una diminuzione di tale gene potrebbe quindi avere un risultato benefico per cercare di aumentare il livello di melatonina e ridurre la concentrazione degli acidi.
- Il gene KMO influenza in modo molto significativo gli acidi quinolinico e xanturenico: per diminuire la concentrazione e la produzione di tali acidi potrebbe essere utile limitare la diponibilità di questo gene. Una considerazione analoga si può fare relativamente al gene IDO (ad eccezione del fegato).
- Altri geni sono invece caratteristici di alcune reazioni in particolare. Un aumento della disponibilità di AANAT aumenta la produzione di melatonina a partire dalla serotonina. Un aumento di QPRT e ACMSD può aiutare a diminuire la concentrazione di acido quinolinico, così come un abbondanza del gene KYNU può diminuire l'acido xanturenico.

Poiché a livello pratico è più facile somministrare dei farmaci che possano inibire determinati geni, è più utile porre in evidenza quei geni la cui riduzione può portare effetti benefici. Riassumendo quindi:

1. la diminuzione di TDO2 comporta un aumento di produzione di serotonina e la diminuzione delle concentrazioni degli acidi (soprattutto nel cervello);
2. la diminuzione di KMO può causare una diminuzione nella produzione di acido quinolinico e xanturenico;
3. se si riduce la disponibilità di IDO, in particolare nel fegato, si osserva una diminuzione nella produzione degli acidi.

## Chapter 4

# Modelli basati su vincoli (constraint-based)

Come abbiamo già accennato precedentemente, il metabolismo può essere analizzato da due diversi punti di vista: strutturale e dinamico. Fino ad ora abbiamo studiato il metabolismo del triptofano dal punto di vista prevalentemente dinamico, mediante l'utilizzo di un modello di equazioni differenziali. Abbiamo però già avuto modo di osservare che, a causa dell'onerosa quantità di informazioni (parametri cinetici ed espressioni geniche) necessarie per eseguire un'analisi accurata di questo tipo, non è stato possibile soddisfare completamente le nostre domande iniziali. Per completare lo studio del metabolismo del triptofano dunque, affronteremo ora anche un tipo di analisi prevalentemente strutturale, basata sulla costruzione di un modello che fa riferimento ai vincoli dettati dalla matrice stechiometrica, che descrive le reazioni presenti nella rete tramite i cosiddetti coefficienti stechiometrici (successivamente sarà chiarito maggiormente questo concetto).

Utilizzando un modello di questo tipo si possono eseguire diverse analisi, tra le quali è nostro particolare interesse evidenziare la Flux Balance Analysis (FBA). Tale metodo si può considerare il punto di partenza per una grande quantità di metodi più elaborati, che sono stati sviluppati in seguito e che permettono di eseguire analisi più dettagliate e più informative. Anche per il nostro studio faremo riferimento ad alcune di tali metodologie.

Così come abbiamo fatto per quanto riguarda il modello cinetico, osserviamo inizialmente quali sono i vantaggi e gli svantaggi dell'utilizzo di questa metodologia di modellizzazione.

- Vantaggi dell'uso di un modello basato su vincoli strutturali:

1. tale modello è tessuto-indipendente e quindi è possibile utilizzare un modello unico, caratteristico di tutti i tessuti dell'organismo;
2. per costruire un modello di questo tipo è sufficiente la conoscenza delle equazioni di bilanciamento di massa e non sono richiesti parametri cinetici;
3. data la semplificazione nella costruzione del modello, è possibile analizzare anche modelli di grandi dimensioni.

- Svantaggi dell'uso di un modello basato su vincoli strutturali:

1. non è possibile effettuare un'analisi nel tempo e le soluzioni che si ottengono sono solo in stato stazionario;
2. la soluzione ottenuta non è necessariamente unica;
3. le informazioni che si possono ricavare sono quindi, in generale, meno informative.

Per analizzare il modello da questo punto di vista ci siamo serviti di COBRA Toolbox, un Toolbox scaricabile gratuitamente ed utilizzabile in ambiente Matlab.

Prima di proseguire con le successive analisi quindi è nostro interesse descrivere la FBA ed alcune metodologie ad essa collegate. Inoltre ci soffermeremo nella descrizione del Toolbox utilizzato, così da poter fornire una guida di riferimento per chiunque fosse interessato ad approcciarsi a questo tipo di strumenti.

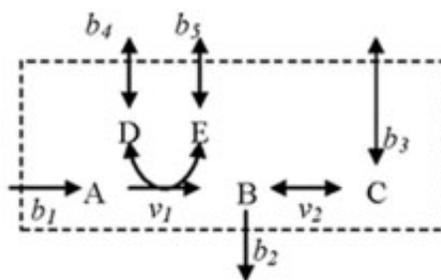


Figure 4.1: [23] *rete metabolica*.

## 4.1 Flux Balance Analysis (FBA)

### 4.1.1 Introduzione

La Flux Balance Analysis è un esempio di una modellazione che segue l'approccio constraint-based e che usa un'ottimizzazione lineare per determinare la distribuzione dei flussi di una rete metabolica in stato stazionario, attraverso la massimizzazione di una determinata funzione obiettivo. Tale funzione può essere, ad esempio, la produzione di ATP o il rate di crescita. L'ottimizzazione è possibile grazie alla conoscenza di un piccolo insieme dei parametri del sistema, combinata con l'applicazione di principi fondamentali della termodinamica e dell'evoluzione.

Per far questo, si assume che la cellula funzioni in modo ottimale rispettando le proprie funzioni metaboliche (massimizzazione della produzione di biomassa o minimizzazione dell'utilizzazione dei nutrienti).

Una volta determinata tale funzione obiettivo, si può risolvere il sistema di equazioni da cui si ottiene una distribuzione di flussi in stato stazionario, usata poi per interpretare le funzioni del sistema biologico in esame.

La FBA permette dunque la simulazione di sistemi sotto varie condizioni sperimentali: per questo può essere applicata in vari campi. Esistono inoltre molte diverse estensioni della Flux Balance Analysis standard, che integrano le informazioni di regolazione genica, oppure che simulano dinamicamente i fenotipi cellulari derivanti da reti integrate), Inoltre si sono sviluppati dei metodi che permettono l'aggiunta di ottimizzazioni multi-obiettivo o di massimizzare più funzioni obiettivo.

In pratica, la FBA consiste di quattro passi:

1. la definizione del sistema;
2. l'acquisizione della matrice stechiometrica;
3. la definizione di una funzione obiettivo biologicamente rilevante e l'aggiunta di vincoli biochimici;
4. l'ottimizzazione.

Per rendere la descrizione più chiara, faremo riferimento ad una semplice rete metabolica, che prenderemo come esempio (Figura 4.1 [23]).

#### 4.1.1.1 Definizione del sistema

Per definire il sistema è necessario elencare tutte le reazioni in termini di metaboliti, geni ed enzimi coinvolti nella catalizzazione delle reazioni. Inoltre occorre indicare se ciascuna reazione avviene in un particolare compartimento della cellula e se è reversibile oppure no.

In questa fase si identificano inoltre le reazioni di trasporto e i metaboliti esterni coinvolti, le risorse di carbonio, i co-fattori onnipresenti<sup>1</sup> e i prodotti finali del pathway<sup>2</sup>.

Per il nostro esempio consideriamo il sistema in Figura 4.1 [23].

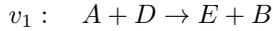
Chiamando con:

<sup>1</sup>**Cofattore:** si intende una piccola molecola di natura non proteica o uno ione metallico che si associa all'enzima e ne rende possibile l'attività catalitica. La maggior parte degli enzimi che richiedono il legame a cofattori, infatti, perdono ogni funzionalità in caso di sua assenza.

<sup>2</sup>**Prodotto finale di un pathway:** si intende la sostanza che risulta essere effettivamente prodotta in seguito alle reazioni che avvengono all'interno del pathway. Per quanto riguarda le tipologie di prodotti, si rimanda al paragrafo 1 sulla descrizione del metabolismo.

- $A, B, C, D, E$  i metaboliti;
- $v_1, v_2$  i flussi delle reazioni interne al sistema;
- $b_1, b_2, b_3, b_4, b_5$  i flussi delle reazioni di scambio con l'esterno,

possiamo quindi descrivere le reazioni nel seguente modo:



$b_1, b_2, \dots, b_5$ : reazioni di scambio con l'esterno per  $A, B, \dots, E$

#### 4.1.1.2 Acquisizione della matrice stechiometrica

Se nella rete sono presenti  $m$  metaboliti (nel nostro esempio  $m = 5$ ) ed  $n$  reazioni (nel nostro esempio  $n = 2 + 5 = 7$ ) che coinvolgono tali metaboliti, il bilancio di massa di un sistema metabolico è descritto dalla matrice stechiometrica  $S \in \mathbb{R}^{m \times n}$  che mette in relazione i rate di flusso (interni  $v_i$  e di scambio  $b_i$ ) delle reazioni  $v \in \mathbb{R}^{n \times 1}$  con le derivate temporali delle concentrazioni metaboliche  $x \in \mathbb{R}^{m \times 1}$ :  $\frac{dx}{dt} = Sv$ , dove  $v = [v_1 v_2 \dots v_{n_i} | b_1 b_2 \dots b_{n_{ext}}]^T$  (gli  $x_i$  rappresentano le concentrazioni dei metaboliti, i  $v_i$  rappresentano i flussi interni, mentre i  $b_i$  i flussi di scambio nel sistema.  $n_i$  è quindi il numero di flussi interni, mentre  $n_{ext}$  è il numero di flussi di scambio con l'esterno). Gli elementi  $S_{ij}$  della matrice stechiometrica rappresentano quindi gli indici stechiometrici, il segno dei quali indica se il metabolita viene prodotto (segno positivo) oppure consumato (segno negativo).

Nel nostro esempio avremo quindi la matrice stechiometrica  $S$ :

$$\begin{array}{c}
 v_1 \quad v_2 \quad b_1 \quad b_2 \quad b_3 \quad b_4 \quad b_5 \\
 A \left( \begin{array}{cc|cccc}
 -1 & 0 & 1 & 0 & 0 & 0 & 0 \\
 1 & -1 & 0 & -1 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
 -1 & 0 & 0 & 0 & 0 & 1 & 0 \\
 1 & 0 & 0 & 0 & 0 & 0 & 1
 \end{array} \right) \\
 B \\
 C \\
 D \\
 E
 \end{array}$$

e il vettore  $v = [v_1 \quad v_2 \quad | \quad b_1 \quad b_2 \quad b_3 \quad b_4 \quad b_5]^T$ ;

In condizione di stato stazionario, ovviamente si ha:

$$\frac{dx}{dt} = Sv = 0.$$

Quindi, nel nostro esempio:

$$\frac{d}{dt} \begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix} = Sv = S \begin{bmatrix} v_1 \\ v_2 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix} = 0.$$

Di conseguenza, la soluzione verrà ricercata nel Kernel della matrice  $S$ .

I vincoli aggiuntivi sono necessari per la risoluzione del problema, in quanto spesso sono più numerose le reazioni ( $n$ ) dei metaboliti ( $m$ ): il problema perciò è sotto-determinato e contiene  $n - m$  gradi di libertà. Tali vincoli sono solitamente di quattro tipi:

1. vincoli chimico-fisici;
2. vincoli spaziali o topologici;
3. condizioni dipendenti da vincoli ambientali;
4. vincoli di regolazione, ovvero vincoli temporanei che nascono legati ad un particolare ambiente.

Si rivela perciò necessario imporre tali vincoli attraverso la valutazione di determinati flussi: solitamente, tali costrizioni sono inserite come lower ed upper bound dei vettori dei flussi, ad esempio:

$$0 < v_i < \infty$$

$$-\infty < b_i < \infty.$$

Nella pratica, si definisce un upper bound finito se si hanno delle conoscenze a priori sulla termodinamica delle cellule o su una misurazione effettuata.

Supponiamo, nel nostro esempio di avere i seguenti vincoli:

$$0 \leq v_1 \leq 10$$

$$-10 \leq v_2 \leq 10$$

$$0 \leq b_1, b_2 \leq 10$$

$$-10 \leq b_3, b_4, b_5 \leq 10$$

#### 4.1.1.3 Definizione della funzione obiettivo

La funzione obiettivo deve rappresentare lo scopo biochimico del sistema stesso.

Una funzione obiettivo lineare si rappresenta nel modo seguente:

$$\max_v f^T v \quad s.t. \quad Sv = 0,$$

dove  $f$  rappresenta la funzione obiettivo scelta, espressa in termini di flussi.

Esempi più comuni di funzioni obiettivo sono la massimizzazione della crescita (biomassa), la minimizzazione di produzione di ATP, la minimizzazione dell'uso di nutrienti, la massimizzazione della produzione di metaboliti (in particolare per determinare le funzioni di una particolare cellula), la massimizzazione della biomassa assieme alla produzione di metaboliti e la minimizzazione della norma del vettore di flusso.

Ovviamente la scelta di tale funzione è influenzata dall'obiettivo dello studio in atto (l'esplorazione delle funzioni e delle limitazioni di una rete biologica, oppure l'analisi della robustezza di una determinata rete, a titolo di esempio).

Nel nostro esempio supponiamo di voler massimizzare la produzione del metabolita  $B$  (vogliamo quindi massimizzare il flusso  $b_2$ ): definiamo quindi il vettore dei coefficienti  $f = [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0]^T$  e poniamo:

$$\max_v f^T v = [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0] \begin{bmatrix} v_1 \\ v_2 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix} \quad s.t. \quad Sv = \begin{bmatrix} -1 & 0 & | & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & | & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & | & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & | & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & | & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix} = 0.$$

Come possiamo osservare quindi si tratta di risolvere un problema di ottimizzazione lineare (LP), soggetto a dei vincoli dettati dalla matrice stechiometrica e da eventuali vincoli legati ad altri tipi di conoscenze, che limitano il valore ottenibile per ciascun flusso.

Occorre inoltre sottolineare che non è necessario massimizzare un solo metabolita per volta: il vettore  $f$  può contenere più di un coefficiente non nullo, a cui si può attribuire anche un valore diverso da 1. Bisogna però fare molta attenzione al risultato ottenuto: il valore che verrà massimizzato infatti sarà il prodotto di  $f^T v$ , che corrisponde alla combinazione lineare dei flussi (interni e di scambio) attraverso i coefficienti contenuti in  $f$ . Ad esempio, se

$$f^T = [1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0],$$

verrà massimizzata la somma

$$v_1 + b_2.$$

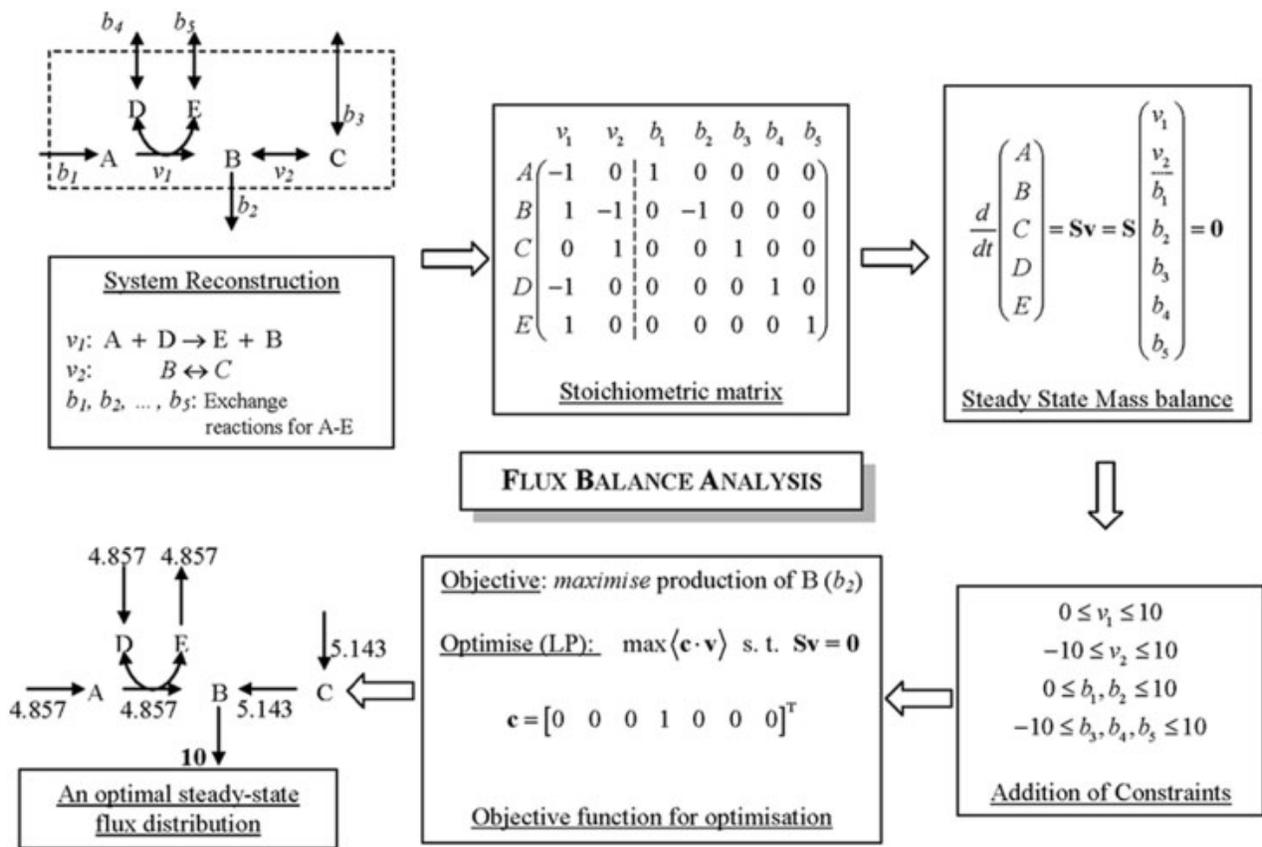


Figure 4.2: [23] *Riassunto FBA*.

Se il nostro scopo è effettivamente quello di massimizzare il valore di tale somma, questo approccio è corretto, ma se vogliamo massimizzare contemporaneamente i due singoli flussi, dobbiamo andare a verificare se il risultato ottenuto è effettivamente quanto stavamo ricercando: tale ottimizzazione infatti può portare anche a delle soluzioni la cui somma dei flussi è effettivamente la più grande possibile, ma che corrisponde ad un valore nullo o comunque non massimo per i flussi presi singolarmente. È questo il motivo per cui in seguito si sono sviluppate altre varianti della FBA che permettono di ottimizzare più funzioni contemporaneamente oppure di implementare una ottimizzazione multi-obiettivo.

#### 4.1.1.4 Ottimizzazione

Nella fase di ottimizzazione viene risolto il sistema di equazioni definito dalla funzione obiettivo, sottostante ai vincoli precedentemente definiti. Il risultato consiste nel vettore  $\hat{v}$  della distribuzione dei flussi in stato stazionario.

Nel nostro esempio, una distribuzione di flussi che risolve il nostro problema è:

$$\hat{v} = [ 4.857 \ 5.143 \ | \ 4.857 \ 10 \ 5.143 \ 4.857 \ 4.857 ]^T.$$

La Figura 4.2 [23] riassume l'esempio appena descritto di analisi FBA.

Per risolvere problemi LP (linear programming) esistono numerosi tool: i più utilizzati sono i seguenti:

- lp\_solve (<http://sourceforge.net/projects/lpsolve/>);
- GLPK (<http://www.gnu.org/software/glpk/>);
- LINDO ([www.lindo.com/](http://www.lindo.com/));
- CPLEX;
- MOSEK (<http://www.mosek.com/>).

**lp\_solve** lp\_solve è un risolutore, scaricabile liberamente, di problemi LP, basato sul metodo del simplesso rivisitato e sul metodo Branch-and-bound per gli interi.

È in grado di risolvere modelli puramente lineari, misti (integer/binary), semi-continui ed SOS (special ordered sets). Non richiede un limite per la dimensione del modello ed accetta come file di input in formato lp o mps, ma si può ottenere un'estensione. È importante notare però che alcuni modelli possono richiedere molto tempo per essere risolti o, addirittura non essere risolti: la probabilità di non riuscita aumenta con la dimensione del modello in input.

**GLPK** Il package GLPK (GNU Linear Programming Kit) viene utilizzato per risolvere problemi LP large-scale, misti a problemi integer programming (MIP) e ad altri problemi correlati. Consiste in un insieme di routine scritte in ANSI C e organizzate in forma di libreria richiamabile.

Supporta il linguaggio GNU MathProg modeling language, che è un sottoinsieme del linguaggio AMPL.

**LINDO** LINDO (Linear, INteractive and Discrete Optimizer) è un tool disponibile in commercio per la soluzione di modelli lineari. Esso consente sia un approccio interattivo che un approccio mediante gestione di file.

LINDO Systems ha inoltre sviluppato l'applicazione LINDO Application Programming Interface (API), per risolvere una grande gamma di problemi di ottimizzazione, inclusi linear programs, mixed integer programs, quadratic programs e general nonlinear non-convex programs.

**CPLEX** CPLEX (o meglio IBM ILOG CPLEX Optimization Studio) prende il nome dall'algoritmo del simplesso implementato in linguaggio C, anche se attualmente comprende anche algoritmi addizionali di programmazione matematica e offre interfacce verso altri ambienti/linguaggi.

Risolve problemi di programmazione lineare intera (anche di notevoli dimensioni), utilizzando le varianti primale o duale del metodo del simplesso o metodi di punto interno, problemi di programmazione quadratica convessa e problemi con vincoli quadratici convessi, utilizzando tecniche di Second-order cone programming (SOCP).

È consentita l'interfaccia verso i linguaggi C++, C#, Java e verso Python (basata su C). Sono stati sviluppati dei connettori verso Microsoft Excel e Matlab. Inoltre è disponibile un eseguibile iterativo "stand-alone", utile per il debugging ed il "performance tuning" dei modelli di ottimizzazione.

CPLEX è utilizzabile tramite software di modellizzazione indipendenti, come AIMMS, AMPL, GAMS, MPL, OpenOpt, OptimJ e TOMLAB.

**MOSEK** MOSEK è un tool per risolvere problemi di ottimizzazione matematica, tra i quali: problemi di programmazione lineare, di programmazione quadratica, problemi conici e problemi mixed integer. Tale strumento trova applicazione soprattutto negli ambiti finanziario, forestale e agricolo.

## 4.2 Varianti della FBA

Uno dei limiti maggiori della Flux Balance Analysis consiste nella determinazione di una sola tra tutte le possibili soluzioni. Infatti, i vincoli che vengono utilizzati per eseguire l'ottimizzazione non garantiscono necessariamente né l'esistenza di una soluzione, né tantomeno l'unicità di tale soluzione. Per questo motivo sono state sviluppate delle varianti della FBA, che hanno lo scopo di scegliere attraverso determinati criteri una soluzione 'ottima' particolare, oppure che permettono di visualizzare l'intero spazio delle soluzioni possibile.

Per quanto riguarda il primo obiettivo, faremo riferimento a due tecniche:

1. Geometric FBA;
2. Parsimonious FBA (pFBA).

Per ottenere una descrizione globale dello span delle soluzioni possibili invece verranno proposte soluzioni quali:

1. Flux Variability Analysis (FVA);
2. Tecniche di campionamento (Sampling);
3. Tecniche di Analisi legate a concetti quali 'extreme pathways' e 'Reaction Participation'.

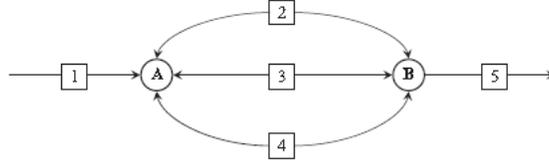


Figure 4.3: [12] *Esempio di semplice rete metabolica che ammette soluzioni multiple: il valore dei flussi delle reazioni 2, 3 e 4 può essere qualunque.*

## 4.2.1 Geometric FBA

Uno dei limiti maggiori della Flux Balance Analysis descritta precedentemente consiste nel fatto che la soluzione, espressa in termini di distribuzione di flussi attraverso le reazioni della rete metabolica non è unica. La nozione matematica di 'equivalenza' tra stati ottimi assunti dalla rete si traduce in biologia come 'fenotipi silenti' (silent phenotypes). Questa proprietà distingue la branca della biologia che si occupa della modellizzazione 'in silico' dalle scienze fisico-chimiche, dove è desiderabile ottenere una soluzione unica.

Nonostante molte soluzioni equivalenti possono avere una certa importanza nella rappresentazione biologica, spesso si vorrebbe poter scegliere una singola soluzione, che sia bene definita e che possa essere in qualche senso rappresentativa di tutto lo spazio delle soluzioni possibili. A tale scopo molti ricercatori spesso scelgono una determinata soluzione rispetto ad un'altra in base a scelte effettuate in modo randomico a partire da uno spazio molto ampio di soluzioni possibili: questa metodologia però porta a dei risultati che molte volte non sono riproducibili e che sono fortemente dipendenti dal tipo di algoritmo utilizzato per risolvere il problema LP.

La Geometric FBA si pone come obiettivo quello di ottenere una soluzione ottima (ottimo locale) che possa essere considerata unica tra tutte le soluzioni dello spazio di quelle ammissibili, basandosi sulla geometria di tale spazio. I metodi proposti dagli autori sono due e si differenziano tra loro in quanto il primo si basa su un teorema fondamentale della matematica ma non può essere implementato in reti di dimensione troppo elevata, mentre il secondo è computazionalmente applicabile a reti anche più estese, a discapito di una base teorica meno consistente (tale metodo permette di ottenere risultati simili al primo).

### 4.2.1.1 Metodi

Per chiarire maggiormente i concetti che verranno esposti, faremo riferimento ad un esempio corrispondente alla rete presentata in Figura 4.3 [12], di cui vorremmo ottenere la distribuzione dei flussi per massimizzare il flusso in uscita  $v_5$ . Tale rete è soggetta ad un vincolo che limita la quantità di nutrienti in entrata (supponiamo, ad esempio,  $v_1 \leq 1$ ). In termini matematici, il nostro problema si traduce nel risolvere l'ottimizzazione:

$$\max Z = f^T v$$

$$\text{subject to } Sv = 0,$$

$$v^{\min} \leq v \leq v^{\max},$$

dove:

$$f = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad S^T = \begin{bmatrix} 1 & 0 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ 0 & -1 \end{bmatrix}, \quad v_{\min} = \begin{bmatrix} 0 \\ -\infty \\ -\infty \\ -\infty \\ 0 \end{bmatrix}, \quad v_{\max} = \begin{bmatrix} 1 \\ \infty \\ \infty \\ \infty \\ \infty \end{bmatrix}.$$

La soluzione di tale FBA porta ad un unico valore massimo  $Z^* = 1$ , ma le distribuzioni dei flussi che permettono di ottenere questo valore sono infinite, poichè infinite sono le combinazioni dei flussi attraverso le reazioni 2, 3 e 4 che portano al risultato ottimo. Ad esempio, una possibile soluzione è

$$v^* = (1 \quad 1 \quad 1000 \quad -1000 \quad 1)^T.$$

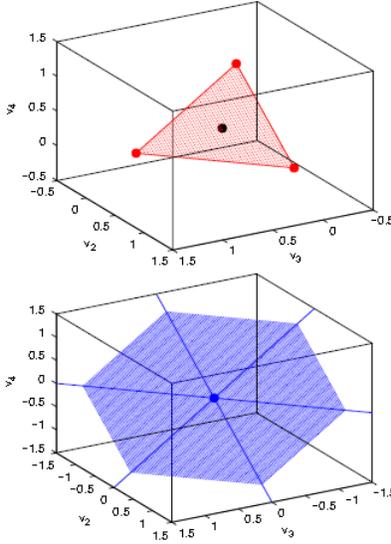


Figure 4.4: [12] *L'insieme delle soluzioni ottime della FBA può essere decomposto nella somma di un inviluppo convesso (in rosso) e in un cono di raggi estremali (in blu). Una scelta naturale come soluzione unica è data dal centro dell'inviluppo (in nero nella Figura in alto).*

La Geometric FBA ha quindi lo scopo di selezionare una sola soluzione ottima all'interno dell'infinito spazio delle soluzioni ottime ammissibili, che sia contemporaneamente unica (e quindi riproducibile, indipendentemente dall'algoritmo utilizzato per risolvere il problema LP), sia ragionevole dal punto di vista biologico.

**Metodo uno** L'insieme di tutte le possibili soluzioni della FBA sono date dall'equazione e dal vincolo seguenti:

$$Av = b, \quad v^{min} \leq v \leq v^{max}, \quad (1)$$

dove

$$A = \begin{pmatrix} S \\ f^T \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ Z^* \end{pmatrix}.$$

Da un punto di vista geometrico, l'Equazione (1) definisce un poliedro. Facendo riferimento alla rappresentazione geometrica (e sfruttando il fatto che  $b \neq 0$ ), il teorema di "Minkowski" (1910) ci suggerisce che tale poliedro può essere decomposto nella somma di una combinazione convessa e di un cono, come è rappresentato in Figura 4.4 [12] ed espresso nella formula seguente:

$$v \in \underbrace{\left\{ \sum_i \lambda_i x_i : \lambda_i \geq 0, \sum \lambda_i = 1 \right\}}_{\text{combinazione convessa}} + \underbrace{\left\{ \sum_j \mu_j y_j : \mu_j \geq 0 \right\}}_{\text{cono}}.$$

La combinazione convessa è definita in modo univoco dai suoi vertici  $x_i$ , mentre il cono è definito in modo non univoco dai suoi raggi estremali  $y_i$ . Tali vertici e raggi possono essere ottenuti tramite l'applicazione dell'algoritmo di Chernikova (Chernikova, 1965; LeVerge, 1992).

In particolare, nel nostro esempio abbiamo tre vertici e sei raggi:

$$x_i = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad y_i = \begin{pmatrix} 0 \\ \pm 1 \\ \mp 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \pm 1 \\ 0 \\ \mp 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ \pm 1 \\ \mp 1 \\ 0 \end{pmatrix}.$$

Da questo risultato si può capire che i raggi estremali  $y_i$  definiscono i cicli interni nella rete. Le soluzioni che ci possono interessare si trovano invece all'interno della combinazione convessa: il punto centrale di tale combinazione è univocamente definito da:

$$v^* = \frac{1}{n} \sum_{i=1}^n x_i = \begin{pmatrix} 1 \\ 1/3 \\ 1/3 \\ 1/3 \\ 1 \end{pmatrix}.$$

Questa è certamente la soluzione più naturale per la nostra Geometric FBA, poichè nessun flusso viene utilizzato soltanto in cicli interni, senza contribuire all'ottimo ricercato. Inoltre, se non si è a conoscenza di alcuna informazione sperimentale riguardo a come le reazioni 2, 3 e 4 vengano usate, tale risultato assume in modo imparziale che ci sia un flusso uguale in tutte e tre le reazioni.

Purtroppo la complessità dell'algoritmo di Chernikova, utilizzato da questo primo metodo, cresce in modo esponenziale con il numero di reazioni presenti nella rete, perciò non è implementabile in sistemi metabolici complessi.

Per questo motivo viene proposto anche un secondo metodo alternativo, che può essere utilizzato anche in sistemi più estesi.

**Metodo due** Facendo riferimento al metodo uno, possiamo interpretare questo secondo metodo in questo modo: poichè non è computazionalmente possibile definire i vertici di una combinazione convessa di una rete estesa, il problema di trovare il più piccolo involucro che possa contenere tutta la combinazione viene visto come un insieme di problemi LP che siano singolarmente risolvibili.

Per cercare di chiarire il metodo, facciamo ancora riferimento all'esempio proposto in Figura 28. A partire dall'equazione (1), l'algoritmo si può così descrivere in pseudo-codice:

```

n := 1
μ1 := (0, ..., 0)
while Δ > ε do
  Zn = min ||v - μn||1 subject to
  Av = b, vmin ≤ v ≤ vmax, ||v - μn||1 = Zi,    i = 1, ..., n - 1
  for each do
    vjmax ↔ max vj subject to Av = b, vmin ≤ v ≤ vmax, ||v - μn||1 = Zi,    i = 1, ..., n
    vjmin ↔ min vj subject to Av = b, vmin ≤ v ≤ vmax, ||v - μn||1 = Zi,    i = 1, ..., n
  n ↔ n + 1
  μn := 1/2 (vmax - vmin)
  Δ ↔ maxi |vimax - vimin|

```

dove  $\|v\|_1 = \sum_i |v_i|$  è la distanza di Manhattan ed  $\varepsilon$  è la specifica tolleranza.

Per capire meglio l'algoritmo, esaminiamo cosa accade applicandolo al nostro esempio:

- Iterazione 1)

Inizialmente assumiamo ragionevolmente (come proposto da *Holzhütter* nel 2004) che la cellula operi per minimizzare il flusso totale ( $\sum_i |v_i|$ ) richiesto per produrre il suo obiettivo  $Z = Z^*$ : questo problema può essere visto come un problema LP, se scomponiamo  $v$  nelle sue componenti positiva e negativa. Questa assunzione si basa sul fatto che il flusso attraverso le reazioni è proporzionale alla concentrazione enzimatica, perciò questa scelta si può interpretare come la volontà di minimizzare la quantità di enzimi richiesti per ottenere l'obiettivo.

Il passo successivo assicura che vengano rimossi tutti i cicli interni che possono produrre flussi  $v_i = \infty$ . Perciò, ritornando alla Figura 4.4, lo spazio delle soluzioni al nostro nuovo problema è dato esattamente dalla combinazione convessa, mentre il cono è vuoto.

Dopo aver trovato il minimo flusso totale attraverso la rete ( $Z_1 = 3$ ), dobbiamo ora trovare i limiti per ciascun flusso, soggetto a questo nuovo vincolo (in pratica è necessario implementare una Flux Variability Analysis, di cui parleremo più approfonditamente in seguito). Ovviamente, nel nostro caso di esempio troveremo  $v_i = 1, \forall i = 1, \dots, 5$ , perciò  $0 \leq v_i \leq 1$  per  $i = 2, 3, 4$ , in particolare. Questi limiti vengono quindi aggiunti come nuovi vincoli del problema da risolvere. Il punto centrale si può calcolare come la media di questi nuovi vincoli:  $\mu_2 = (1 \quad \frac{1}{2} \quad \frac{1}{2} \quad \frac{1}{2} \quad 1)^T$  è quindi la soluzione più rappresentativa tra tutte le soluzioni ottenibili in questa prima iterazione.

- Iterazione 2)

Mentre nel primo metodo descritto era possibile trovare e quindi scegliere come soluzione rappresentativa il centro della combinazione convessa ( $v^*$ ), in questo caso possiamo solo limitarci a costruire una 'scatola' attorno a tale combinazione convessa, attraverso la definizioni di nuovi limiti. Purtroppo però il centro di tale box, calcolato al passo precedente ( $\mu_2$ ) non è necessariamente la soluzione del problema originale. Perciò ora cerchiamo di minimizzare la distanza tra tale distribuzione di flussi e il centro, vincolando il problema al nuovo vincolo  $Z_1$ , ottenuto all'iterazione precedente.

Eseguito questa operazione, si ottengono nuovi vincoli:  $0 \leq v_i \leq \frac{1}{2}$  per  $i = 2, 3, 4$ , che corrispondono al nuovo centro  $\mu_3 = (1 \quad \frac{1}{4} \quad \frac{1}{4} \quad \frac{1}{4} \quad 1)^T$ .

- Iterazione n)

In pratica, ogni iterazione dell'algoritmo aggiunge un nuovo vincolo al problema, e il nuovo vettore dei flussi viene posto sempre al centro dei vincoli ottenuti fino a quel momento: in questo modo restringiamo sempre di più la 'scatola' che racchiude lo spazio delle soluzioni possibili. Dopo un numero finito di iterazioni, si osserva che i vincoli convergono ad un'unica soluzione, con una tolleranza definita. In particolare, nel nostro semplice esempio è necessario effettuare ben 22 iterazioni, con una tolleranza di  $\varepsilon = 10$ . La soluzione finale ottenuta è:

$$v^* = (1 \quad \frac{1}{3} \quad \frac{1}{3} \quad \frac{1}{3} \quad 1)^T,$$

che, come possiamo osservare, in questo caso coincide con la soluzione ottenuta nel metodo uno.

#### 4.2.1.2 Osservazioni conclusive

Il maggior vantaggio della Geometric FBA è quello di restituire una soluzione unica e ben definita tra tutte quelle ottenibili con la Flux Balance Analysis. Tale risultato è completamente riproducibile e non dipende dal risolutore utilizzato.

Ovviamente però, dal punto di vista biologico, la distribuzione esatta dei flussi che viene utilizzata dalla cellula dipende da una grande varietà di stimoli e non è possibile determinarla sulla base di informazioni inerenti soltanto alla struttura della rete. Non è detto quindi che tale algoritmo produca il risultato corretto in questo senso; esso però fornisce la distribuzione minima di flussi necessari per garantire l'obiettivo richiesto. Se si assume che la distribuzione dei flussi è legata alla quantità di enzimi necessari, questo risultato è equivalente alla minimizzazione della quantità di enzimi necessaria per raggiungere l'obiettivo. Inoltre tale soluzione rimuove tutti quei cicli interni che porterebbero a dei flussi irrealizzabili dal punto di vista termodinamico. In pratica quindi, la soluzione proposta dalla Geometric FBA si può considerare 'centrale' rispetto allo spazio di tutte le soluzioni ottime e può quindi definirsi rappresentativa per tutte le soluzioni, in assenza di altre conoscenze.

Non è da sottovalutare però la compessità computazionale di tale algoritmo, che cresce molto velocemente con la dimensione della rete in esame.

## 4.2.2 Parsimonious Flux Balance Analysis (pFBA)

La pFBA è una variante della Flux Balance Analysis standard, che viene utilizzata per determinare se l'espressione di geni e proteine supportano l'ottimizzazione ottenuta massimizzando il flusso obiettivo. Tale metodo utilizza le simulazioni in silico per identificare le proprietà funzionali dei geni presenti nel pathway sotto determinate condizioni di crescita. In pratica la pFBA introduce un nuovo vincolo rispetto alla FBA standard, che necessita della conoscenza dei geni corrispondenti alle reazioni della rete.

La pFBA assume che in situazione di crescita esponenziale si abbia una selezione a favore delle strutture che crescono più velocemente e che richiedono il minor flusso totale attraverso la rete metabolica (si cerca quindi di minimizzare la massa totale di enzimi necessari per implementare la soluzione ottima). Questo vincolo aggiuntivo introduce un miglioramento nella FBA standard. Va osservato che, nonostante questa assunzione non sia sempre verificata nella realtà, può essere considerata ragionevole in una vasta molteplicità di casi.

La parsimonious FBA quindi si presenta come una ottimizzazione di tipo 'bilevel LP', volta a calcolare la distribuzione ottima di flussi che garantisca la massima crescita e, allo stesso tempo, la minimizzazione della somma di tutti i geni associati ai flussi delle reazioni. Operando in questo senso, la pFBA riesce a

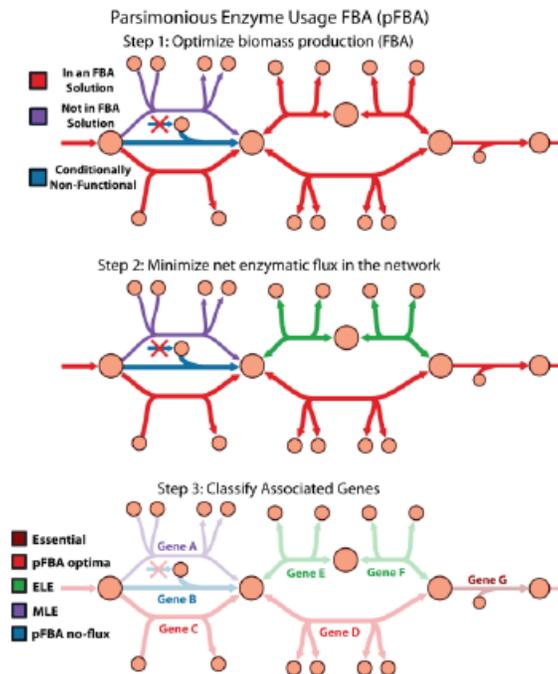


Figure 4.5: [17] **pFBA**: *step1) FBA standard per ottimizzare il rate di crescita; step2) minimizzazione dei flussi legati alle reazioni associate ai geni; step3) la pFBA trova il sottoinsieme di geni e proteine che può contribuire in modo più efficiente alla rete sotto i determinati vincoli di crescita. Classifica perciò i geni e le proteine in 5 classi: 1) Essential; 2) pFBA optima; 3) enzymatically less efficient (ELE), che richiedono più step enzimatici rispetto ad altri pathway alternativi che rispondono alle stesse esigenze della cellula; 4) metabolically less efficient (MLE), che richiederebbero una diminuzione in termini di crescita, se venissero utilizzati; 5) pFBA no-flux, che non sono in grado di trasportare alcun flusso sotto le condizioni definite.*

determinare il sottoinsieme di geni e proteine che può contribuire in modo più efficiente alla rete sotto i determinati vincoli di crescita. Opera quindi una classificazione di geni e proteine in 5 classi:

1. Essential: tali geni/reazioni sono essenziali al fine di poter ottimizzare la funzione obiettivo selezionata;
2. pFBA optima: classe che contiene i geni/le reazioni che permettono di ottenere contemporaneamente il valore ottimo massimo possibile della funzione obiettivo e la minima attività enzimatica possibile;
3. enzymatically less efficient (ELE), che richiedono più step enzimatici rispetto ad altri pathway alternativi che rispondono alle stesse esigenze della cellula;
4. metabolically less efficient (MLE), che richiederebbero una diminuzione in termini di crescita, se venissero utilizzati;
5. pFBA no-flux, che non sono in grado di trasportare alcun flusso sotto le condizioni definite.

In Figura 4.5 [17] possiamo visualizzare quanto appena descritto in un esempio: nella situazione rappresentata, il gene A (classificato come MLE) rappresenta un enzima che usa un co-fattore subottimo per catalizzare una reazione, perciò, se viene utilizzato, causa una riduzione del rate di crescita. Il gene B invece (classificato come pFBA no-flux) non è in grado di trasportare alcun flusso nelle condizioni descritte, poichè non può ricevere né produrre un metabolita precursore necessario. I geni E ed F richiedono due enzimi differenti per catalizzare la stessa reazione che il gene D può catalizzare da solo: per questo motivo E ed F sono classificati come ELE. Poichè la rimozione del gene G provocherebbe un blocco totale dei flussi nella rete, tale gene viene classificato come Essential. Infine, i geni C e D dell'esempio rappresentano il pathway più efficiente (in termini topologici e metabolici) e perciò appartengono all'insieme dei geni classificati come pFBA optima.

Per capire meglio come viene implementata la classificazione dei geni, possiamo osservare la Figura 4.6 [17].

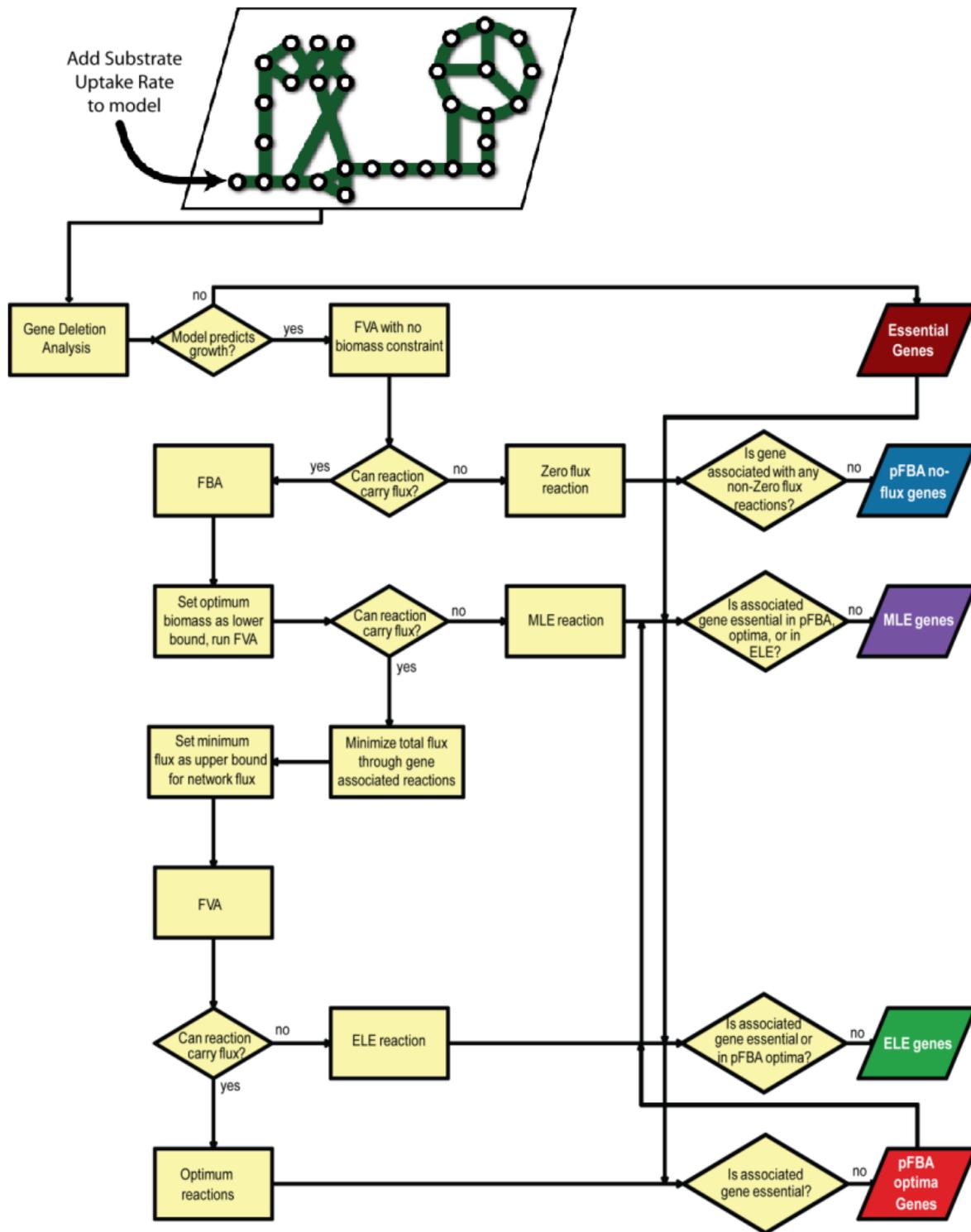


Figure 4.6: [17] **Schema rappresentativo della simulazione e della classificazione dei geni mediante pFBA.** 1) Delezione di geni + FBA standard per testare l'essenzialità delle reazioni; 2) FVA, (senza vincolo di biomassa), per determinare gli eventuali geni che non sono in grado di trasportare alcun flusso. 3) si utilizza il valore ottimo ottenuto dalla precedente FBA come limite inferiore per la reazione da ottimizzare e viene effettuata una FVA per determinare le reazioni MLE (metabolically less-efficient). 4) minimizzazione dei flussi delle reazioni associate ai geni, (risoluzione di un problema LP) e il risultato viene settato come upper bound della somma di tutti i flussi presenti nella rete. 5) FVA (con vincoli di massima crescita e minima somma di flussi, ottenuti nei passi precedenti): si determinano così le reazioni che sono attive se sottoposte ad entrambi i vincoli. In relazione ai set di reazioni che si sono così venuti a determinare, vengono classificati in conseguenza anche i geni.

1. Inizialmente viene utilizzata la FBA standard per testare l'essenzialità delle reazioni (cioè viene eseguita dopo aver eliminato di volta in volta ciascun gene: se l'eliminazione di un particolare gene non permette di ottenere alcuna soluzione, significa che tale gene è essenziale per l'obiettivo della FBA);
2. in seguito si implementa una Flux Variability Analysis (FVA), eliminando il vincolo della biomassa, per determinare gli eventuali geni che non sono in grado di trasportare alcun flusso.
3. Come terzo step, si utilizza il valore ottimo ottenuto dalla precedente FBA come limite inferiore per la reazione da ottimizzare nei passaggi successivi. Viene quindi nuovamente effettuata una FVA per determinare le reazioni MLE (metabolically less-efficient).
4. A questo punto si esegue una minimizzazione dei flussi delle reazioni associate ai geni, attraverso la risoluzione di un problema LP e il risultato viene settato come upper bound della somma di tutti i flussi presenti nella rete.
5. Infine un'ulteriore FVA viene effettuata sottoponendo la rete ai vincoli di massima crescita e minima somma di flussi, ottenuti nei passi precedenti: si determinano così le reazioni che sono attive se sottoposte ad entrambi i vincoli.

In relazione ai set di reazioni che si sono così venuti a determinare, vengono classificati in conseguenza anche i geni: tutti i geni necessari per la crescita in silico, vengono definiti 'Essential', i geni non essenziali associati alle reazioni che rimangono attive massimizzando la crescita e minimizzando la somma dei flussi, vengono definiti 'pFBA optima', i geni associati alle reazioni che non sono in grado di trasportare alcun flusso sono detti geni 'pFBA no-flux'. Si considerano appartenenti alla classe 'ELE' quei geni associati a reazioni che possono trasportare flusso mentre si massimizza la biomassa, ma che non possono farlo se contemporaneamente si minimizza la somma dei flussi. Infine, i geni rimanenti, che corrispondono a quelle reazioni che possono trasportare flusso solo se non si ottimizza la biomassa, vengono associati all'insieme degli 'MLE'.

### 4.2.3 Flux Variability Analysis (FVA)

Abbiamo osservato più volte che la FBA standard spesso genera soluzioni multiple in termini di distribuzioni di flusso. La Flux Variability Analysis (FVA) è una variante della FBA, che è in grado di determinare il range del valore del flusso all'interno del quale può variare ciascuna reazione della rete, per avere lo stesso valore ottimo del flusso obiettivo. In pratica, la FVA viene utilizzata per trovare il valore massimo e minimo del flusso attraverso ciascuna reazione della rete, mantenendo il valore ottimo del flusso obiettivo (calcolato tramite FBA), ad una determinata percentuale del suo valore massimo (ad esempio si può vincolare il flusso attraverso la reazione obiettivo al 100% del suo valore ottimo).

Questa funzione è molto utile in diverse applicazioni, ad esempio:

- studio delle soluzioni alternative della FBA;
- studio della distribuzione dei flussi sotto le condizioni di crescita sub-ottima;
- analisi della flessibilità e della ridondanza della rete.

Per ottenere tale risultato, la FVA risolve anch'essa dei problemi di ottimizzazione:

1. Ricordiamo innanzitutto il problema LP che viene risolto dalla FBA standard:

$$\begin{aligned} \max_v \quad & Z = f^T v \\ \text{subject to} \quad & Sv = 0; \\ & v^{\min} \leq v \leq v^{\max}. \end{aligned}$$

2. La FVA pone  $f = w$ , dove  $w$  rappresenta il vettore dei coefficienti della funzione obiettivo che si vuole analizzare e, dopo aver risolto il problema al punto 1. con  $f = w$ , risolve due problemi di ottimizzazione per ciascun flusso  $v_i$  di interesse:

$$\begin{aligned}
& \max_v / \min_v v_i \\
& \text{subject to } Sv = 0; \\
& w^T v \geq \gamma Z_0; \\
& v^{\min} \leq v \leq v^{\max}.
\end{aligned}$$

dove  $Z_0 = w^T v_0$  è una soluzione ottima del problema al punto 1) e  $\gamma$  è un parametro che determina se l'analisi è eseguita in condizioni subottime ( $0 \leq \gamma < 1$ ) o in condizioni di ottimalità ( $\gamma = 1$ ).

Se si assume che tutte le  $n$  reazioni della rete sono di interesse, la Flux Variability Analysis richiede la risoluzioni di  $2n$  problemi LP.

### 4.3 Sampling (metodo Monte Carlo per reti metaboliche)

Lo studio delle reti metaboliche tramite l'approccio constraint-based è molto utile per predire gli effetti fenotipici dei KnockOut e della manipolazione genetica. Però, data l'incertezza legata alle variabili e ai parametri in gioco dei modelli estesi, è spesso utile effettuare uno studio tramite campionamento dello spazio delle soluzioni plausibili col metodo Monte Carlo. Questa tecnica offre molti vantaggi, come la capacità di operare anche in presenza di dati mancanti, di applicare tecniche di post-processing, di quantificare l'incertezza e, di conseguenza, di ottimizzare gli esperimenti minimizzando tale incertezza.

L'argomento è molto ampio, ma noi ci soffermeremo sulla parte che ci riguarda, ovvero porremo attenzione a come tale metodo si possa utilizzare per analizzare lo spazio delle possibili soluzioni della FBA tramite campionamento.

Il Monte Carlo sampling viene utilizzato per generare un insieme di distribuzioni di flusso, che possano anche contenere dei loop. Il concetto di loop può essere critico nel momento in cui si vuole ottenere una distribuzione ottima di flussi, poichè la presenza di tali cicli interni può portare a soluzioni diverse che danno lo stesso valore ottimo finale, se i flussi partecipanti al loop non partecipano invece alla funzione da ottimizzare (un valore qualsiasi di tali flussi non influenza l'ottimo ottenuto). Il metodo si basa sull'algoritmo ACHR (Artificially Centered Hit and Run), descritto nella sottosezione 11.1, con alcune piccole modifiche. Inizialmente viene generato un set di punti non uniformi e pseudo-random (warm-up points). Tramite una serie di iterazioni, ognuno di questi punti viene fatto muovere in maniera random, sempre rimanendo all'interno dello spazio delle possibili soluzioni.

Le fasi dello spostamento di ciascun punto si possono così riassumere:

1. viene scelta una direzione randomicamente;
2. viene calcolato il limite massimo di quanto si può muovere il punto nella direzione selezionata (verso positivo e negativo);
3. viene scelto, in modo random, un punto lungo questa linea.

Dopo una molteplicità di iterazioni, l'insieme dei punti è smistato e può essere considerato come un campionamento uniforme dell'intero spazio delle soluzioni.

Per generare l'insieme dei warm-up points viene utilizzato un Linear Programming (LP). Per ogni punto, i coefficienti ottimi vengono posti in un vettore random contenente valori in  $[-1, 1]$ . Questa procedura genera un punto in angoli casuali nello spazio delle soluzioni.

La direzione viene scelta attraverso il metodo descritto da Smith RL. Kaufmann DE. ('Direction choice for accelerated convergence in hit-and-run sampling.' *Operations Research*, 46:84–95, 1998): viene calcolato il punto centrale tra tutti i punti e la direzione è data dalla distanza tra un punto scelto in modo casuale e tale punto centrale. Questa tecnica di scelta della direzione ha l'effetto di polarizzare le direzioni lungo le direzioni più lunghe dello spazio, velocizzando il rate di mescolamento, mantenendo un campionamento uniforme.

Uno dei lati negativi dell'algoritmo ACHR consiste nel fatto che non è ben definita una condizione di termine. Si introduce quindi il concetto di 'mixed fraction', definito come una misura del numero di iterazioni richieste per ottenere un mescolamento corretto. In pratica, viene creata una partizione sull'insieme dei punti, tracciando una linea lungo il valore mediano, tenendo metà dei punti da una parte e metà dall'altra. La 'mixed fraction' è definita come il conteggio di quanti punti attraversano questa linea mediana dall'inizio del campionamento alla fine, inteso come frazione del numero di punti totali nell'insieme. Inizialmente la 'mixed fraction' è posta ad 1, in quanto tutti i punti sono dalla stessa parte della linea. Quando si raggiunge un

mescolamento perfetto, ogni punto ha una probabilità del 50% di attraversare la linea di partizione, perciò la 'mixed fraction' sarà molto vicina a 0.5.

#### 4.3.0.1 Sampling di uno spazio di soluzioni privo di loop

È possibile implementare il campionamento dello spazio delle soluzioni privato dei loop, eseguendo un post-processing al metodo Monte Carlo analizzato appena sopra. Si può infatti eliminare i loop, dopo aver generato un insieme di distribuzioni di flussi. Uno dei diversi modi che si può utilizzare a questo scopo consiste nel trovare il flusso più vicino ( $w_i$ ) ad una distribuzione di flusso ( $v_i$ ) trovata precedentemente, che non produca cicli:

$$\begin{aligned}
 & \min_w |w_j - v_i| \\
 \text{soggetto ai vincoli:} \\
 & \sum_k S_{kj} \cdot w_k = 0 \\
 & lb_j \leq w \leq ul_j \\
 & -1000(1-a_i) \leq w_i \leq 1000a_i \\
 & -1000(a_i) + 1(1-a_i) \leq G_i \leq -1(a_i) + 1000(1-a_i) \\
 & N_{int}G = 0 \\
 & a_i \in \{0, 1\} \quad G_i \in \mathbb{R}
 \end{aligned}$$

Se la norma da minimizzare viene considerata come norma Euclidea (norma 2), questo problema diventa un problema MIQP (si minimizza  $(w_j - v_i)^2$ ). Invece, se si considera la norma 1, il problema può essere implementato come un problema MILP, introducendo le due variabili ausiliari  $v^+$  e  $v^-$ , tali che  $|w_j - v_i| = (v^+ + v^-)$ :

$$\begin{aligned}
 & \min_w (v^+ + v^-) \\
 \text{soggetto ai vincoli:} \\
 & v_j^+ \geq v_j - w_j \\
 & v_j^- \geq w_j - v_j \\
 & \sum_k S_{kj} \cdot v_k = 0 \\
 & lb_j \leq v_j \leq ul_j \\
 & -1000(1-a_i) \leq v_i \leq 1000a_i \\
 & -1000(a_i) + 1(1-a_i) \leq G_i \leq -1(a_i) + 1000(1-a_i) \\
 & N_{int}G = 0 \\
 & a_i = \{0, 1\}, \quad G_i \in \mathbb{R}, \quad v_j^+, v_j^- > 0.
 \end{aligned}$$

In Figura 4.7 [5] possiamo osservare un esempio di Monte Carlo sampling per lo studio del modello iT431.

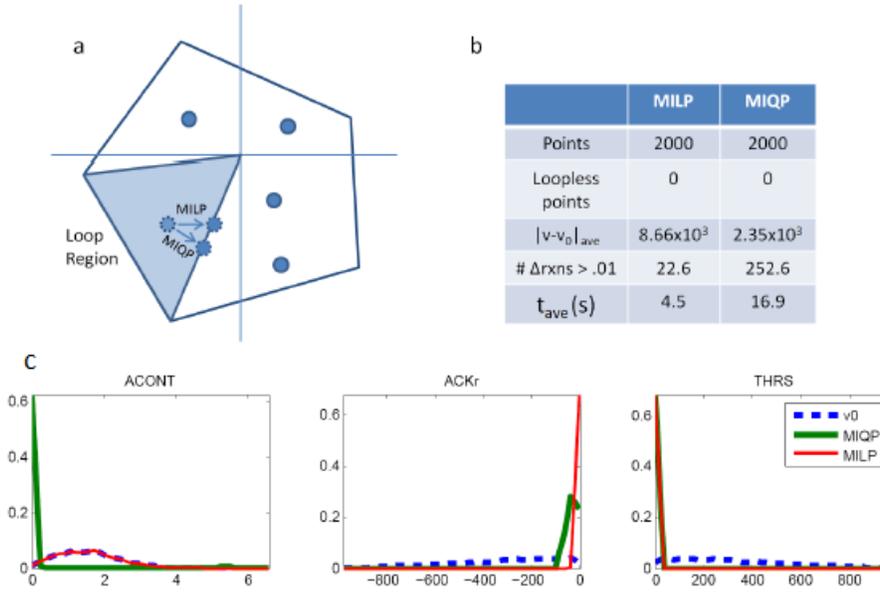


Figure 4.7: [5] **Esempio di Monte Carlo sampling.** a) I punti vengono campionati inizialmente dall'intero spazio delle soluzioni in stato stazionario. I punti che si trovano in una regione non fisicamente possibile (in blu) vengono spostati nel punto più vicino della regione valida (usando la distanza in norma 1 o in norma 2). b) Differenze tra i due campionamenti riferiti alle diverse norme utilizzate. c) Istogrammi raffiguranti le distribuzioni attraverso tre particolari reazioni, nei tre casi: punti iniziali ( $v_0$ ), punti privati dei loop nei due metodi (MIQP e MILP).

### 4.3.1 Algoritmo ACHR

Per descrivere l'algoritmo di campionamento ACHR, focalizziamo la nostra attenzione sullo spazio del flusso ( $v$ ) che descrive l'insieme delle possibili distribuzioni di flusso. Tale spazio è cos' definito:

$$S \cdot v = 0$$

$$lb \leq v \leq ub.$$

Una delle più importanti proprietà di tale spazio, che viene sfruttata dall'algoritmo ACHR, è la sua convessità. (La convessità è definita come la proprietà per cui per ciascuna coppia di punti nello spazio,  $x$  e  $y$ , il segmento lineare che li unisce ( $c \cdot x + (1 - c) \cdot y$  per  $c \in \{0, 1\}$ ) è interamente contenuto nello spazio stesso).

Per ottenere un campionamento uniforme è necessario applicare una tecnica basata sull'eliminazione di campioni. Nonostante esistano più metodi per ottenere tale obiettivo, noi ci soffermeremo solo su quello implementato nel COBRA Toolbox, ovvero la tecnica chiamata 'Hit and Run Sampling'.

#### 4.3.1.1 Hit and run Sampling

La procedura seguita da tale metodo, si può riassumere in 4 fasi:

1. viene scelto un punto qualsiasi  $x_0$  all'interno dello spazio;
2. ad ogni iterazione viene scelta una direzione  $c$  in modo randomico;
3. vengono calcolati i limiti di una linea che passa per  $x_0$  ed è posta nella direzione  $c$ : vengono quindi calcolati i limiti di  $\alpha$  ( $\alpha_{min}$ ,  $\alpha_{max}$ ) tali che  $x_{k+1} = x_k + \alpha c$  si trovi ancora all'interno dello spazio;
4. viene scelto in modo casuale un  $\alpha$  tra  $\alpha_{min}$  e  $\alpha_{max}$  e si pone  $x_{k+1} = x_k + \alpha c$ .

Tramite questa procedura viene generata una serie di punti  $x_0, x_1, \dots, x_k$ , che costituiscono un campionamento uniforme dello spazio al tendere di  $k$  a  $+\infty$ .

Il problema principale di questa tecnica viene chiamato 'poor mixing' ed è determinato dal fatto che ciascun punto dipende da quello precedente: il numero di passi richiesti per ottenere un mescolamento tale per cui i punti  $x_k$  e  $x_{k+n}$  siano indipendenti può essere molto elevato. Tale problema si nota soprattutto se una delle dimensioni dello spazio è molto più grande delle altre (situazione che tra l'altro si verifica molto spesso nei modelli biologici).

Fortunatamente è stato implementato un miglioramento dell'Hit and Run Sampler, per risolvere tale problema. L'idea di fondo di questa tecnica sta nel modo di scegliere la direzione in modo totalmente casuale, bensì in modo tale da massimizzare la dimensione del salto e da ottenere più rapidamente il mescolamento.

Tale metodo (suggerito da Kaufmann et al) usa i punti  $x_0, x_1, \dots, x_k$  per scegliere la direzione:

1. calcola il baricentro di tali punti:  $\bar{x} = \sum_{i=1}^k x_i$ ;
2. sceglie  $c = x_i - \bar{x}$ , per un  $i$  scelto in modo casuale.

La direzione viene quindi scelta come la differenza tra uno dei punti già visitati (scelto randomicamente) e il punto centrale. Tale scelta si basa sull'assunzione che i punti già visitati abbiano già la struttura dello spazio e quindi uno di tali punti scelto casualmente tenderà a distare dal centro precisamente nella dimensione lungo la quale lo spazio è più esteso.

Per poter inizializzare l'algoritmo occorre quindi un numero minimo di punti da utilizzare per calcolare  $\bar{x}$ : inizialmente viene generato perciò un numero minimo di punti (detti 'warmup' points), solitamente generati tramite un problema LP, necessari per avviare la procedura descritta.

## 4.4 Analisi del modello (lunghezza degli extreme pathways e Reaction Participation)

Per poter caratterizzare completamente la condizione di stato stazionario di una rete metabolica si può fare ricorso al concetto di extreme pathways, che, come avremo modo di vedere, è direttamente collegato ad ulteriori nuovi concetti, quali la Reaction Participation e la correlazione tra le reazioni del modello.

### 4.4.1 Extreme pathways

Le capacità fenotipiche di una rete metabolica possono essere caratterizzate da un insieme di pathway unici e indipendenti (Schilling et al. 2000).

Gli extreme pathways corrispondono alle distribuzioni di flusso attraverso le reazioni della rete metabolica in condizione di stato stazionario. Perciò essi non descrivono solamente un set lineare di reazioni che legano il substrato ai prodotti, bensì caratterizzano i flussi relativi attraverso tutte le reazioni necessarie per convertire i substrati in prodotti, per bilanciare tutti i gruppi di cofattori e per secernere ogni prodotto secondario necessario per mantenere la rete in uno stato omeostatico (cioè in una condizione di stabilità).

Il nome di tali pathway (extreme) è legato al fatto che costituiscono i bordi dello spazio delle soluzioni e quindi caratterizzano le funzioni estremali della rete. La Figura 4.8 [7] ci aiuta ad immaginare questi extreme pathways come i generatori di un cono convesso in uno spazio pluridimensionale, che circoscrive tutti i possibili fenotipi ottenibili in stato stazionario.

Tutte le potenziali distribuzioni di flusso (ossia tutti i possibili fenotipi metabolici) in stato stazionario sono delle combinazioni lineari a coefficienti non negativi degli extreme pathways. In pratica quindi gli extreme pathway caratterizzano i lower e gli upper bound della conversione di qualsiasi substrato in qualsiasi prodotto.

#### 4.4.1.1 Lunghezza del pathway e matrice della Reaction Participation

Per costruire una matrice relativa agli extreme pathways (matrice  $P$ ), si crea una matrice in cui ogni colonna rappresenta un extreme pathway e ogni riga una reazione della rete. Il valore dell'elemento in posizione  $ij$  corrisponde al livello di flusso relativo attraverso l' $i$ -esima reazione nel  $j$ -esimo pathway.

Per costruire tale matrice si utilizzano tutti gli extreme pathways: in Figura 4.9 [7] possiamo osservare un esempio di tale matrice relativo ad una rete contenente tre extreme pathways, che in generale possono

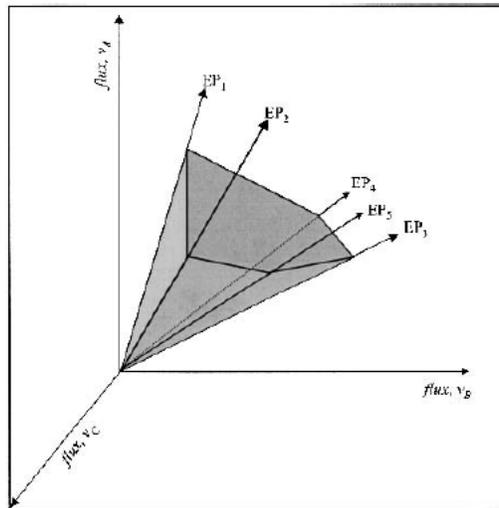


Figure 4.8: [7] Rappresentazione schematica di un cono convesso caratterizzato da 5 extreme pathways ( $EP_1$ ,  $EP_2$ ,  $EP_3$ ,  $EP_4$  ed  $EP_5$ ), che circoscrivono lo spazio delle soluzioni per tre flussi ( $v_A$ ,  $v_B$  e  $v_C$ ). Tutti i punti all'interno del cono possono essere descritti da una combinazione lineare a coefficienti non negativi degli extreme pathways.

avere un numero qualsiasi di input e di output. Nell'esempio possiamo osservare che i pathway  $EP_1$  ed  $EP_2$  non sono delle semplici reazioni lineari, bensì contengono due output (E e il prodotto secondario).

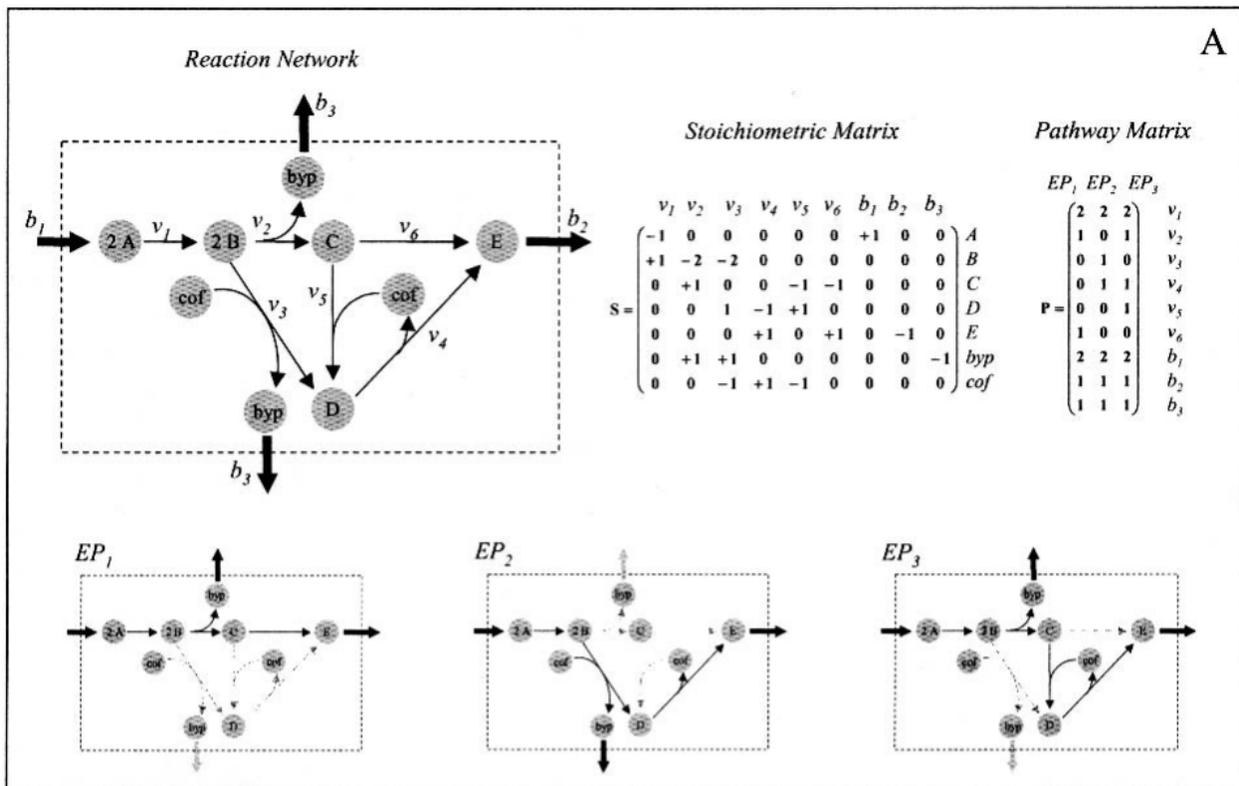


Figure 4.9: [7] Esempio di un sistema di cui vengono rappresentate la matrice stechiometrica e la matrice degli extreme pathways, rappresentati nella parte inferiore.

La lunghezza di un extreme pathway si definisce come il numero di reazioni che vi partecipano.

La matrice lunghezza ( $P_{LM}$ ) si può calcolare direttamente dalla matrice stessa  $P$ , come possiamo osservare dalla Figura 4.10.

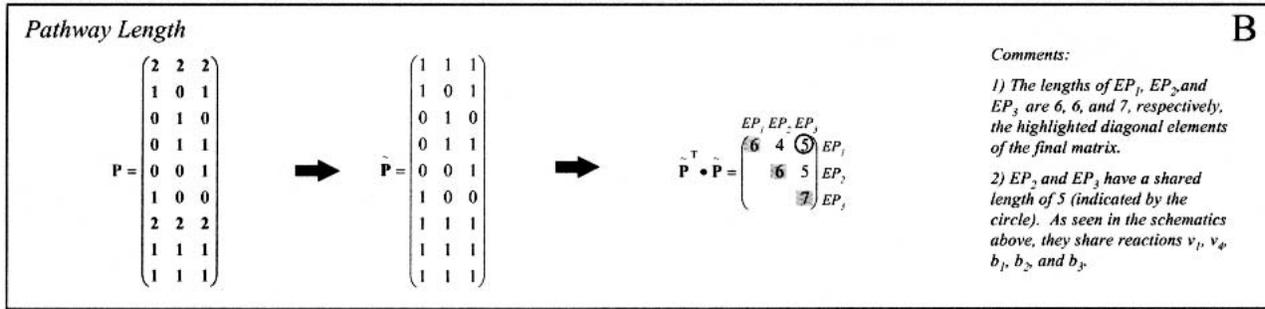


Figure 4.10: Esempio delle lunghezze degli extreme pathways.

Come prima cosa occorre scrivere la matrice  $P$  in forma binaria ( $\tilde{P}$ ), nella cui posizione  $ij$  compare un 1 se la reazione  $i$  viene utilizzata all'interno dell'extreme pathway  $j$  e uno 0 altrimenti. A questo punto la matrice lunghezza  $P_{LM}$  (simmetrica) si può ottenere calcolando

$$P_{LM} = \tilde{P}^T \cdot \tilde{P}.$$

I termini diagonali di  $P_{LM}$  indicano la lunghezza di ciascuno degli extreme pathway. Nell'esempio riportato in Figura 37, il primo termine diagonale vale 6: significa che al pathway  $EP_1$  partecipano 6 reazioni. I termini al di fuori della diagonale indicano il numero di reazioni che ciascuna coppia di pathway ha in comune. Nell'esempio analizzato possiamo osservare che i pathway  $EP_3$  (terza colonna) ed  $EP_1$  (prima riga) hanno 5 reazioni in comune. Dalla Figura 4.9 possiamo verificare che tali reazioni sono  $b_1$ ,  $v_1$ ,  $v_2$ ,  $b_2$  e  $b_3$ .

Dalla matrice binaria  $\tilde{P}$  si può ricavare anche la matrice delle Reaction Participation ( $R_{PM}$ ), anch'essa simmetrica:

$$R_{PM} = \tilde{P} \cdot \tilde{P}^T.$$

La Reaction Participation è definita come la percentuale di extreme pathways che utilizza una data reazione.

La matrice  $R_{PM}$  del nostro esempio può essere visualizzata in Figura 4.11.

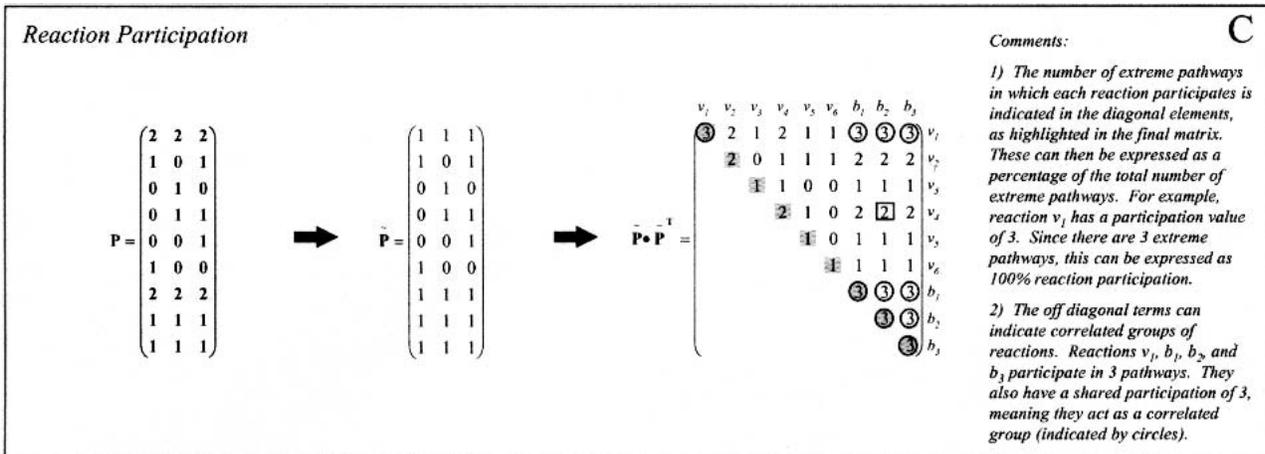


Figure 4.11: Esempio di matrice delle Reaction Participation e proprietà ad essa relative.

I termini diagonali di  $R_{PM}$  si riferiscono al numero di pathway a cui partecipa la reazione corrispondente. Nell'esempio in Figura 4.11, il primo termine diagonale, corrispondente alla reazione  $v_1$ , assume il valore 3, quindi tale reazione partecipa a tutti e tre gli extreme pathways della rete (si può verificare osservando la Figura 4.9). Tali valori possono essere considerati come percentuali del numero totale degli extreme pathways: per fare questo si normalizza la matrice  $R_{PM}$  relativamente al numero totale dei pathway (nell'esempio della

Figura 4.11 i pathway sono 3). Il primo termine diagonale dell'esempio sarebbe quindi corrispondente ad un 100% di reaction participation, poichè  $v_1$  partecipa a tutti e 3 i pathway.

I termini fuori diagonale indicano il numero di extreme pathways che contengono entrambe le reazioni corrispondenti. Nell'esempio in Figura 4.11 le reazioni  $b_2$  (ottava colonna) e  $v_4$  (quarta riga) sono contenute entrambe in 2 (67%) pathways (osservando la Figura 4.9 possiamo verificare che si tratta dei pathways  $EP_2$  ed  $EP_3$ ).

Gli elementi cerchiati nella matrice  $R_{PM}$  rappresentata in Figura 4.11 mostrano coppie di reazioni che partecipano esattamente agli stessi extreme pathways. In questo esempio particolare, ciascuna di queste coppie partecipa alla totalità dei pathway, ciò significa che  $v_1$ ,  $b_1$ ,  $b_2$  e  $b_3$  sono sempre presenti. Tali reazioni formano quindi un gruppo di reazioni tali che se una di queste viene utilizzata, allora necessariamente anche tutte le altre devono essere attive.

#### 4.4.1.2 Correlazione tra i valori delle Reaction Participation e definizione di Sottoinsiemi di Reazioni

Le reazioni che possono essere utilizzate in un particolare extreme pathway non sono indipendenti. Come analizzato precedentemente, gli elementi non diagonali della matrice  $R_{PM}$  della Reaction Participation possono essere utilizzati per determinare quali sottoinsiemi di reazioni appaiono assieme in tutti gli extreme pathways. Questi sottoinsiemi di reazioni forniscono quindi dei sottoinsiemi di enzimi ad esse collegati che devono essere co-regolati.

## Altri strumenti di analisi utili nello studio delle reti metaboliche

Prima di analizzare quali operazioni a livello genetico si possono effettuare in un modello metabolico, vorremmo soffermarci su altri due strumenti di analisi, che possono rivelarsi utili nella costruzione di un modello (Gap filling) e nell'analisi della dipendenza del valore ottimo ottenuto con la FBA da particolari reazioni presenti nella rete (Robustness Analysis).

## 4.5 Gap filling

A causa di un'incompleta conoscenza delle reti metaboliche, molto spesso i modelli contengono dei gap, ovvero lacune dovute a informazioni biochimiche mancanti, che possono far sì che il risultato ottenuto in silico sia differente dai dati sperimentali. Diventa quindi importante cercare di individuare tali lacune e successivamente usare degli algoritmi di ottimizzazione per colmarle.

### 4.5.1 Ricerca dei gap in un modello metabolico

In una rete metabolica i gap si manifestano in due forme:

1. metaboliti che non possono essere prodotti da nessuna reazione della rete né importati da alcuna reazione di scambio;
2. metaboliti che non vengono consumati da nessuna reazione della rete, né esportati da alcun pathway di secrezione.

In Figura 4.12 [11] possiamo osservare questi due casi in un esempio. Il metabolita A viene definito come 'root no-production metabolite' poichè non vi è alcuna produzione né meccanismo di trasporto che lo possa introdurre nella rete. Il metabolita C è definito come 'downstream no-production metabolite', in quanto esiste una reazione che lo produrrebbe, ma essa non può trasportare alcun flusso a causa della mancanza del metabolita A. In modo analogo, i metaboliti B e D vengono ripetutamente definiti come 'root no-consumption metabolite' e 'downstream no-consumption metabolite'.

#### 4.5.1.1 Algoritmo di identificazione dei no-production metabolites

Inizialmente analizziamo l'algoritmo finalizzato ad identificare i metaboliti 'root no-production' sotto le condizioni di stato stazionario: in stato stazionario, se sono presenti  $M$  metaboliti in  $N$  reazioni, la condizione

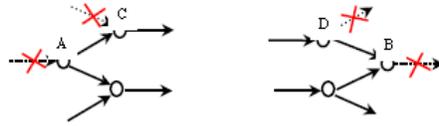


Figure 4.12: [11] *Esempi di gap metabolici.*

da imporre è la seguente:

$$\sum_{j \in M} S_{ij} v_j = b_i, \quad \forall i = 1, \dots, N \quad (1)$$

dove  $b_i$  è un parametro che indica se il metabolita  $i$  viene assorbito ( $b_i < 0$ ) o secreto ( $b_i > 0$ );  $v_j$  è il flusso attraverso la reazione  $j$  ed  $S_{ij}$  è il corrispondente indice stechiometrico.

Un metabolita  $i$  è definito 'no-production metabolite' se, analizzando tutta la  $i$ -esima colonna della matrice  $S$  non si trova alcun elemento positivo (per le reazioni irreversibili) o diverso da zero (per quelle reversibili).

Notiamo inoltre che per identificare i metaboliti 'root no-consumption metabolites' è sufficiente utilizzare una procedura simmetrica a quella appena descritta.

Per quanto riguarda l'identificazione dei metaboliti 'downstream no-production', se siamo nel caso di modelli a singolo compartimento (supponiamo per semplicità di essere in un modello coinvolto solo nel citosol), non ci si può limitare ad analizzare la matrice stechiometrica  $S$ . Occorre invece utilizzare una procedura di ottimizzazione per identificarli e localizzarli con precisione. Si assume che per tutti i metaboliti nel citosol sia assicurato un termine di consumo: il vincolo precedente (1) viene quindi riscritto come:

$$\sum_{j \in M} S_{ij} v_j \geq 0, \quad \forall i = 1, \dots, N \quad (2)$$

A questo punto, per identificare il sottoinsieme di  $N$  contenente i 'no-production metabolites', che chiameremo insieme  $NP$ , è necessario introdurre due variabili binarie  $x_i^{np}$  e  $w_{ij}$ , definite come:

$$x_i^{np} = \begin{cases} 1 & \text{se il metabolita } i \text{ può essere prodotto,} \\ 0 & \text{altrimenti} \end{cases}$$

$$w_{ij} = \begin{cases} 1 & \text{se la reazione } j \text{ che produce } i \text{ è attiva} \\ 0 & \text{altrimenti} \end{cases}$$

L'ottimizzazione da eseguire per identificare tutte i downstream no-production metabolites in aggiunta ai root no-production metabolites si rivela essere la seguente:

$$\begin{aligned} \max \sum_i (x_i^{np}) & & (\text{GapFind}) & & (3) \\ \text{subject to:} & & & & \\ S_{ij} v_j \geq \varepsilon w_{ij} & & \forall i \in N, j | (S_{ij} > 0 \ \& \ j \in IR) & & (4) \\ S_{ij} v_j \leq M w_{ij} & & \forall i \in N, j | (S_{ij} > 0 \ \& \ j \in IR) & & (5) \\ S_{ij} v_j \geq \varepsilon - M(1 - w_{ij}) & & \forall i \in N, j | (S_{ij} \neq 0 \ \& \ j \in R) & & (6) \\ S_{ij} v_j \geq M w_{ij} & & \forall i \in N, j | (S_{ij} \neq 0 \ \& \ j \in R) & & (7) \\ \sum_{j'} w_{ij} \geq x_i^{np} & & \forall i \in N & & (8) \\ lb_j \leq v_j \leq ub_j & & \forall j \in Model & & (9) \\ \sum_j S_{ij} v_j \geq 0 & & \forall i \in N & & (10) \\ x_i^{np} \in \{0, 1\} & & \forall i & & \\ w_{ij} \in \{0, 1\} & & \forall i, j & & \\ \{j' \in M | (S_{ij} > 0 \ \& \ j \in IR) \text{ or } (S_{ij} \neq 0 \ \& \ j \in R)\} & & & & \end{aligned}$$

Dove:

**R e IR:** stanno per reazione reversibile e irreversibile, rispettivamente;

(4) e (5): assicurano che per ogni reazione irreversibile che produce il metabolita  $i$ , la variabile binaria  $w_{ij}$  assuma il valore 1 solamente se la reazione produce almeno  $\varepsilon$  unità di metabolita  $i$  (cioè  $S_{ij} v_j \geq 0$ );

- (6) e (7): assicurano che per ogni reazione reversibile a cui partecipa il metabolita  $i$ ,  $w_{ij}$  valga 1 solamente se la reazione produce almeno  $\varepsilon$  unità di metabolita  $i$ ;
- (8): assicura che il metabolita  $i$  debba avere almeno una via produttiva se  $x_i^{np} = 1$ ;
- (9): assicura che siano rispettati i vincoli di lower ed upper bound dei flussi.
- (10) : incorpora il vincolo (2).

Per quanto riguarda la relazione (3), osserviamo che si vuole massimizzare la somma delle variabili binarie  $x_i^{np}$  su tutti i metaboliti, assicurando l'identificazione di tutti i metaboliti che abbiano almeno una via di produzione. Perciò, se nella soluzione ottima  $x_i^{np}$  è uguale a 0, significa che il metabolita  $i$  non può essere prodotto e quindi deve appartenere all'insieme  $NP$ . Poichè questa metodologia rintraccia anche tutti i root no-production metabolites, già identificati precedentemente, questi vengono rimossi da  $NP$ , per poter distinguere i due tipi di gap.

Nel caso in cui siano presenti più compartimenti (la Figura 4.13 [11] ne riporta un esempio, in cui solo i metaboliti del citosol possono essere mandati all'esterno nello spazio extra-cellulare, mentre i metaboliti degli altri compartimenti possono essere scambiati solo con il citosol. Ciò implica che per questi ultimi metaboliti le quantità prodotta e consumata devono coincidere), il vincolo (10) deve essere sostituito con due vincoli differenti:

$$\sum_j S_{ij}v_j \geq 0, \quad \forall i \in \text{citosol} \quad (11)$$

$$\sum_j S_{ij}v_j = 0, \quad \forall i \notin \text{citosol} \quad (12)$$

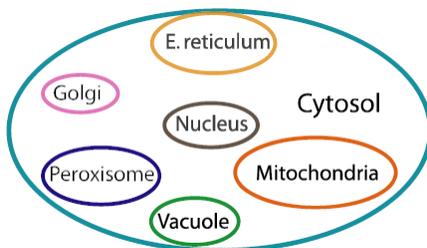


Figure 4.13: [11] *Esempio di modello multi-compartimentale (Saccharomyces cerevisiae), in cui tutti i compartimenti interagiscono solo con il citosol, che è l'unico ad interagire con l'esterno.*

Come già osservato precedentemente, la procedura è analoga (simmetrica) nel caso dei gap riguardanti i metaboliti che non vengono consumati nella rete.

## 4.5.2 Ricerca dei candidati per riempire i gap

Una volta individuati i gap presenti nel modello metabolico, esistono dei metodi per cercare le reazioni candidate a colmare tali lacune.

In Figura 4.14 [10] possiamo visualizzare in modo schematico la procedura che viene utilizzata a tale scopo relativa ad un esempio relativo modello del metabolismo di Escherichia coli K-12 MG1655.

In questa sede vogliamo focalizzare l'attenzione sull'algoritmo utilizzato per individuare le reazioni che è necessario aggiungere alla rete per eliminare i gap (box rosso della Figura 41). L'algoritmo che verrà descritto qui di seguito è un ibrido tra due algoritmi di metabolic engineering sviluppati precedentemente, con qualche aggiunta. Gli algoritmi appena citati sono i seguenti:

1. il Bugard et al. algorithm (Bugard AP, Vaidyaraman S, Maranas CD (2001) Biotechnol Prog 17:791-797);
2. il Pharkya et al. algorithm (Pharkya P, Burgard AP, Maranas CD (2004) Genome Res 14: 2367-2376).

L'algoritmo infatti utilizza alcuni vincoli matematici utilizzati da uno o dall'altro. Il Bugard et al. algorithm calcola il numero minimo di reazioni all'interno di una rete che sono necessarie per raggiungere le condizioni di crescita ottime e subottime (con condizione ottima di crescita si intende il valore massimo di biomassa raggiungibile). Il Pharkya et al. algorithm invece identifica inizialmente le reazioni enzimatiche che necessitano di essere aggiunte alla rete per produrre un particolare metabolita di interesse. Lo scopo di tale algoritmo

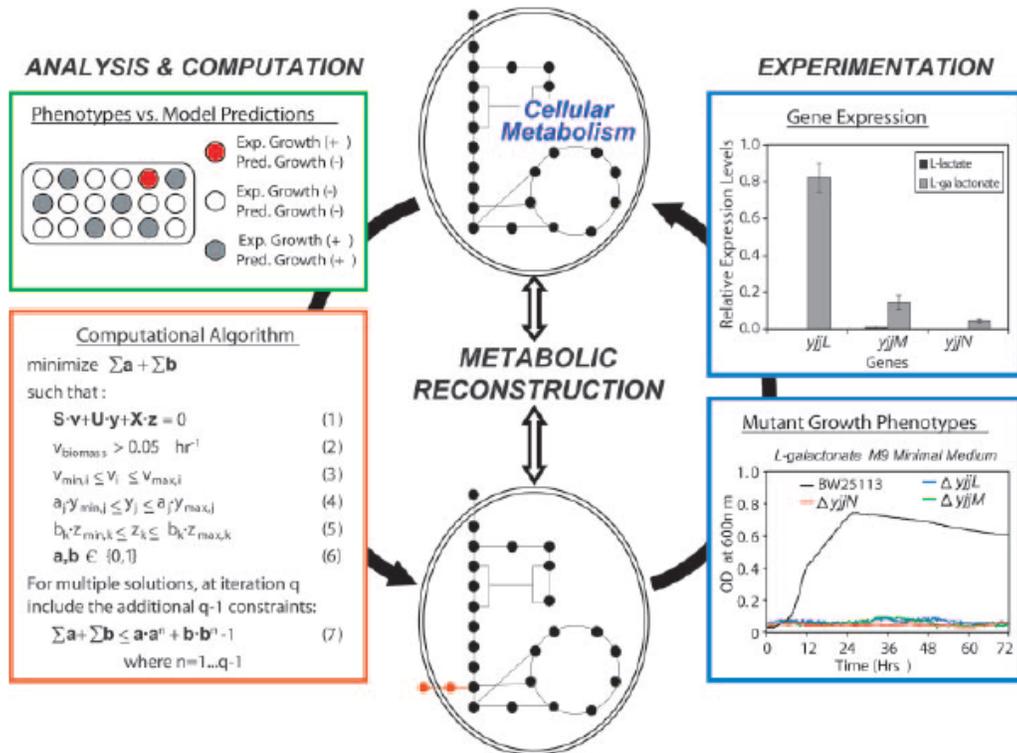


Figure 4.14: [10] Esempio di procedura per identificare le reazioni mancanti. Avendo a disposizione la ricostruzione di un modello metabolico, si possono confrontare le previsioni di crescita con i fenotipi osservati sperimentalmente (box verde). Laddove si osserva sperimentalmente una crescita non prevista dal modello, tramite un algoritmo di ottimizzazione (box rosso) si possono identificare le reazioni mancanti scegliendole da un database universale. Nei box blu possiamo osservare come si possano utilizzare dei tet sperimentali relativi a mutazioni genetiche per identificare i geni responsabili dell'aggiunta di particolari attività enzimatiche, analizzando conseguentemente in quali reazioni tali geni sono coinvolti.

è quello di identificare quali proteine sia necessario aggiungere e quali rimuovere in modo da ottenere la massimizzazione della biomassa ma anche della secrezione di un particolare metabolita. Anzichè selezionare le reazioni da aggiungere per produrre un metabolita, l'algoritmo descritto in seguito seleziona le reazioni che occorre aggiungere per produrre le varie componenti della funzione biomassa, ossia tutti i metaboliti che costituiscono la cellula, come ad esempio gli amminoacidi e i nucleotidi). Inoltre vengono aggiunte alcune reazioni di scambio che si ipotizza debbano essere introdotte nella rete, per permettere la secrezione di prodotti differenti.

#### 4.5.2.1 Algoritmo per il riempimento dei Gap

Per ciascuna condizione di crescita, l'algoritmo calcola il numero minimo di reazioni che necessitano di essere introdotte dalle matrici  $U$  (matrice delle reazioni metaboliche note) e  $X$  (matrice delle reazioni di scambio note) per permettere la crescita. I vincoli e l'obiettivo dell'ottimizzazione sono molto simili a quelli utilizzati nei metodi di metabolic engineering precedentemente nominati (1 e 2). Tale algoritmo si può riassumere nel modo seguente:

$$\min \left( \sum_j a_j + \sum_k b_k \right)$$

subject to :

$$Sv + Uy + Xz = 0 \quad (1)$$

$$v_{biomass} > 0.05 \quad (2)$$

$$v_{min,j} \leq v_j \leq v_{max,j} \quad (3)$$

$$a_j y_{min,j} \leq y_j \leq a_j y_{max,j} \quad (4)$$

$$b_k z_{min,k} \leq z_k \leq b_k z_{max,k} \quad (5)$$

$$a, b \in \{0, 1\} \quad (6)$$

$$Sa + Sb = a^* a^n + b^* b^n \quad n = 1, \dots, q-1 \quad (7)$$

Prima di analizzare il significato dei vincoli, è meglio chiarirsi le idee sulle matrici e le variabili in gioco:

$S$ : matrice stechiometrica;

$U$ : matrice contenente gli indici stechiometrici di tutte le reazioni contenute nel database universale, che include solo le reazioni bilanciate per tutti gli elementi, eccetto H, del database di Kegg;

$X$ : matrice stechiometrica inerente alle reazioni di scambio, che permettono quindi a ciascun metabolita di essere aggiunto o rimosso dal sistema. Tale matrice è diagonale con valori sulla diagonali pari a  $-1$ ;

$a$  e  $b$ : vettori binari, che contengono valori non nulli solo in corrispondenza alle reazioni che devono essere aggiunte dalle matrici  $U$  ed  $X$ , per permettere la crescita della cellula;

$v$ ,  $y$  e  $z$ : vettori dei flussi attraverso le reazioni del modello, del database universale e delle reazioni di scambio, rispettivamente.

Possiamo ora analizzare il significato dei vincoli sopra elencati più nel dettaglio:

(1): vincolo del bilanciamento di massa in stato stazionario;

(2): obbliga un valore minimo del rate di crescita;

(3) ÷ (5): assicurano che vengano rispettati i limiti di lower ed upper bound per ciascun flusso;

(7): vincolo che permette di calcolare soluzioni multiple: si implementa un approccio iterativo, in cui ad ogni iterazione vengono aggiunti dei vincoli basati sulle soluzioni precedenti.  $a^n$  e  $b^n$  sono i vettori ottenuti dalle precedenti iterazioni.

Se non si utilizza quest'ultimo vincolo (7), i sei precedenti sono sufficienti per calcolare una sola soluzione.

## 4.6 Robustness Analysis

Nell'ambito delle reti metaboliche, la robustezza è definita come una misura del cambiamento del flusso massimo ottenuto dalla massimizzazione effettuata tramite FBA, al variare del flusso attraverso una o più particolari reazioni della rete. In pratica, con la Robustness Analysis si vuole valutare quanto il flusso ottimo è dipendente da determinate reazioni della rete.

La caratteristica di robustezza di una rete metabolica viene determinata calcolando inizialmente la distribuzione ottima dei flussi ottenuta massimizzando il flusso obiettivo (FBA). Chiamiamo questo vettore di flussi ottenuto 'wildtype' flux.

In un secondo momento il vettore dei flussi attraverso le reazioni che si vogliono analizzare viene fatto variare dal valore minimo (lower bound) al valore massimo (upper bound) che può assumere ciascuna delle reazioni in esame. In relazione ad ogni modifica di flusso, viene calcolato il valore del flusso attraverso la reazione da ottimizzare.

In questo modo è possibile verificare quanto è sensibile il flusso obiettivo alle variazioni di determinate reazioni della rete metabolica.

## 4.7 Metabolic engineering

Quando si ha a disposizione un modello di una rete metabolica, completa delle informazioni geniche relative a ciascuna reazione della rete, si possono eseguire delle analisi che permettono di studiare delle strategie di silenziamento di opportune proteine, al fine di ottimizzare anche determinati prodotti oltre alla funzione obiettivo definita. Queste tecniche rientrano sotto il nome di 'Metabolic engineering'.

In questa sede faremo riferimento a tre tecniche specifiche:

1. OptKnock;
2. OptGene;
3. Genetic design local search (GDLS).

### 4.7.1 OptKnock

OptKnock è una tecnica che suggerisce quali proteine silenziare (e quindi le reazioni ad essi connesse) al fine di effettuare una doppia ottimizzazione: da un lato viene ottimizzato il flusso posto come obiettivo dal modello (ad esempio la biomassa) e dall'altro si ottimizza la sovrapproduzione di specifici metaboliti prodotti all'interno della rete. Tale algoritmo risolve quindi una coppia di ottimizzazioni annidate tra loro e restituisce un insieme di geni da porre in KnockOut (KO).

Analizziamo ora nel dettaglio come avviene tale ottimizzazione.

Per massimizzare il flusso obiettivo ( $v_{cellular\ objective}$ ), mantenendo livelli di ATP e di glucosio fisicamente sostenibili, dobbiamo risolvere il problema LP:

$$\begin{array}{ll}
 \max v_{cellular\ objective} & (Primal) \\
 \text{subject to :} & \sum_{j=1}^M S_{ij}v_j = 0; \quad \forall i \in N \\
 & v_{pts} + v_{glk} = v_{glc\ uptake} \\
 & v_{atp} > v_{atp\ main} \\
 & v_{biomass} > v_{biomass}^{target} \\
 & v_j \geq 0 \quad \forall j \in M_{irrev} \\
 & v_j \leq 0 \quad \forall j \in M_{secre\ only} \\
 & v_j \in \mathbb{R} \quad \forall j \in M_{rev}
 \end{array}$$

dove  $M$  è il numero di reazioni nella rete,  $N$  è il numero di metaboliti,  $S_{i,j}$  è il coefficiente del metabolita  $i$  nella reazione  $j$ ,  $v_{biomass}$  è la combinazione di reazioni che definisce la biomassa,  $v_{atp\ main}$  è il minimo valore di ATP richiesto (non associato alla crescita),  $v_{glc\ uptake}$  è il rate di assorbimento di glucosio (fissato) e include sia i flussi assorbiti dal sistema di fosfotransferasi<sup>3</sup> ( $v_{pts}$ ), sia dalla glucochinasi<sup>4</sup> ( $v_{glk}$ ). Infine il set

<sup>3</sup>Le **fosfotransferasi** sono una categoria di enzimi che catalizzano il trasferimento di gruppi fosfato da una molecola ad un'altra.

<sup>4</sup>La **glucochinasi** è un enzima, appartenente alla categoria delle transferasi, presente nel pancreas, nel fegato, nell'intestino e nel cervello, che catalizza la seguente reazione:  $ATP + D - glucosio = ADP + D - glucosio 6 - fosfato$ .

delle reazioni è suddiviso in reazioni reversibili ( $M_{rev}$ ) e irreversibili ( $M_{irrev}$ ), mentre i metaboliti che sono solamente secreti dalla rete sono inclusi nell'insieme  $M_{secre}$ .

Per incorporare il problema dell'eliminazione di determinate reazioni, consideriamo una variabile  $y_j$  a valori binari, che vale 1 se la reazione corrispondente è attiva e 0 se non lo è. Ciascuna reazione attiva ha un upper bound ( $v_j^{max}$ ) e un lower bound ( $v_j^{min}$ ), ottenuti massimizzando e minimizzando ciascun flusso, soggetto ai vincoli dettati nel problema di massimizzazione dell'obiettivo cellulare (*Primal*).

Per identificare quali reazioni vanno bloccate ( $y_j = 0$ ) affinché la secrezione di un particolare componente biochimico ( $v_{biochemical}$ ) sia massimizzata durante l'ottimizzazione della biomassa, viene risolto un problema che vede annidati assieme due problemi di ottimizzazione:

$$\begin{array}{l}
 \max_{y_j} v_{biochemical} \\
 \text{subject to} \\
 \left( \begin{array}{l}
 \max v_{biomass} \\
 \text{subject to :} \\
 \sum_{j=1}^M S_{ij} v_j = 0; \\
 v_{pts} + v_{glk} = v_{glc \text{ uptake}} \\
 v_{atp} > v_{atp \text{ main}} \\
 v_{biomass} > v_{biomass \text{ target}} \\
 v_j^{min} y_i \leq v_j \leq v_j^{max} y_i \quad \forall i \in M
 \end{array} \right) \quad \begin{array}{l}
 (OptKnock) \\
 (Primal) \\
 \forall i \in N \\
 \forall i \in M
 \end{array} \\
 \sum_{j \in M} (1 - y_j) \leq K \\
 y_j \in \{0, 1\} \quad \forall j \in M
 \end{array}$$

dove  $K$  è il massimo numero di reazioni che possono essere eliminate. Il problema di massimizzazione dell'obiettivo cellulare (*Primal*) contiene dei vincoli ulteriori che garantiscono il raggiungimento di un livello minimo di biomassa e che i flussi attraverso le reazioni proposte per l'eliminazione ( $y_j = 0$ ) siano settati a zero.

In Figura 4.15 [15] possiamo visualizzare la struttura dell'ottimizzazione bilevel che caratterizza OptKnock.

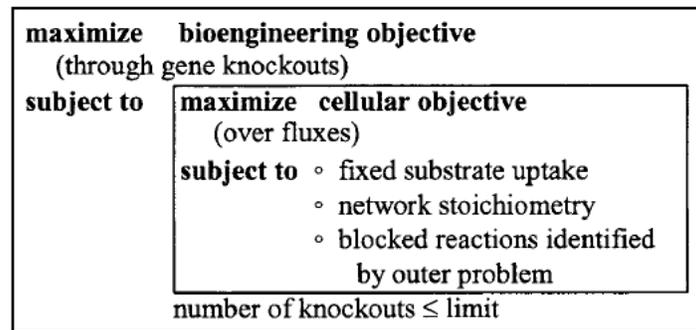


Figure 4.15: [15] **Struttura bilevel di OptKnock.** Il problema interno realizza l'allocazione dei flussi basandosi sull'ottimizzazione di un particolare obiettivo cellulare (ad esempio, la massimizzazione della biomassa). Il problema più esterno poi massimizza la produzione di determinati prodotti all'interno della rete, restringendo l'accesso alle sole reazioni necessarie, compatibilmente con il problema all'interno.

La soluzione diretta di questi due stadi di ottimizzazione è intrattabile in caso di reti estese a causa dell'annidamento dei due problemi. Per risolvere questo problema, Burgard et al. hanno sviluppato un approccio efficiente adottando la teoria della dualità dei problemi LP, secondo cui per ogni problema LP (primal) esiste una unica soluzione al problema di ottimizzazione (*Dual*), il cui valore ottimale dell'obiettivo è uguale a quello del problema primario.

Il problema duale (Ignizio and Cavalier, 1994) associato al problema interno (OptKnock) è il seguente:

$$\min v_{atp\ main} \cdot \mu_{atp} + v_{biomass}^{target} \cdot \mu_{biomass} + v_{glc\ uptake} \cdot glc \quad (Dual)$$

$$\begin{aligned} \text{subject to} \quad & \sum_{i=1}^N \lambda_i^{stoich} S_{i,glk} + \mu_{glk} + glc = 0 \\ & \sum_{i=1}^N \lambda_i^{stoich} S_{i,pts} + \mu_{pts} + glc = 0 \\ & \sum_{i=1}^N \lambda_i^{stoich} S_{i,biomass} + \mu_{biomass} = 1 \\ & \sum_{i=1}^N \lambda_i^{stoich} S_{i,j} + \mu_j = 0 \\ & \mu_j^{min} \cdot (1 - y_j) \leq \mu_j \leq \mu_j^{min} \cdot (1 - y_j), \\ & \mu_j \geq \mu_j^{min} \cdot (1 - y_j), \\ & \mu_j \leq \mu_j^{max} \cdot (1 - y_j), \\ & \mu_j \in \mathbb{R}, \\ & \lambda_i^{stoich} \in \mathbb{R}, \\ & glc \in \mathbb{R} \end{aligned} \quad \begin{aligned} & \forall j \in M, j \neq glk, pts, biomass \\ & \forall j \in M_{rev} \& j \notin M_{seccr\ only} \\ & \forall j \in M_{rev} \& j \in M_{seccr\ only} \\ & \forall j \in M_{irrev} \& j \notin M_{seccr\ only} \\ & \forall j \in M_{irrev} \& j \in M_{seccr\ only} \\ & \forall j \in N \end{aligned}$$

dove  $\lambda_i^{stoich}$  è la variabile duale associata ai vincoli stechiometrici,  $glc$  è la variabile associata al vincolo di assorbimento del glucosio e  $\mu_j$  è la variabile duale associata ad ogni altro vincolo corrispondente al flusso  $v_j$  nel Primal.  $\mu_j^{min}$  e  $\mu_j^{max}$  sono identificati minimizzando e poi massimizzando il loro valore, sotto i vincoli dettati dal problema Duale.

Se i problemi Primal e Dual sono legati tra loro, il valore della loro funzione obiettivo deve essere lo stesso, se si opera l'ottimizzazione. Ciò significa che ogni soluzione ottima ad entrambi i problemi può essere caratterizzata definendo i loro obiettivi uguali tra loro e mettendo assieme i rispettivi vincoli. In questo modo la formulazione bilevel dell'algoritmo di OptKnock può diventare il seguente problema MILP ad un unico livello:

$$\begin{aligned} \max v_{biochemical} \quad & \text{subject to} \\ & \sum_{j=1}^M S_{ij} v_j = 0; \\ & v_{pts} + v_{glk} = v_{glc\ uptake} \\ & v_{atp} > v_{atp\ main} \\ & \sum_{i=1}^N \lambda_i^{stoich} S_{i,glk} + \mu_{glk} + glc = 0 \\ & \sum_{i=1}^N \lambda_i^{stoich} S_{i,pts} + \mu_{pts} + glc = 0 \\ & \sum_{i=1}^N \lambda_i^{stoich} S_{i,biomass} + \mu_{biomass} = 1 \\ & \sum_{i=1}^N \lambda_i^{stoich} S_{i,j} + \mu_j = 0 \\ & \sum_{j \in M} (1 - y_j) \leq K \\ & v_{biomass} > v_{biomass}^{target} \\ & \mu_j^{min} \cdot (1 - y_j) \leq \mu_j \leq \mu_j^{min} \cdot (1 - y_j), \\ & \mu_j \geq \mu_j^{min} \cdot (1 - y_j), \\ & \mu_j \leq \mu_j^{max} \cdot (1 - y_j), \\ & \mu_j \in \mathbb{R}, \\ & v_j^{min} y_i \leq v_j \leq v_j^{max} y_i \\ & \lambda_i^{stoich} \in \mathbb{R}, \\ & y_j \in \{0, 1\} \end{aligned} \quad \begin{aligned} & (OptKnock) \\ & \forall i \in N \\ & \forall j \in M, j \neq glk, pts, biomass \\ & \forall j \in M_{rev} \& j \notin M_{seccr\ only} \\ & \forall j \in M_{rev} \& j \in M_{seccr\ only} \\ & \forall j \in M_{irrev} \& j \notin M_{seccr\ only} \\ & \forall j \in M_{irrev} \& j \in M_{seccr\ only} \\ & \forall i \in M \\ & \forall j \in N \\ & \forall j \in M \end{aligned}$$

Una delle caratteristiche più importanti di questa nuova formulazione è la possibilità di trovare sempre una soluzione.

## 4.7.2 OptGene

OptGene è un ulteriore algoritmo che permette di sperimentare modificazioni genetiche in silico. I vantaggi di questo algoritmo rispetto ad altri (ad esempio, OptKnock) sono sostanzialmente due:

1. OptGene richiede un minor tempo computazionale per ottenere il risultato e perciò permette di risolvere più facilmente analisi riguardanti reti complesse;
2. OptGene permette la formulazione di funzioni obiettivo non lineari, che sono di ampio interesse in molti problemi. Sarà quindi possibile ottimizzare anche delle funzioni che, a differenza di quanto visto finora, non siano necessariamente esprimibili come prodotto tra un vettore di coefficienti e un vettore di flussi.

#### 4.7.2.1 Algoritmo

In questa sede faremo riferimento a due diverse versioni dell'algoritmo di OptGene, che si differenziano principalmente sulla rappresentazione del genotipo metabolico:

- binOptGene: rappresentazione binaria;
- intOptGene: rappresentazione intera.

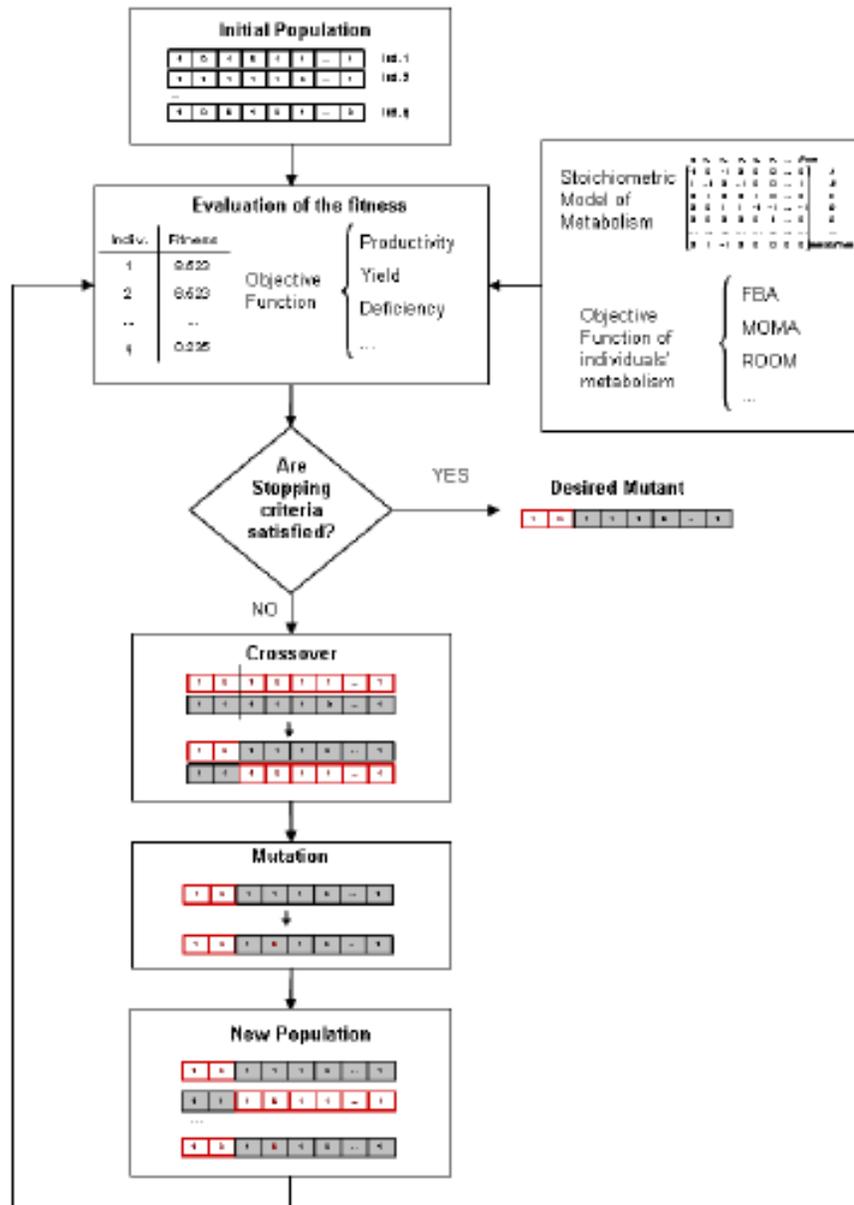


Figure 4.16: [13] **Overview dell'algoritmo OptGene.** Una popolazione di individui viene inizializzata specificando la presenza/assenza di ciascun gene in ciascun individuo. Viene assegnato uno score a ciascun individuo utilizzando FBA, MOMA o altri metodi e la funzione obiettivo. Gli individui vengono selezionati in base al loro punteggio di fitness e vengono accoppiati tra loro per produrre nuova prole. In modo casuale vengono introdotte delle mutazioni negli individui, in modo da ottenere una nuova popolazione. Tale ciclo evolutivo viene ripetuto fino a che si ottengono uno o più individui con un particolare carattere fenotipico desiderato

In Figura 4.16 [13] possiamo visualizzare la struttura dell’algoritmo binOptGene, di seguito invece possiamo riassumere i passi fondamentali dell’algoritmo per entrambe le rappresentazioni:

1. Pre-processing del modello: vengono eliminate dal modello eventuali reazioni duplicate e dead-end; i pathway lineari vengono rappresentati come un’unica reazione e le reazioni letali non vengono incluse tra i possibili target dell’algoritmo genetico. Questa fase permette una sensibile riduzione della dimensione della rete e quindi del numero di soluzioni ottime locali.
2. Rappresentazione cromosomica del genotipo metabolico: ogni reazione del modello può essere associata ad uno o più geni. Nella rappresentazione binaria ciascuno di questi geni viene associato ad una variabile binaria che indica la sua presenza (1) o assenza (0): un insieme di tali variabili forma quindi un ‘individuo’ (o ‘cromosoma’), che rappresenta un particolare mutante a cui mancano determinate reazioni che compaiono nel wild type. In Figura 4.17 [13] è visualizzabile quanto appena descritto. Nella rappresentazione intera invece gli individui sono composti da numeri interi che rappresentano solamente il numero di geni che sono stati posti in KnockOut: in questo modo si può imporre il numero di geni da eliminare direttamente modificando la dimensione degli individui. Il fenotipo di ciascun individuo si può ottenere mediante FBA o altri algoritmi: il nuovo problema è quindi quello di determinare il set di geni da eliminare per ottenere il fenotipo desiderato.

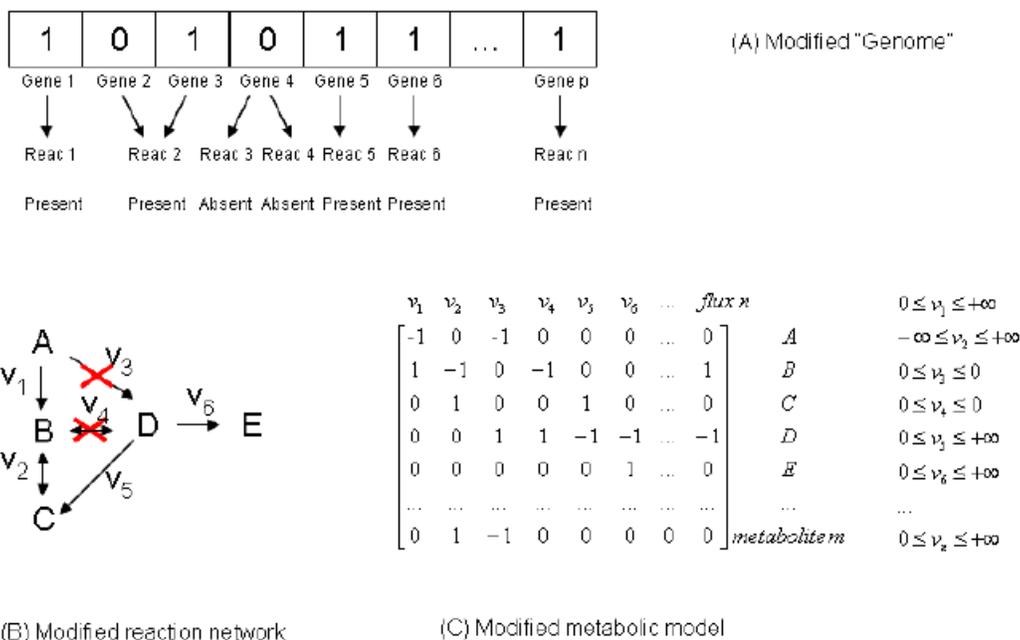


Figure 4.17: [13] *Rappresentazione di un genotipo metabolico.* (A): Ad ogni gene dell’organismo è associata una variabile binaria che rappresenta la sua presenza (1) o la sua assenza (0) nel mutante. (B): I geni sono associati ad una o più reazioni. (C): Se una determinata reazione è assente, i suoi lower ed upper bound vengono posti uguali a zero, modificando il modello iniziale.

3. Inizializzazione della popolazione: per poter inizializzare l’algoritmo è necessario definire una popolazione iniziale, che può essere determinata in vari modi: ad esempio assegnando in modo casuale gli stati (assente/presente) ai geni oppure assegnando lo stato di presenza a tutti i geni. (Solitamente si preferisce la prima tra queste due possibilità).
4. Assegnazione di uno score agli individui: in base al valore della funzione obiettivo desiderata, che può essere calcolata tramite FBA, MOMA (minimization of metabolic adjustment), ROOM (regulatory on-off minimization) o altri metodi, viene assegnato uno score a ciascun individuo, che determinerà se tale individuo verrà riprodotto e/o propagato nella nuova generazione.
5. Crossover dei cromosomi: In base allo score ottenuto al passo precedente, gli individui migliori vengono incrociati tra loro (solitamente vengono selezionati gli individui con uno score maggiore) per ottenere

una nuova prole.

6. **Mutazioni:** gli individui scelti per essere propagati nella nuova popolazione sono, con una certa probabilità, dei mutanti (cioè con dei geni posti in KnockOut).
7. **Nuova popolazione e terminazione:** le mutazioni e gli incroci effettuati danno origine ad una nuova popolazione, che può essere soggetta nuovamente ad un nuovo ciclo evolutivo (valutazione, crossover e mutazione). Questo ciclo viene ripetuto finché non si ottiene un individuo con il fenotipo desiderato.

Va sottolineato però che OptGene può portare a diverse soluzioni ottime, tale fatto può essere visto come un vantaggio se si considera che si può scegliere la soluzione più facilmente realizzabile nella realtà.

### 4.7.3 Genetic design local search (GDLS)

Per raggiungere il duplice obiettivo di massimizzare un obiettivo cellulare (ad esempio la biomassa) e allo stesso tempo di massimizzare la produzione di determinati metaboliti all'interno della rete, si può utilizzare l'algoritmo di Genetic design local search (GDLS), che implementa una ricerca locale tramite molti percorsi alternativi. In questo modo, a differenza di una bi-level FBA, che spesso non permette di raggiungere una soluzione, data la complessità computazionale richiesta, si riesce ad esplorare lo spazio delle manipolazioni genetiche in modo efficiente e diminuendo la complessità del problema, tramite l'implementazione di una ricerca locale.

Anche se GDLS può essere utilizzato per determinare il genetic design coinvolto in qualsiasi tipo di manipolazione, in questa descrizione faremo riferimento all'ottimizzazione dei KnockOut.

Per descrivere il funzionamento dell'algoritmo introduciamo una nuova terminologia: con obiettivo "naturale" faremo riferimento all'obiettivo cellulare (ad esempio la biomassa), mentre con obiettivo "sintetico" indicheremo il flusso inerente alla produzione di determinati metaboliti all'interno della rete.

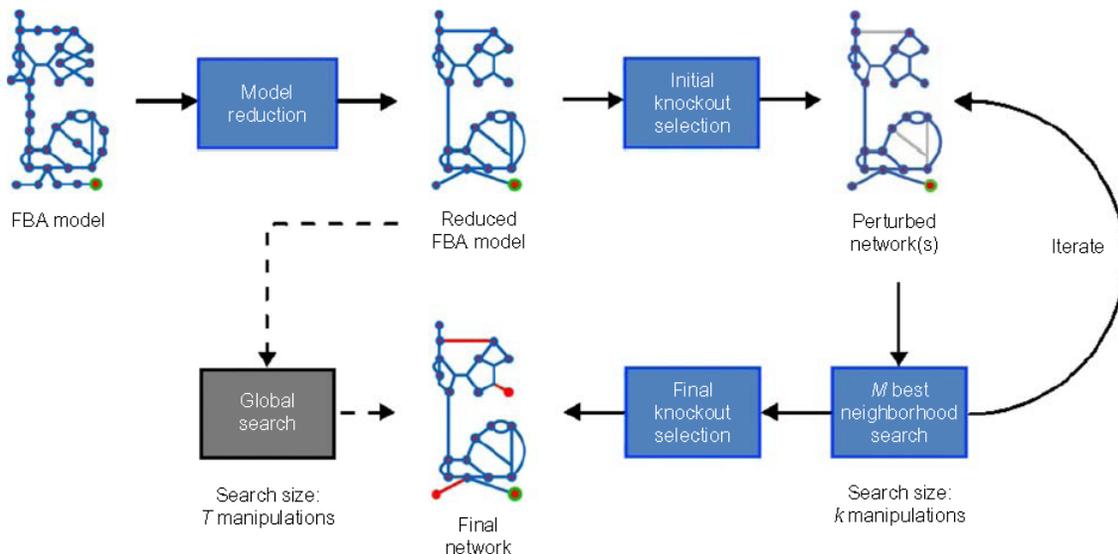


Figure 4.18: [21] **Overview di GDLS.** Il modello utilizzato da FBA viene inizialmente ridotto in un modello equivalente contenente meno geni, reazioni e metaboliti. Su questa rete ridotta viene effettuata una selezione iniziale di un KnockOut, causando una perturbazione della rete. A questo punto viene effettuata una ricerca tra i vicini, durante la quale MILP è utilizzato per ricercare le  $M$  migliori strategie di manipolazione genetica, che differiscono dal punto iniziale per al massimo  $k$  manipolazioni aggiuntive. Le  $M$  reti perturbate così ottenute vengono utilizzate per un'altra ricerca locale: si continua fino a che non si verifica più nessun miglioramento all'interno dell'intera gamma delle  $T$  manipolazioni possibili nel modello FBA ridotto. Alternativamente, si può operare tramite una ricerca globale utilizzando semplicemente MILP per ricercare la migliore strategia di manipolazione genetica che differisce dal modello ridotto inizialmente per al massimo  $T$  manipolazioni.

In Figura 4.18 [21] è rappresentata un'overview del funzionamento dell'algoritmo, che si può riassumere nelle seguenti fasi:

1. Il modello FBA ricevuto in input viene ridotto ad una dimensione minore, eseguendo alcune modifiche;
2. rimozione delle reazioni dead-end, ossia delle reazioni nelle quali i metaboliti sono associati solamente a quella reazione e quindi non possono trasportare flusso;
3. tutte le reazioni tali per cui i metaboliti sono associati esattamente a due reazioni (linked reactions), vengono ridotte in un'unica reazione. Tale operazione si può fare perchè, in questo caso, per avere la conservazione dei metaboliti, le due reazioni si trovano sempre nello stesso rapporto:

$$S_{ij1}v_{j1} + S_{ij2}v_{j2} = 0;$$

l'equazione unica risultante è quindi:

$$v_{j1} = -\frac{S_{ij2}}{S_{ij1}}v_{j2}.$$

4. viene massimizzato e minimizzato ciascun flusso, soggetto ai vincoli del problema, per ottenere dei vincoli stretti per ciascun flusso:

$$\begin{array}{ll} \max/\min & v_j \\ \text{subject to} & Sv = 0 \\ & a \leq v \leq b. \end{array}$$

Siano  $v_{j,l}$  e  $v_{j,u}$  il risultato di tale massimizzazione/minimizzazione. Se per ogni  $j$ , si ha che  $v_{j,u} \leq v_{j,l}$ , viene rimossa la reazione  $j$ , perchè il flusso attraverso questa dovrà sempre essere nullo.

5. Queste modifiche vengono ripetute iterativamente fino a che non si trova più alcuna reazione da modificare/rimuovere.
6. I KnockOut sono modellati in termini di gene set, che può coinvolgere una o più reazioni usando la mappa GPR, che associa ciascuna reazione ai geni corrispondenti. Inizialmente quindi viene scelto un set di KnockOut, nella rete già ridotta al punto precedente (tali geni possono essere selezionati in modo random: ogni diversa inizializzazione può portare a risultati differenti).
7. Dopo aver settato in modo corretto il problema MILP che caratterizza GDLS, l'algoritmo inizia ad eseguire delle ricerche, iterativamente, mantenendo una popolazione di soluzioni di dimensione  $M$ . Questo significa che, a partire dal set iniziale di KnockOut, viene applicata una routine di ottimizzazione MILP, per ricercare  $M$  set di KnockOut, che si differenziano dal set iniziale per al massimo  $K$  KnockOut. Poichè i KnockOut possono anche essere rimossi dopo essere già stati selezionati, GDLS può operare anche in backtracking. Idealmente tale ottimizzazione MILP dovrebbe portare ad una soluzione, ma talvolta può succedere che i pacchetti software che vengono utilizzati per implementare la routine non portino a soluzioni fisicamente possibili: ciò significa che si possono ottenere al massimo  $M$  soluzioni a questo step, ma non necessariamente esattamente  $M$ .
8. Successivamente, le  $M$  soluzioni trovate vengono usate come set iniziale di KnockOut per l'iterazione successiva, in cui viene applicata la stessa procedura di ottimizzazione MILP del passo precedente. Di conseguenza, questa seconda iterazione genera  $M^2$  soluzioni uniche, tra le quali vengono selezionate le  $M$  migliori, da porre come set iniziale di KnockOut per l'iterazione ancora successiva.
9. Queste operazioni vengono iterate fino a che il set delle migliori soluzioni non è soggetto più ad alcuna modifica, oppure fino a che il set di migliori soluzioni diventa vuoto.

Da quanto descritto appare chiaro che i fattori chiave dell'ottimizzazione MILP implementata da GDLS sono:

- $K$ : il numero di KnockOut che possono essere aggiunti o rimossi da ogni soluzione ad ogni iterazione (cioè il raggio d'azione di ogni iterazione);
- $M$ : il numero di soluzioni che viene mantenuto dopo ogni iterazione.

Solitamente, un  $K$  elevato permette una migliore capacità di ricerca, mentre un basso valore per  $K$  ed  $M$  comportano tempo computazionale minore a costo di soluzioni con flusso sintetico inferiore.

Solitamente questo approccio necessita di un tempo computazionale inferiore rispetto all'approccio di ricerca globale implementato da OptKnock, tuttavia non è garantito che GDLS identifichi una soluzione ottima globale, poichè le sue ricerche vengono sempre effettuate a livello locale.

#### 4.7.3.1 Algoritmo di ottimizzazione MILP per la ricerca locale

Per permettere a GDLS di operare a livello genetico, è necessario implementare la mappa GPR, che definisce come determinate manipolazioni genetiche influiscano nelle reazioni della rete. Per un insieme di  $L$  manipolazioni genetiche, rappresentiamo tale mappa GPR con una matrice  $G \in \mathbb{R}^{L \times n}$ , dove l'elemento  $G_{l,j}$  di  $G$  vale 1 se la manipolazione genica  $l$ -esima influenza la reazione  $j$  e vale 0 in caso contrario.

Inizialmente il problema viene posto come una ottimizzazione bi-level:

$$\begin{aligned}
 & \max g^T v \\
 & \text{subject to} \quad \sum_{l=1}^L y_l \leq C \\
 & \quad y_l \in \{0, 1\}, \quad l = 1, \dots, L \\
 & \quad \max f^T v \\
 & \quad \text{subject to} \quad Sv = 0 \\
 & \quad (1 - y)^T G_j a_j \leq v_j \leq (1 - y)^T G_j b_j, \quad j = 1, \dots, n
 \end{aligned}$$

dove  $g$  è il vettore dell'obiettivo "sintetico", il cui  $j$ -esimo elemento  $g_j$  rappresenta il peso della reazione  $j$  all'interno della funzione obiettivo sintetico;  $y$  è il vettore dei KnockOut, il cui elemento  $l$ -esimo  $y_l$  vale 1 se i geni coinvolti nella manipolazione  $l$  sono posti in KnockOut e 0 altrimenti;  $G_j$  identifica la  $j$ -esima colonna di  $G$  e infine  $C$  è il numero massimo di KnockOut consentiti.

Tale problema bi-level può essere convertito in un problema MILP equivalente, sfruttando il problema duale dell'ottimizzazione dell'obiettivo naturale, che si può descrivere nel modo seguente:

$$\begin{aligned}
 & \max g^T v \\
 & \text{subject to} \quad \sum_{l=1}^L y_l \leq C \\
 & \quad y_l \in \{0, 1\}, \quad l = 1, \dots, L \\
 & \quad Sv = 0 \\
 & \quad (1 - y)^T G_j a_j \leq v_j \leq (1 - y)^T G_j b_j, \quad j = 1, \dots, n \\
 & \quad f^T v = \sum_{j=1}^n \nu_j b_j - \mu_j a_j \\
 & \quad f_j - \sum_{i=1}^m \lambda_i S_{ij} - \nu_j + \mu_j - \xi_j = 0, \quad j = 1, \dots, n \\
 & \quad -Dy^T G_j \leq \xi_j \leq Dy^T G_j, \quad j = 1, \dots, n \\
 & \quad \mu, \nu \geq 0.
 \end{aligned}$$

dove  $\lambda$  è la variabile duale del vincolo  $Sv = 0$ ;  $\mu$  e  $\nu$  sono le variabili duali dei lower ed upper bound rispettivamente;  $\xi$  è la variabile duale corrispondente al vincolo  $v_j = 0$  se  $y_j = 0$  e infine  $D$  è un numero scalare sufficientemente grande per assicurare che  $\xi_j$  sia non vincolato se  $y^T G_j$  è diverso da zero.

Supponiamo ora di iniziare con un vettore di KnockOut  $v^{(0)}$ . Ad ogni iterazione viene mantenuto un insieme di strategie di KnockOut  $Y^{(i)}$ , perciò poniamo  $Y^{(0)} := \{v^{(0)}\}$  e  $i = 0$ .

Per ogni vettore di KnockOut  $\tilde{y}$  in  $Y^{(i)}$  risolviamo il problema MILP  $M$  volte usando il vincolo aggiuntivo:

$$\sum_{l:\tilde{y}_l=0} y_l + \sum_{l:\tilde{y}_l=1} (1 - y_l) \leq K. \quad (1)$$

Tale vincolo serve a limitare la dimensione del vicinato in cui eseguire la ricerca al valore  $K$ . Dopo aver risolto il problema MILP con questo nuovo vincolo, definiamo l'insieme  $Y^{(i+1)}$ , contenente le soluzioni ottime  $y^*$ . Poi, viene risolto nuovamente il problema MILP per  $M - 1$  volte con l'aggiunta di un altro ulteriore vincolo (oltre al precedente):

$$\sum_{l:\tilde{y}_l=0} y_l + \sum_{l:\tilde{y}_l=1} (1 - y_l) \geq 1 \quad (2)$$

per tutti gli  $\tilde{y}$  contenuti in  $Y^{(i+1)}$ . Se non otteniamo una soluzione ottima o se tale soluzione non è fisicamente possibile, semplicemente non modifichiamo  $Y^{(i+1)}$  e continuiamo. Da ciascun  $\tilde{y}$  in  $Y^{(i)}$  otteniamo  $M$  soluzioni migliori al problema MILP con il vincolo (1), che poniamo nel set  $Y^{(i+1)}$ , che sostanzialmente può contenere fino a  $|Y^{(i)}| \cdot M$  elementi. Successivamente rimuoviamo da  $Y^{(i+1)}$  alcuni elementi per ridurlo ad una dimensione di  $M$ , tenendo solo le soluzioni migliori.

Infine si incrementa  $i$  e si ripete la procedura dall'inizio fino a che  $Y^{(i+1)} = Y^{(i)}$ .

# Chapter 5

## COBRA Toolbox

Per un descrizione approfondita di COBRA Toolbox, rimandiamo all'Appendice B, in cui è descritta anche la procedura necessaria per l'installazione. In questa sede ci limiteremo ad alcune nozioni basilari per introdurre il software e ad alcuni aspetti critici e suggerimenti utili per l'utilizzo di tale strumento per fini simili a quelli della nostra tesi.

### 5.1 Struttura di COBRA Toolbox

Per effettuare tutte le nostre analisi è stato utilizzato COBRA Toolbox: un software creato per eseguire la ricostruzione e l'analisi di rete metaboliche constraint-based. Questo pacchetto è eseguibile in ambiente Matlab e contiene i metodi necessari per poter simulare il comportamento cellulare utilizzando un approccio constraint-based, cioè implementando la ricostruzione del modello oggetto di studio tramite i vincoli dettati dalla matrice stechiometrica.

COBRA Toolbox consiste in un insieme di file, che possono essere salvati in una cartella locale nel computer dell'utilizzatore.

Una prima versione di questo software è stata realizzata nel 2007 e comprendeva già diverse funzionalità:

- previsione del comportamento della cellula (sia in stato stazionario sia in termini dinamici) ottimizzando la crescita della stessa;
- analisi degli effetti dovuti al KnockOut di determinati geni all'interno della rete;
- analisi della robustezza, intesa come dipendenza della crescita dalla variazione di determinati flussi;
- campionamento dello spazio dei possibili stati del modello del metabolismo della cellula;
- determinazione di moduli all'interno della stessa rete.

Nel 2011 è stata realizzata una seconda versione di tale Toolbox, che aggiunge alla precedente altre funzionalità molto utili nello studio del comportamento delle cellule:

- capacità di rintracciare e riempire eventuali gap all'interno della rete;
- analisi C;
- metabolic engineering;
- omics-guided analysis;
- visualizzazione della rete metabolica.

L'immagine in Figura 5.1 raffigura la strategia utilizzata da COBRA per ricostruire ed analizzare la rete. Per costruire il modello, COBRA richiede in ingresso una lista delle reazioni biochimiche coinvolte nella rete e le relazioni esistenti tra queste reazioni e i geni e le proteine coinvolti. Inserendo inoltre le informazioni riguardo i flussi di scambio con l'esterno (la disponibilità di nutrienti) e i lower e gli upper bound per ciascuna reazione, il software è in grado di realizzare un modello corrispondente alla rete, sottoforma di struttura ('struc' in Matlab), che può essere utilizzato per eseguire le diverse analisi.

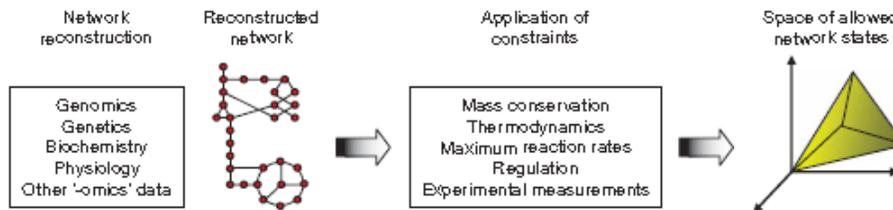


Figure 5.1: *Strategia utilizzata da COBRA. Viene utilizzato un set di vincoli di diverse tipologie per definire un range di “stati” che la rete può potenzialmente raggiungere. Nel caso di rete metabolica questi stati corrispondono a distribuzioni di flussi.*

Uno dei maggiori vantaggi che offre questo software consiste nel poter effettuare un gran numero di studi sulla rete operando “in silico”: i risultati ottenuti ovviamente non possono essere assunti come assolutamente corretti, a causa dei limiti computazionali degli strumenti utilizzati, ma hanno un forte valore se usati come ipotesi di partenza per effettuare poi, in un secondo momento, degli esperimenti reali. Avere la possibilità di eseguire preliminarmente queste analisi permette di ridurre i costi, le risorse e il tempo necessari per gli esperimenti reali.

Prima di entrare nel dettaglio dei metodi utilizzati per la nostra analisi è bene però tenere a mente alcune limitazioni del software:

1. la maggiorparte dei metodi implementati nel Toolbox calcolano una distribuzione dei flussi, che consiste nel quantificare i flussi di ciascuna reazione all’interno della rete, assegnando un valore specifico. Nella realtà questo non è sempre possibile, infatti, soprattutto nelle reti molto estese, i vincoli stabiliti determinano un range di valori che può essere attribuito a ciascun flusso, ma non si può caratterizzare una soluzione unica e ben determinata.
2. alcuni dei metodi proposti si basano sul fatto che le cellule cerchino di diventare sempre più grandi e quindi di massimizzare la biomassa: neppure questa ipotesi è sempre verificata nella realtà. tuttavia può essere considerata un’assunzione accettabile in molti interessanti casi di studio.

## 5.2 Nozioni basilari per l’utilizzo di COBRA Toolbox

Poichè abbiamo già descritto in precedenza le componenti fondamentali per l’analisi di una rete metabolica basata sul metodo constraint-based, ci limitiamo qui a riportare gli elementi necessari per la costruzione del modello tramite questo software.

- La componente principale richiesta in ingresso dal Toolbox è la matrice stechiometrica  $S$ , che rappresenta in modo matematico le relazioni presenti nella rete. Come descritto in precedenza, essa rappresenta ciascuna reazione come una colonna e ciascun metabolita come una riga: per ciascuna reazione  $j$ , nella colonna  $j$ -esima sarà rappresentato un 1 in corrispondenza alle righe dei metaboliti che vengono prodotti dalla reazione, un  $-1$  in corrispondenza alle righe dei metaboliti consumati e uno 0 in tutte le altre posizioni.
- In corrispondenza a ciascuna reazione della rete, è richiesta la definizione di un lower e di un upper bound, che definiscono i limiti ammissibili per i flussi di ciascuna reazione (se non si vuole limitare tali flussi, si settano tali vincoli a  $\pm\infty$ ). È necessario però che almeno una reazione abbia dei limiti finiti, altrimenti molto difficilmente si potranno ottenere dei risultati significativi.

In Figura 5.2 possiamo osservare le componenti descritte appena sopra.

## 5.3 Funzionalità di COBRA Toolbox

Come abbiamo già osservato in precedenza, esistono due versioni di questo software, di cui la seconda è un’estensione della prima. In Tabella 5.1 possiamo visualizzare rapidamente quali sono le funzionalità del

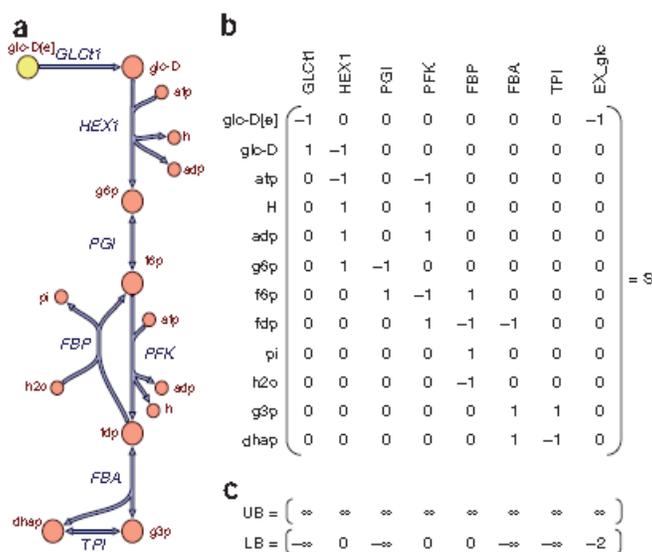


Figure 5.2: Esempio degli elementi fondamentali per l'analisi metabolica. a) rete metabolica; b) matrice stechiometrica  $S$  relativa alla rete rappresentata in a); c) lower e upper bound delle reazioni della rete.

	COBRA 1.0	Novità in COBRA 2.0
FBA	Growth-rate optimization	Loop law
	Robustness analysis	Geometric FBA
	Gene deletion studies	
	Flux variability	
	MOMA	
Fluxomics	-	C13 data fitting and flux estimation
Gap filling	-	Experimental design
		detectDeadEnds
		gapFind
Input/output	Read/write SBML (level 2, version 1)	Read/write SBML (level 2, version 4)
Metabolic engineering	-	OptKnock
		OptGene
		GDLS
Reconstruction	Model curation tools	Create submodels using omics data
Sampling	Artificial centering hit and run (ACHR) sampling	Updated ACHR sampling (parallel/multipoint)
Test suite	-	Examples
		Verify installations
Visualization	-	Display maps
		Overlay data - flux distributions
		flux variability

Table 5.1: Caratteristiche e funzionalità di COBRA Toolbox 1.0 e 2.0

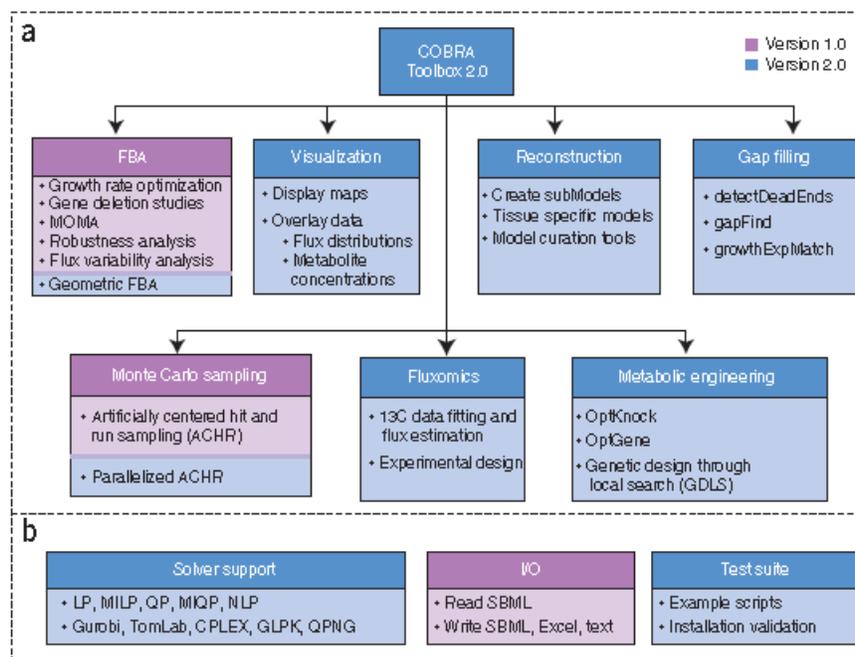


Figure 5.3: **Overview di COBRA Toolbox.** a) Sette categorie di metodi inclusi nella versione 2.0 del Toolbox. b) Solver supportati dai metodi implementati; formati accettati in ingresso e in uscita; sono inclusi inoltre degli esempi e dei metodi per verificare la corretta installazione dei tools.

Toolbox legate alla prima versione (COBRA 1.0) e quali alla seconda (COBRA 2.0). La Figura 5.3 inoltre ci aiuta nel capire come tali funzioni siano collegate tra loro.

I modelli possono essere caricati in due formati:

- SBML (System Biology Markup Language);
- Excel.

Per poter utilizzare SBML è necessario aver precedentemente installato la libreria libSBML e la SBMLToolbox per Matlab.

Alcuni modelli campione possono essere liberamente scaricati dal database BiGG (<http://bigg.ucsd.edu/>) oppure da Model SEED (<http://seed-viewer.theseed.org/models>).

La struttura del file in formato SBML è visibile in Appendice B.12, mentre il formato Excel sarà presentato più avanti nella descrizione dei modelli utilizzati. Molte funzioni di COBRA richiedono delle informazioni che non sono necessariamente contenute nel formato standard SBML, in particolare: l'associazione tra geni e reazioni (Figura 5.4) è necessaria per poter collegare la rete metabolica al genoma, l'indicazione del sottosistema è utile per effettuare una classificazione ontologica, la formula e la carica di ciascun metabolita coinvolto sono necessari per poter garantire che il modello sia fisicamente consistente. Infine si ha la possibilità di aggiungere altre informazioni aggiuntive, come ad esempio KEGG o CAS IDs, che possono rivelarsi utili per un'analisi più dettagliata dei risultati.

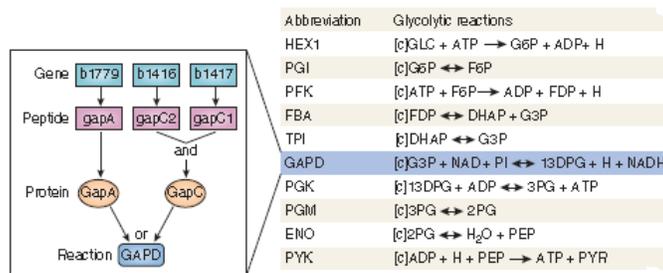


Figure 5.4: Esempio di relazione tra geni e reazioni.

## Aspetti Critici e suggerimenti:

### Inizializzazione del Toolbox e scelta dei risolutori

Subito dopo aver inizializzato il Toolbox con il comando:

```
initCobraToolbox;
```

è bene ricordarsi di modificare i risolutori, se non sono settati correttamente già di default. Ricordiamo infatti che alcuni risolutori non portano a delle soluzioni corrette, ad esempio:

- per problemi 'LP': NON utilizzare Gurobi. (Ad esempio si può utilizzare 'gplk');
- per problemi 'MILP': NON utilizzare 'gplk'. (Ad esempio si può utilizzare 'Gurobi').

### Test

Prima di eseguire qualsiasi tipo di analisi è bene eseguire un test di tutte le funzionalità del Toolbox al fine di verificare se quelle di interesse funzionano correttamente. Ricordiamo che il comando

```
testAll();
```

permette di eseguire con un solo comando il test di tutte le funzionalità principali.

### Importazione del modello

Ricordiamo che per importare un modello esso deve essere in un formato compatibile.

- Se il modello è in formato SBML, utilizzare il comando:

```
model = readCbModel([filename]);
```

- Se il modello è in formato Excel, utilizzare il comando:

```
model = xls2model([filename], [BiomassReaction]).
```

In entrambi i casi:

1. ricordarsi di inserire tutte le reazioni di scambio necessarie affinché le reazioni possano avvenire:

$$EX_{metab}(c) : \quad [c] : metab \rightleftharpoons$$

2. per verificare se tutte le reazioni sono potenzialmente possibile, suggeriamo di utilizzare la Robustness Analysis per tutte le reazioni interne della rete e osservare la matrice contenente i valori dei Flussi di Controllo in uscita: se una reazione interna assume sempre valore 0 in tutti gli step, significa che per qualche motivo essa non potrà mai avere luogo. Deve esserci perciò un errore di modello.

### Scelta della funzione obiettivo e verifica

La scelta della funzione obiettivo non è una questione facile. Consigliamo in ogni caso di ottimizzare un solo flusso durante l'ottimizzazione, al fine di ottenere il risultato richiesto.

- La situazione più semplice è quella in cui si vuole ottimizzare la biomassa: basterà porre al valore 1 la componente del vettore *modello.c* relativa alla reazione *Biomassa*.
- Se invece non si vuole massimizzare la crescita di una cellula, occorre scegliere con cautela l'obiettivo: se ad esempio, come nel nostro caso, si vuole osservare come si distribuiscono i flussi in situazione in cui si produce un determinato metabolita, poichè si suppone che esso serva al funzionamento del sistema (nel nostro caso ci riferiremo alla melatonina), si può porre a 1 la componente del vettore *modello.c* relativa alla reazione che produce oppure che utilizza tale metabolita.

Un metodo per settare l'obiettivo di interesse, senza rischiare di commettere errori consiste nel azzerare completamente il vettore *modello.c* e poi settare a 1 solo la componente relativa alla reazione di interesse:

$$\begin{aligned} \text{modello.c}(1 : \text{end}) &= 0; \\ \text{modello.c}([\text{indice}_{rxn}]) &= 1; \end{aligned}$$

Per verificare di aver settato correttamente l'obiettivo, consigliamo di utilizzare il comando:

$$\text{checkObjective}(\text{model});$$

che restituisce la reazione da ottimizzare.

## Chapter 6

# Modelli FBA del metabolismo del triptofano

### 6.1 Considerazioni generali

Al fine di poter utilizzare il Toolbox COBRA per effettuare le analisi di nostro interesse, è stato necessario costruire un modello del metabolismo del triptofano nell'essere umano tale da essere compatibile con i formati accettati in ingresso al Toolbox.

Inizialmente abbiamo fatto riferimento al modello presente nel sito di Kegg<sup>1</sup>, che permette di scaricare i pathway di interesse in formato *KGML*. Attraverso due diversi convertitori (Kegg Converter e Kegg Translator) è stato possibile convertire il file in formato SBML, accettato in ingresso da COBRA Toolbox.

Ci siamo però trovati di fronte ad alcune difficoltà, che ci hanno spinto a scegliere una via alternativa. Tali difficoltà possono essere brevemente riassunte nei seguenti punti:

- il modello, comprendendo una piccola componente dell'intero metabolismo umano, non contiene le reazioni di scambio, che permettono ai metaboliti che non vengono prodotti direttamente dal pathway di interesse, di entrare né uscire dalla rete. Questo problema porta a delle soluzioni sempre nulle durante l'esecuzione della Flux Balance Analysis, a causa della mancanza degli ingressi.
- il modello così convertito non contiene le informazioni sui geni corrispondenti alle reazioni. Questa informazione mancante non permette di analizzare i KnockOut né di verificare quali siano le reazioni e i geni essenziali per la produzione di metaboliti di interesse.
- Nonostante sia possibile inserire nuove reazioni (ad esempio reazioni di scambio) e la mappa GPR in un modello in formato SBML, a causa di una scarsa conoscenza della struttura di questi file, nell'attuare questo passaggio esiste il rischio di introdurre degli errori all'interno della rete.
- Non esiste una mappa della rete relativa al modello in questione: non sarebbe possibile quindi utilizzare tutte le funzionalità del Toolbox inerenti alla rappresentazione dei flussi che utilizzano la costruzione di un oggetto 'map'.

In seguito a queste difficoltà riscontrate, si è preferito costruire il modello in formato Excel, più facilmente gestibile dall'utente.

Per quanto riguarda la scelta del modello da utilizzare, abbiamo seguito due strade differenti:

*Modello completo*) fa riferimento al modello del metabolismo del triptofano scaricabile dal database BiGG<sup>2</sup>, che permette di scaricare in formato *.txt* le liste delle reazioni e dei metaboliti coinvolti nella rete. Il sito permette anche il download del modello completo del metabolismo in formato SBML, ma, almeno in un primo momento, questo modello è troppo esteso per i nostri scopi. Dal database BiGG si può inoltre scaricare la mappa del modello in formato *.txt*, compatibile con gli ingressi accettati da COBRA Toolbox. Purtroppo, come vedremo meglio in seguito, è stato necessario fare uso di un ulteriore programma per selezionare la mappa di interesse.

---

<sup>1</sup>[http://www.genome.jp/kegg-bin/show\\_pathway?org\\_name=hsa&mapno=00380&mapscale=&show\\_description=hide](http://www.genome.jp/kegg-bin/show_pathway?org_name=hsa&mapno=00380&mapscale=&show_description=hide)

<sup>2</sup><http://bigg.ucsd.edu/>

*Modello ridotto*) A partire dalle informazioni ricavate dall'analisi stechiometrica del modello cinetico analizzato precedentemente (ovvero dalle equazioni di bilanciamento di massa che descrivono le reazioni coinvolte nel pathway, ottenibili tramite l'utilizzo del software Copasi), è stato possibile ricavare tutte le informazioni per la costruzione di un secondo modello basato su vincoli strutturali. Poiché il modello cinetico di partenza non era completamente modellizzato, questo secondo modello è meno completo del primo, ma ci permetterà di effettuare alcune comparazioni tra i risultati ottenuti dall'analisi strutturale e quelli ottenuti tramite il modello cinetico.

Da ora in poi, per riferirci ai modelli basati su vincoli strutturali, parleremo di 'modelli FBA', per distinguerli dal modello cinetico, basato sulle equazioni differenziali.

## 6.2 Costruzione dei modelli FBA in formato Excel e visualizzazione

### 6.2.1 Struttura del file contenente il modello

Il Toolbox COBRA richiede che il file in ingresso in formato Excel sia costituito da due cartelle:

1. 'reactions';
2. 'metabolites'.

#### 6.2.1.1 Cartella 'reactions'

La cartella 'reactions' deve contenere 15 colonne, che descrivono tutte le informazioni relative alle reazioni del modello. Per capire il significato di ciascuna colonna, riporteremo in aggiunta anche un esempio di una singola reazione. Quando una colonna viene indicata come 'facoltativa', è da intendersi che essa deve comparire nel file, ma può essere vuota.

1. 'Abbreviation': nome abbreviato della reazione ('ACSRTNMT');
2. 'Name': nome completo della reazione ('S-Adenosyl-L-homocysteine: N-acetylserotonin O-methyltransferase');
3. 'Reaction': equazione che descrive la reazione (completa dei compartimenti) ('[c] : Nacsertn + amet -> ahcys + h + melatn');
4. 'GPR': informazioni (regole booleane) inerenti ai geni associati alla reazione ('438.1' oppure 'Asmt.1');
5. 'Genes': colonna facoltativa che indica quali geni partecipano alla legge booleana inerente alla reazione di interesse ('438.1' oppure 'Asmt.1');
6. 'Protein': colonna facoltativa che indica le regole booleane in termini di proteine ('Asmt.1');
7. 'Subsystem': sottosistema in cui compare la reazione ('Tryptophan metabolism');
8. 'Reversible': valore booleano che indica se la reazione è reversibile ('1') o irreversibile ('0'). (Nel caso di esempio la reazione è irreversibile, perciò indicheremo il valore '0');
9. 'Lower bound': limite inferiore del flusso attraverso la reazione ('0');
10. 'Upper bound': limite superiore del flusso attraverso la reazione ('100');
11. 'Objective': colonna facoltativa che setta i coefficienti della funzione obiettivo, ciò significa che se una determinata reazione partecipa alla funzione obiettivo avrà un valore positivo in questa colonna, altrimenti il valore verrà posto a zero ('0');
12. 'Confidence Score': può valere '0, 1, 2, 3, 4' ('4');
13. 'EC. Number': enzima che catalizza la reazione ('2.1.1.4');
14. 'Notes': colonna facoltativa che può contenere alcune informazioni aggiuntive sulla reazione ('The enzyme carries out similar reactions and the KEGG map includes this reaction as well.');
15. 'References': colonna facoltativa che può indicare dove sono state reperite le diverse informazioni ('').

### 6.2.1.2 Cartella 'metabolites'

La cartella 'metabolites' riguarda tutti i metaboliti che partecipano alla rete metabolica e deve contenere 11 colonne. Se un metabolita compare in più compartimenti, esso deve comparire più volte nella lista.

1. 'Abbreviation': nome abbreviato del metabolita ('ahcys');
2. 'Name': nome completo del metabolita ('S-Adenosyl-L-homocysteine');
3. 'Formula (neutral)': formula chimica del metabolita (se neutro);
4. 'Formula (charged)': formula chimica del metabolita (se carico) ('C14H20N6O5S');
5. 'Charge': carica del metabolita ('0');
6. 'Compartment': compartimento in cui compare il metabolita ('Cytosol, Endoplasmic Reticulum, Nucleus, Mitochondria');
7. 'KEGG ID': identificativo nel database di Kegg ('C00021');
8. 'PubChem ID': identificativo nel database PubChem ('C00021');
9. 'ChEBI': identificativo nel database ChEBI ('16680');
10. 'InChI string' ('1S/C14H20N6O5S/c15-6(14(23)24)1-2-26-3-7-9(21)10(22)13(25-7)20-5-19-8-11(16)17-4-18-12(8)20/h4-7,9-10,13,21-22H,1-3,15H2,(H,23,24)(H2,16,17,18)/t6-,7+,9+,10+,13+/m0/s1 ');
11. 'Smiles' ('N[C@@H](CCSC[C@H]1O[C@H]([C@H](O)[C@@H]1O)N1C=NC2=C(N)N=CN=C12)C(O)=O');

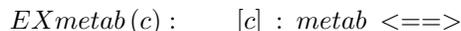
### 6.2.2 Riperimento delle informazioni e assunzioni

Per poter completare il file, riempiendo tutte le colonne necessarie per la costruzione dei modelli, ci siamo serviti di diversi database disponibili online:

- dal database BiGG (per il modello completo) abbiamo potuto scaricare facilmente le liste delle reazioni e dei metaboliti partecipanti alla rete. Tutte le reazioni descritte (eccetto quelle di scambio) appartengono al sottosistema del metabolismo del triptofano, ad eccezione di '2OXOADOXm' che partecipa al metabolismo della Lisina.
- Per quanto riguarda le informazioni genetiche abbiamo utilizzato ancora una volta il database BiGG, confrontando però le informazioni con quelle ottenute dal database 'Recon X' (<http://humanmetabolism.org/>).
- Molte informazioni relative ai metaboliti sono state ritrovate nei database 'Recon X' e 'HMDB' (Human Metabolome Database, <http://www.hmdb.ca/>).

Le liste scaricate da BiGG non contenevano le informazioni relative alle reazioni di scambio, perciò esse sono state inserite manualmente seguendo le seguenti assunzioni:

- Per tutti i metaboliti che compaiono nella cartella 'metabolites' è stata inserita una reazione di scambio nella forma:



dove *metab* è il metabolita di interesse e *c* è il compartimento a cui stiamo facendo riferimento. Affinchè il modello funzioni, deve comparire una reazione di scambio per ciascuno dei compartimenti in cui il metabolita è coinvolto. La doppia direzione della freccia indica che la reazione è irreversibile (il metabolita può quindi entrare ed uscire dalla rete).

- Ovviamente non dobbiamo permettere a tutti i metaboliti di entrare o uscire autonomamente dalla rete: quelli che vengono prodotti dalle reazioni della rete e che sono caratteristici del nostro modello non devono poter entrare nella rete in altro modo e, viceversa, quelli che vengono interamente utilizzati nella rete non devono avere la possibilità di uscire. Per far sì che ciò sia verificato è sufficiente settare a 0 i lower e/o gli upper bound delle reazioni di scambio interessate. Per decidere a quali metaboliti consentire

l'entrata e/o l'uscita, abbiamo fatto riferimento alla mappa del modello (in cui sono visualizzabili i metaboliti che partecipano a ciascuna reazione ma che non derivano dalla reazione che le precede e che vengono prodotti dalla reazione ma non vengono utilizzati da quella seguente) e alle equazioni esplicite delle reazioni (ai metaboliti che vengono solamente consumati occorre permettere di entrare nella rete, mentre quelli che vengono solamente prodotti devono poter uscire).

Per quanto riguarda i limiti inferiore e superiore di ciascun flusso di reazione, non essendo a conoscenza di ulteriori informazioni biologiche, si è assunto che:

- una reazione reversibile ha lower bound posto a  $-100$  ed upper bound posto a  $100$ ;
- una reazione irreversibile ha lower bound posto a  $0$  ed upper bound posto a  $100$ ;
- le reazioni di scambio seguono le regole sopra descritte.

Nell'esecuzione delle specifiche analisi, tali limiti possono essere modificati direttamente nel modello caricato in Matlab.

Poichè la funzione obiettivo cambia in base alle diverse analisi che si vogliono effettuare, di volta in volta i coefficienti di tale funzione sono stati modificati direttamente dal modello in Matlab, attribuendo dei valori specifici al vettore 'c' contenuto nella struttura del modello caricato.

### 6.2.3 Visualizzazione del modello completo

Come già accennato precedentemente, dal database BiGG è possibile scaricare la mappa del modello in formato '.txt', compatibile con gli ingressi accettati da COBRA Toolbox. Purtroppo, anche in questo caso, il sito non consente di scaricare solamente la mappa di interesse, bensì tutte le mappe relative ai pathway di metabolismo degli amminoacidi. Siamo stati perciò costretti ad utilizzare sempre questa mappa completa (e quindi più complessa) durante tutte le esecuzioni. Per poter poi riportare le immagini e i risultati relativi solamente ai pathway di interesse, ci siamo serviti del programma 'Inkscape', che è in grado di gestire i file in formato '.svg', restituiti dalle funzioni del Toolbox. In Figura 6.1 possiamo osservare la mappa del metabolismo del triptofano così ottenuta.

Per quanto riguarda il modello FBA ridotto (ottenuto a partire dalle informazioni ricavate dal modello cinetico), non è stato possibile costruire una mappa rappresentativa del pathway.

## 6.3 Analisi effettuate

Nonostante COBRA Toolbox offra una vasta disponibilità di strumenti, la nostra attenzione verrà posta prevalentemente su alcune analisi specifiche, che si sono rivelate essere di interesse. In particolare sono state effettuate le seguenti analisi:

1. analisi pFBA e conseguente classificazione dei geni, al fine di selezionare quali geni contribuiscano in modo più o meno significativo alle reazioni di interesse;
2. analisi FBA e FVA in due diverse situazioni, che verranno specificate meglio nel seguito: situazione normale e situazione di limitata produzione di melatonina. Tali analisi hanno lo scopo di rispondere alla domanda: "In situazione di limitata produzione di melatonina, si verifica un aumento nella produzione di acido quinolinico e/o acido xanturenico rispetto ad una situazione di normalità?";
3. analisi dei KnockOut che influenzano le reazioni di interesse. Tale analisi ha lo scopo di rispondere a due quesiti:
4. è possibile evitare un'eventuale sovrapproduzione degli acidi in situazione di sottoproduzione di melatonina?
5. quali geni è bene spegnere per ottenere una modifica delle reazioni di interesse ma, nello stesso tempo, evitare di causare altre ulteriori modifiche alle reazioni che non riguardano il nostro studio?



### 6.3.1 Reazioni di interesse

Per rispondere ai nostri quesiti dunque, focalizzeremo la nostra attenzione su alcune particolari reazioni del modello FBA completo:

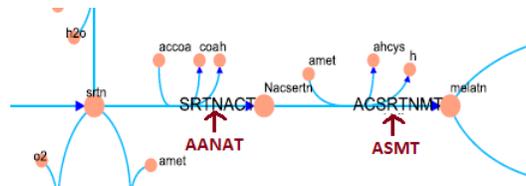
- SRTNACT e ACSRTNMT per la produzione di melatonina;
- QUILSYN per la produzione di acido quinolinico;
- 3HKYNAKGAT per la produzione di acido xanturenico.

L'osservazione di queste reazioni sarà sufficiente per poter cercare di dare una risposta ai nostri quesiti iniziali.

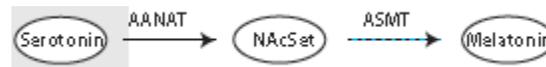
### 6.3.2 Simulazione della sottoproduzione di melatonina

Per rispondere ai nostri quesiti, risulta quindi fondamentale scegliere una metodologia per simulare nel modello una situazione di sottoproduzione di melatonina. Innanzitutto osserviamo le reazioni relative alla produzione di melatonina nei due modelli FBA analizzati:

- Modello completo)



- Modello ridotto)



In entrambi i casi quindi, per produrre melatonina a partire dalla serotonina sono necessari due step (due reazioni in serie). Ciascuna di queste reazioni è regolata da un gene in particolare: la prima dal gene AANAT, mentre la seconda dal gene ASMT.

La sottoproduzione di melatonina si può quindi simulare in due modi differenti:

- agendo sui geni che permettono le reazioni:



- agendo sui limiti massimi permessi (upper bound) dei flussi di reazione (il limite può essere posto, ad esempio, a 10 anziché a 100):



Poichè le reazioni coinvolte sono due, ci sono quindi quattro modalità possibili per simulare la situazione richiesta. Nel nostro caso, almeno inizialmente, abbiamo preferito agire sugli upper bound dei flussi delle reazioni, anche se nella realtà è stata scelta una modalità ibrida, in quanto sono state limitate tutte le reazioni del modello soggette alla regolazione da parte dello stesso gene.

Essential	Optima	ELE	MLE		
Aanat.1	Tph2.1	Ech1.1-x	Inmt.1	Abp1.1	Aox1.1
Ddc.1	Cyp1a1.1	Ehhadh.1-x	Acmsd.1	Gcdh.1-m	Acat1.1-m
Asmt.1	Cyp1a2.1	Aldh2.1-m	Dld/Pdhx.1-m	Gcdh.2-m	MaoA.1
	Cyp1b1.1	Aldh1b1.1-m	Dlst.1-m	Hadh2.1-m	MaoB.1
	Cyp2c19.1	Hsd17b4.1-x	Echs1.1-m	Hadha.1-m	Ogdh.1-m
	Indo.1	Acat2.1	Aldh1a1.1	Hadhb.1-m	Ogdh.2-m
	Tph1.1	Aldh4a1.1-m	Aldh9a1.1	Hadhsc.1-m	Aldh7a1.1
		Aldh4a1.2-m	Aldh3a2.1	Aoc2.1	Aadat.1
			Haa0.1	Aoc2.2	Aadat.2
			Auh.1-m	Kmo.1	Aldh1a2.2
			Tdo2.1	Ccb1.1	Aldh1a2.3
			Slc25a21.1-m	Aldh1a2.1	Kynu.1

Table 6.1: *Classificazione dei geni nel modello FBA completo ottenuto tramite pFBA.*

## 6.4 Risultati ottenuti dal modello FBA completo

Per eseguire le nostre analisi, uno degli aspetti più critici è stato quello di definire una funzione obiettivo, così come è richiesto dai diversi algoritmi utilizzati. Dopo aver eseguito diversi tentativi, abbiamo scelto di ottimizzare la funzione relativa al flusso che produce melatonina ('ACSRTNMT'), poichè si suppone che, in condizione di benessere, la cellula lavori al fine di produrre melatonina.

Poichè nella realtà non è però necessario che l'obiettivo biologico del metabolismo del triptofano sia proprio quello di massimizzare tale produzione, dobbiamo saper valutare i risultati con attenzione e con le dovute precauzioni, tenendo a mente che si tratta solamente di simulazioni basate su modelli matematici, che possono non rispecchiare appieno la realtà. In ogni caso, tale scelta dell'obiettivo è utile per rispondere alle domande relative al nostro studio, poichè è nostro scopo osservare cosa accade se, pur cercando di produrre melatonina, il sistema non è in grado di svolgere appieno questa funzione.

### 6.4.1 Analisi pFBA

Innanzitutto ricordiamo che la parsimonious FBA viene utilizzata per classificare i geni (e di conseguenza le reazioni ad essi connesse) in cinque classi differenti:

1. Essential: tali geni/reazioni sono essenziali al fine di poter ottimizzare la funzione obiettivo selezionata;
2. pFBA optima: classe che contiene i geni/le reazioni che permettono di ottenere contemporaneamente il valore ottimo massimo possibile della funzione obiettivo e la minima attività enzimatica possibile;
3. enzymatically less efficient (ELE), che richiedono più step enzimatici rispetto ad altri pathway alternativi al fine di ottenere lo stesso valore per la funzione obiettivo;
4. metabolically less efficient (MLE), che richiederebbero una diminuzione del valore obiettivo ottenuto, se venissero utilizzati;
5. pFBA no-flux, che non sono in grado di trasportare alcun flusso sotto le condizioni definite.

Tale analisi è stata eseguita sia nel modello normale, sia in quello soggetto a sottoproduzione di melatonina: i risultati ottenuti però sono coincidenti.

Nelle tabelle 6.1 e 6.2 possiamo osservare la classificazione dei geni e delle relative reazioni così ottenute. La Figura 6.2 rappresenta visivamente tale classificazione: questa rappresentazione può facilitare a comprendere (osservando la struttura del modello) i risultati ottenuti. Si può osservare infatti che le reazioni Essenziali sono tali per cui, se ne viene a mancare anche una soltanto, è impossibile ottimizzare la funzione richiesta, poichè non c'è alcun percorso che la connetta all'ingresso del triptofano nella cellula.

### 6.4.2 Analisi FBA ed FVA

Ponendo come funzione obiettivo la produzione di melatonina (ACSRTNMT), l'esecuzione della Flux Balance Analysis, nel caso di situazione normale (soggetto 'sano') e di sottoproduzione di melatonina (che da ora in

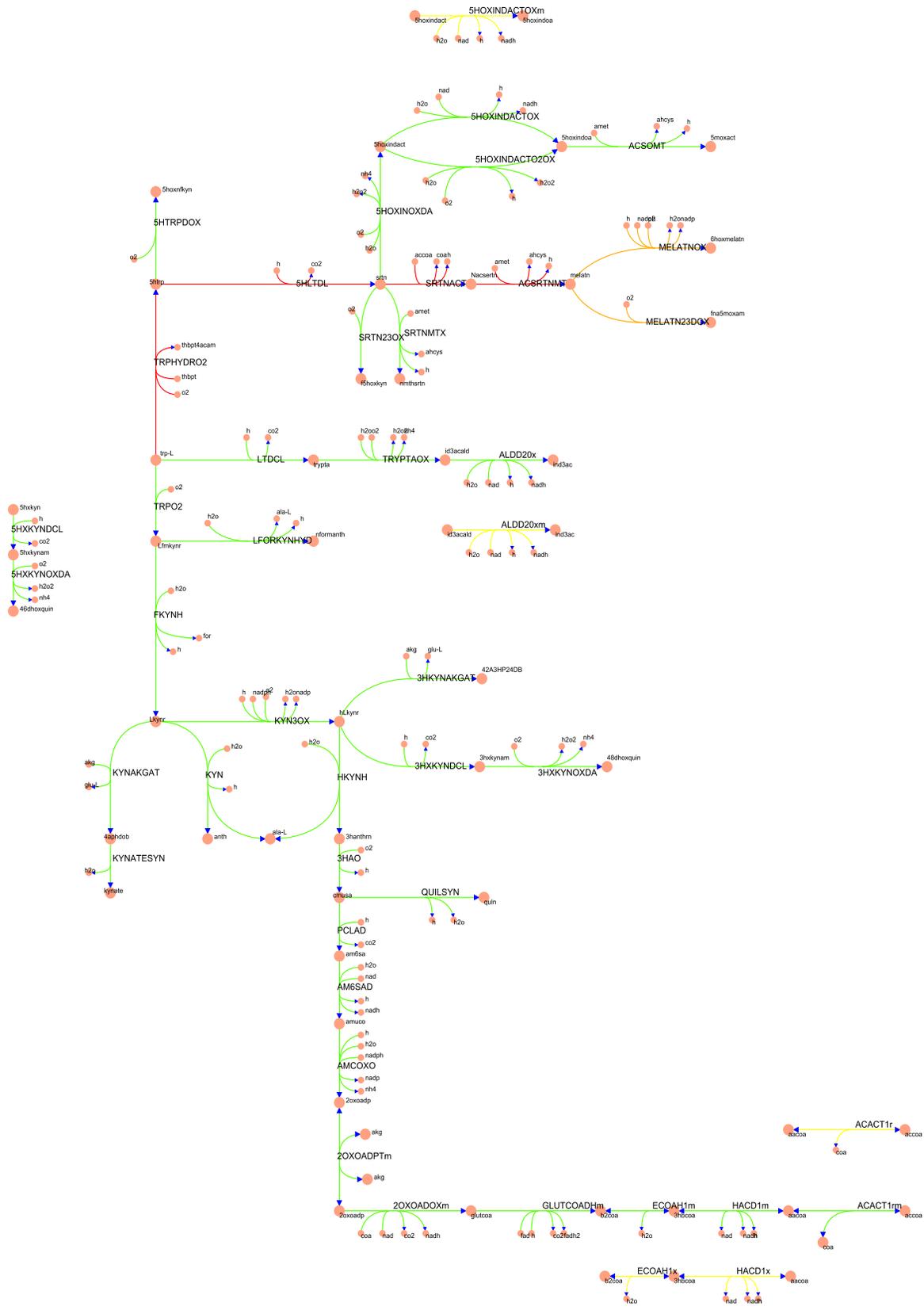


Figure 6.2: *Classificazione delle reazioni nel modello FBA completo ottenuto tramite pFBA. Rosso: Essential; arancione: pFBA optima; giallo: ELE, verde: MLE.*

Essential	Optima	ELE	MLE		
5HLTDL	MELATN23DOX	5HOXINDACTOX <sub>m</sub>	2OXOADOX <sub>m</sub>	ACACT1 <sub>rm</sub>	KYN3OX
ACSRTNMT	MELATNOX	ACACT1 <sub>r</sub>	2OXOADPT <sub>m</sub>	ACSOMT	KYNAKGAT
SRTNACT		ALDD20 <sub>xm</sub>	3HAO	ALDD20 <sub>x</sub>	KYNATESYN
TRPHYDRO2		ECOAH1 <sub>x</sub>	3HKYNAKGAT	AM6SAD	LFORKYNHYD
		HACD1 <sub>x</sub>	3HXKYNDCL	AMCOXO	LTDCL
			3HXKYNOXDA	ECOAH1 <sub>m</sub>	PCLAD
			5HOXINDACTO2OX	FKYNH	QUILSYN
			5HOXINOXDA	GLUTCOADH <sub>m</sub>	SRTN23OX
			5HTRPDOX	HACD1 <sub>m</sub>	SRTNMTX
			5HXKYNDCL	HKYNH	TRPO2
			5HXKYNOXDA	KYN	TRYPTAOX

Table 6.2: *Classificazione delle reazioni nel modello FBA completo ottenuto tramite pFBA.*

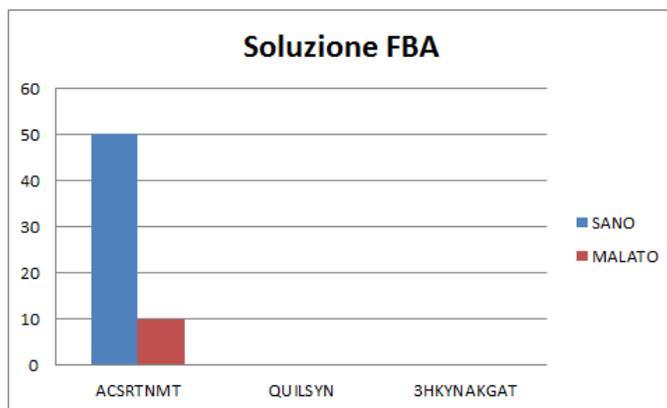


Figure 6.3: *Risultati di FBA in modello FBA completo relativo alle reazioni: ACSRTNMT (produzione di melatonina), QUILSYN (produzione di acido quinolinico) e 3HKYNAKGAT (produzione di acido xanturenico).*

poi definiremo come 'malato'), restituisce i valori dei flussi delle reazioni di interesse riportati in Figura 6.3, ovvero:

- ACSRTNMT (produzione di melatonina): 50 nel caso 'sano' e 10 (valore massimo possibile) nel caso 'malato';
- QUILSYN (produzione di acido quinolinico): 0 in entrambi i casi;
- 3HKYNAKGAT (produzione di acido xanturenico): 0 in entrambi i casi.

Poichè la Flux Balance Analysis restituisce solamente una delle possibili distribuzioni dei flussi tali da permettere l'ottimizzazione della funzione obiettivo, è interessante osservare anche i risultati della Flux Variability Analysis, che permette di conoscere l'intero span delle soluzioni ottime possibili, così da visualizzare se effettivamente, ottimizzando la melatonina, non si possa osservare alcun flusso attraverso le reazioni che producono acido quinolinico e xanturenico. In Figura 6.4 appare chiaro che, mentre nel caso 'sano' anche la FVA restituisce flussi nulli attraverso le reazioni che producono quinolinato ed acido xanturenico, nel caso 'malato', si verifica invece un possibile valore di flusso non nullo lungo entrambe le reazioni (in particolare nel caso dell'acido xanturenico). Ciò significa che, nel caso di sottoproduzione di melatonina, alcune soluzioni non azzerano i flussi di tali reazioni pur massimizzando, per quanto possibile, la produzione di melatonina.

### 6.4.3 Analisi dei KnockOut

Poichè l'analisi FVA ha rilevato che, in situazione di limitata produzione di melatonina, si può verificare un aumento, rispetto al caso normale, di produzione di acidi quinolinico e xanturenico, è interessante osservare

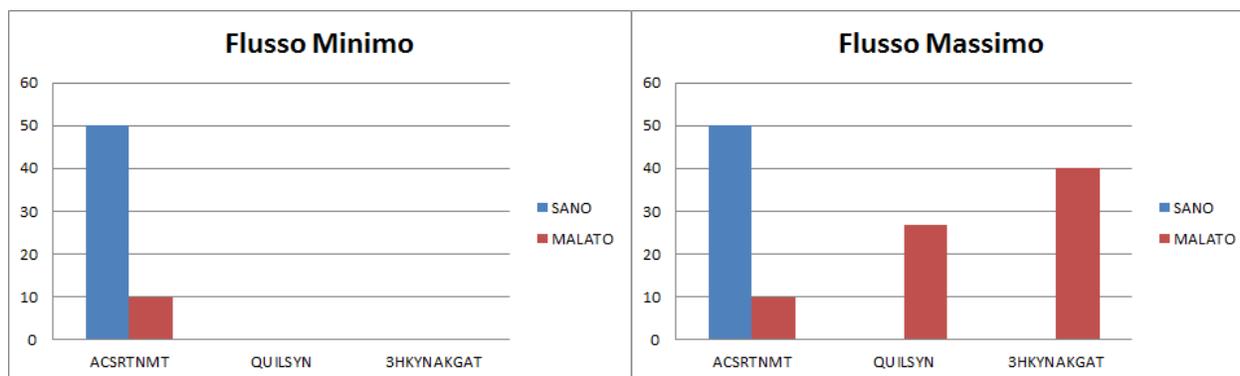


Figure 6.4: Risultati di FVA in modello FBA completo relativo alle reazioni: ACSRTNMT (produzione di melatonina), QUILSYN (produzione di acido quinolinico) e 3HKYNAKGAT (produzione di acido xanturenico). Sono rappresentati i valori massimo e minimo di ciascun flusso di reazione, massimizzando la produzione di melatonina.

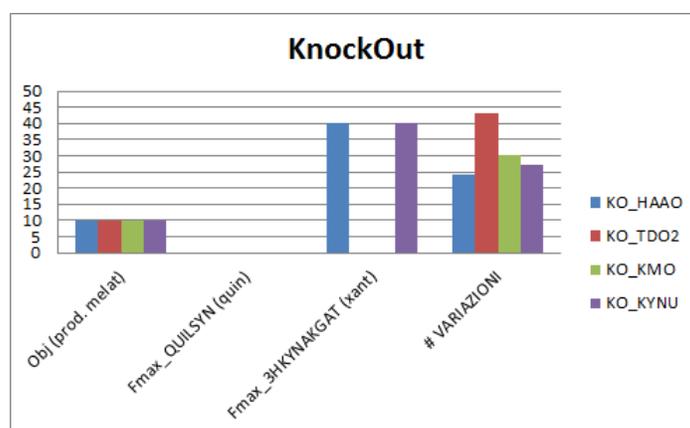


Figure 6.5: Risultati ottenuti attraverso il KnockOut di 4 geni differenti (HAAO, TDO2, KMO e KYNU) eseguito uno per volta.

a questo punto se il KnockOut di particolari geni (e il conseguente blocco delle reazioni ad essi collegate) può far sì che questo aumento non si verifichi più.

Basandoci sui risultati ottenuti mediante la pFBA, siamo stati in grado di selezionare i geni che influenzano i flussi di tali reazioni e di 'spegnerli' uno alla volta.

In questo caso si vuole osservare se il KnockOut del gene porta ad avere un flusso massimo nullo attraverso le reazioni QUILSYN e 3HKYNAKGAT come risultato della FVA anche nel caso 'malato', limitando però il numero totale di variazioni inerenti ad altre reazioni. Infatti il nostro scopo è quello di limitare le sovrapproduzioni degli acidi, ma non vogliamo modificare altre parti del modello, che non sono strettamente legate al nostro studio. A parità di capacità di annullare i flussi lungo le reazioni di interesse, verrà selezionata 'migliore' la scelta di porre in KnockOut il gene che produce il numero minimo di variazioni totali.

In Figura 6.5 possiamo visualizzare i risultati ottenuti con il modello FBA completo.

Come si può osservare dalla Figura 6.5, sia lo spegnimento del gene TDO2, sia quello del gene KMO, consentono di annullare contemporaneamente i flussi massimi ottenibili massimizzando la produzione di melatonina attraverso le reazioni che producono l'acido quinolinico e l'acido xanturenico. Tuttavia, porre in KnockOut il gene TDO2 comporta un numero di variazioni attraverso i flussi del modello molto elevato. Perciò, nonostante anche il KO del gene KMO comporti all'incirca 30 variazioni, esse sono in numero inferiore al caso TDO2 e perciò, da questi risultati, sembrerebbe che la scelta migliore per ottenere i nostri scopi sia quello di bloccare il gene KMO.

Il KO dei geni HAAO e KYNU invece porterebbe a dei benefici solamente inerenti all'acido quinolinico, quindi non rispondono ai nostri requisiti.

Essential	Optima	ELE	MLE		Essential	Optima	ELE	
Ddc	Tph1	Wars	Acmsd	Kynu	Ddc	Tph1	Wars	Kat.3
Aanat	Tph2		Afmid	MaoA	Aanat	Tph2	Acmsd	Kynu
			Cat	MaoB			Afmid	MaoA
			Hao	Nmnat			Cat	MaoB
			Il4i	Qprt			Hao	Nmnat
			Ido	Slc7a5			Il4i	Qprt
			Inmt	Slc7a8			Ido	Slc7a5
			Kat.1	Tdo2			Inmt	Slc7a8
			Kat.2	Kmo			Kat.1	Tdo2
			Kat.3				Kat.2	Kmo

Table 6.3: *Classificazione dei geni nel modello FBA ridotto nel caso 'sano' (sx) e nel caso 'malato' (dx) ottenuto tramite pFBA.*

## 6.5 Risultati ottenuti dal modello FBA ridotto

Anche nello studio del modello FBA ridotto abbiamo scelto come funzione obiettivo la reazione che produce melatonina: in questo caso AANAT\_Serotonin. Le precauzioni descritte relativamente ai risultati ottenuti tramite il modello FBA completo vanno quindi tenute a mente anche in questo secondo caso.

### 6.5.1 Analisi pFBA

Nel caso del modello FBA ridotto, l'analisi pFBA ha restituito dei risultati leggermenti differenti se applicata al modello 'sano' o 'malato'. In particolare, quei geni/reazioni che nel caso 'sano' venivano classificate come MLE (metabolically less efficient), nel caso di limitata produzione di melatonina, sono state classificate come ELE (enzymatically less efficient). Questo probabilmente si verifica perchè, nel caso del modello 'malato', il valore massimo ottenibile per la funzione obiettivo non è molto elevato e, l'utilizzo delle reazioni che nel caso 'sano' portavano ad una diminuzione del valore ottimo, non causa invece nel caso 'malato' tale riduzione.

La tabella 6.3 riporta la classificazione dei geni nel caso 'sano' e di limitata produzione di melatonina. Poichè in questo modello i nomi delle reazioni sono strettamente legati al nome dei geni che le regolano, riteniamo superfluo riportare anche la classificazione delle reazioni, poichè la classificazione dei geni è già esplicativa. Indichiamo solamente le informazioni relative alla produzione di acido quinolinico, poichè tale reazione è spontanea e quindi non vi è alcun gene che la regola: nel caso di modello 'sano', tale reazione si classifica come MLE, nel caso malato come ELE.

### 6.5.2 Analisi FBA ed FVA

Poichè nel modello FBA ridotto la reazione che produce acido xanturenico può essere regolata da uno qualsiasi dei tre geni KAT.1, KAT.3 e KAT.3, nelle tabelle riassuntive, osserveremo i flussi relativi alle reazioni regolate da ciascuno dei tre geni.

Anche in questo caso, massimizzando la produzione di melatonina, l'esecuzione della Flux Balance Analysis restituisce dei flussi nulli attraverso le reazioni che producono gli acidi sia nel caso normale, sia nel caso di limitata produzione di melatonina. In Figura 6.6 possiamo osservare i risultati, riassunti qui di seguito:

- AANAT\_Serotonin (produzione di melatonina): 100 nel caso 'sano' e 10 (valore massimo possibile) nel caso 'malato';
- Quinolinic Acid formation (produzione di acido quinolinico): 0 in entrambi i casi;
- KAT1/KAT2/KAT3\_3HKyn (produzione di acido xanturenico): 0 in entrambi i casi in tutte le reazioni.

Analogamente al caso del modello FBA completo, anche il modello ridotto, l'analisi FVA rivela invece dei possibili valori non nulli dei flussi relativi alla produzione degli acidi nel caso di limitata produzione di melatonina. In Figura 6.7 possiamo visualizzare i risultati ottenuti tramite questa analisi.

Ancora una volta sembra essere l'acido xanturenico a subire l'aumento più elevato.

È quindi interessante osservare cosa può cambiare effettuando dei KnockOut anche in questo secondo modello.

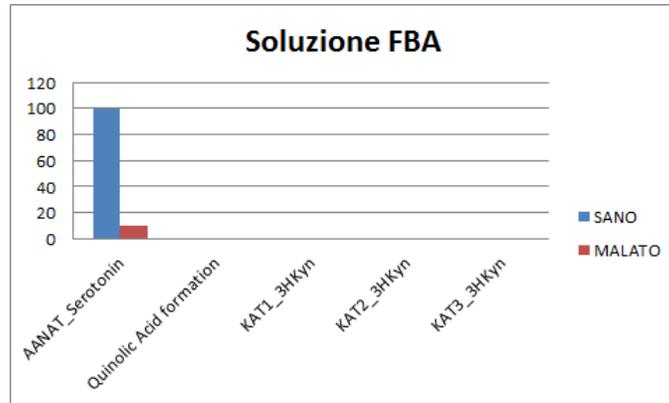


Figure 6.6: Risultati di FBA in modello FBA ridotto relativo alle reazioni: AANAT\_Serotonin (produzione di melatonina), Quinolinic Acid formation (produzione di acido quinolinico) e KAT1/KAT2/KAT3\_3HKyn (produzione di acido xanturenico).

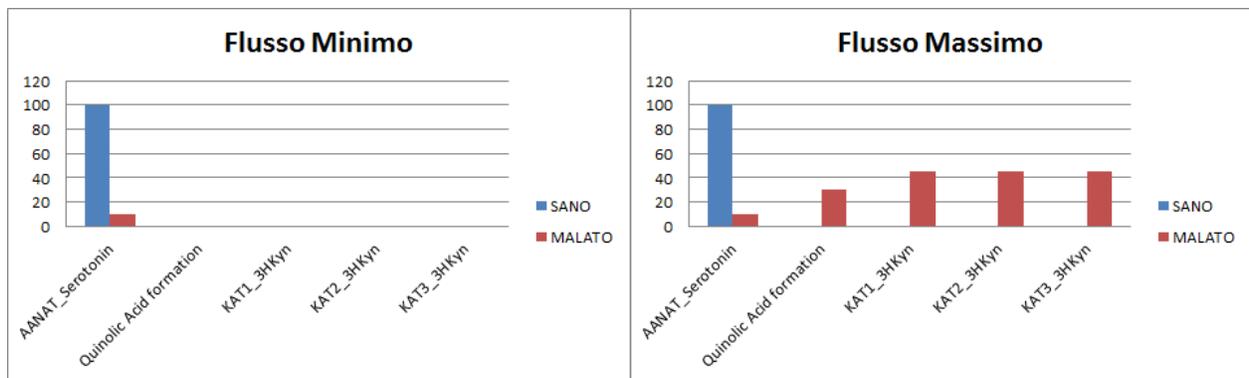


Figure 6.7: Risultati di FVA in modello FBA ridotto relativo alle reazioni: AANAT\_Serotonin (produzione di melatonina), Quinolinic Acid formation (produzione di acido quinolinico) e KAT1/KAT2/KAT3\_3HKyn (produzione di acido xanturenico). Sono rappresentati i valori massimo e minimo di ciascun flusso di reazione, massimizzando la produzione di melatonina.

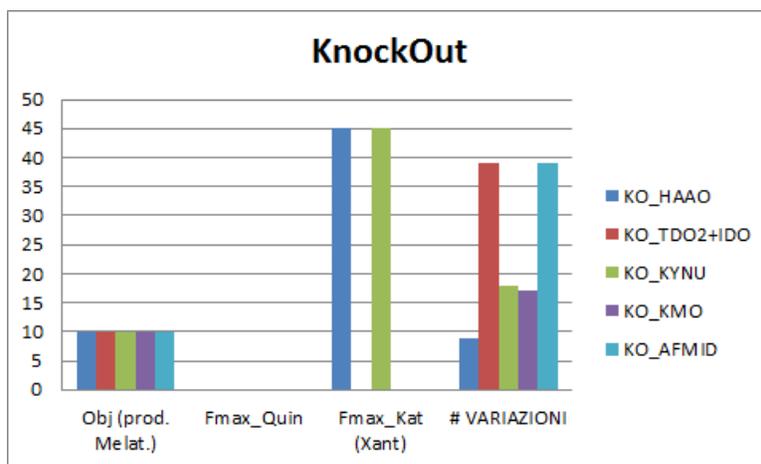


Figure 6.8: Risultati ottenuti attraverso il KnockOut di 4 geni differenti (HAAO, TDO2 e IDO, KYNU, KMO e AFMID) eseguito, eccetto nel caso TDO2+IDO, uno per volta.

### 6.5.3 Analisi dei KnockOut

Poichè nel modello FBA ridotto i geni TDO2 e IDO regolano la stessa funzione, è stato analizzato anche il comportamento del modello 'malato' nel caso in cui entrambi i geni venissero posti in KnockOut contemporaneamente.

La Figura 6.8 riporta i risultati ottenuti bloccando rispettivamente i geni HAAO, TDO2 e IDO, KYNU, KMO e AFMID, poichè si è precedentemente verificato che tali geni influenzano le reazioni di interesse.

Si osservi che in tutti i casi analizzati, il valore della funzione obiettivo non subisce variazioni.

Anche in questo caso vogliamo osservare quali KO ci permettono di eliminare i flussi attraverso le reazioni che producono gli acidi, modificando il minor numero di reazioni nel modello. Si può facilmente notare che lo spegnimento dei geni HAAO e KYNU non porta alcun beneficio per quanto riguarda l'acido xanturenico. Inerentemente agli altri KO, si osserva che i flussi vengono eliminati ma l'unico che non comporta un numero troppo elevato di variazioni sembra essere il KO del gene KMO. Tale risultato rispecchia quello ottenuto tramite l'analisi del modello FBA completo.

## 6.6 Integrazione dei risultati ottenuti tramite modello cinetico e modello FBA ridotto

Come abbiamo osservato in precedenza, il modello FBA ridotto utilizzato, pur essendo meno completo rispetto al modello FBA completo, ha il vantaggio di essere in un certo senso comparabile con il modello cinetico, in quanto è una rappresentazione strutturalmente diversa dello stesso identico modello.

Tramite il modello FBA ridotto abbiamo potuto osservare quali KnockOut possono portare dei benefici nel caso di pazienti diabetici, causando la minima interferenza con le altre reazioni del metabolismo, che non riguardano i prodotti di interesse per l'analisi. I risultati ottenuti possono essere quindi analizzati più approfonditamente simulando tali KnockOut nel modello cinetico ottenuto con i dati reali di pazienti soggetti alla malattia del diabete (modello relativo al muscolo).

Il nostro scopo è ora quindi quello di verificare se effettivamente si notano gli stessi risultati anche nel modello cinetico, più vicino ad una rappresentazione reale dell'attività biologica.

Dall'analisi del modello FBA ridotto (che ha restituito risultati simili al modello FBA completo), sembra che il KnockOut più utile possa essere quello del gene KMO.

L'applicazione di tale KO nel modello reattivo ai pazienti diabetici (osservato in più situazioni possibili), non restituisce esattamente il risultato sperato. La Figura 6.9 infatti rappresenta le differenze percentuali  $\Delta$  di flussi relativi alle reazioni di interesse tra paziente malato (prima nel modello 'wild-type' (privo di KO) e poi nel modello in cui è stato bloccato il gene KMO) e sano. Quello che si osserva è che grazie al KO, i flussi inerenti alla produzione degli acidi vengono bloccati totalmente. Poichè nel caso dei pazienti sani tali flussi non sono esattamente nulli, il nostro scopo non dev'essere quello di azzerarli, quanto piuttosto quello di eliminare o ridurre il più possibile la differenza rispetto al caso di paziente sano. Inoltre, tramite questo

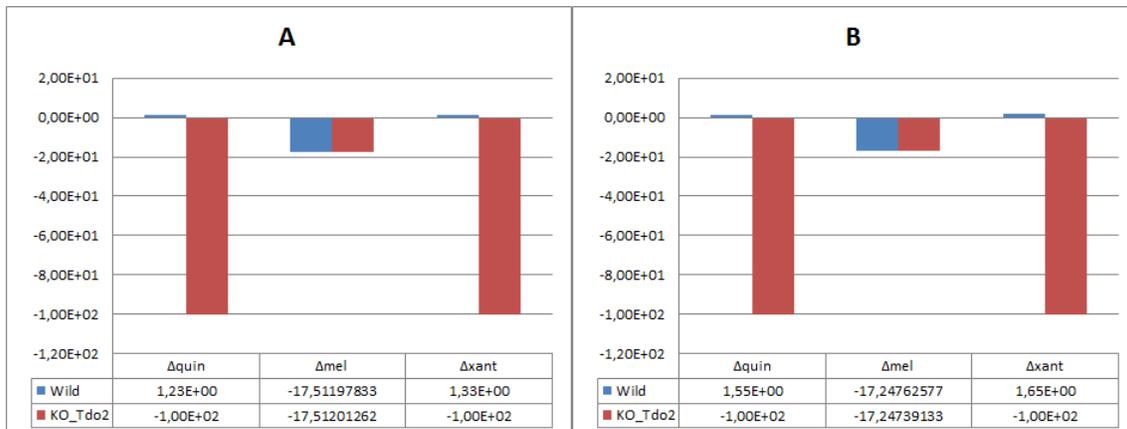


Figure 6.9: Risultati ottenuti con il KO del gene *KMO* nel modello cinetico (nel muscolo) dei pazienti diabetici.  $\Delta$  = differenza percentuale dei diabetici rispetto ai pazienti sani, nel caso di modello wild-type e con KO del gene *KMO* applicato al modello di paziente malato.

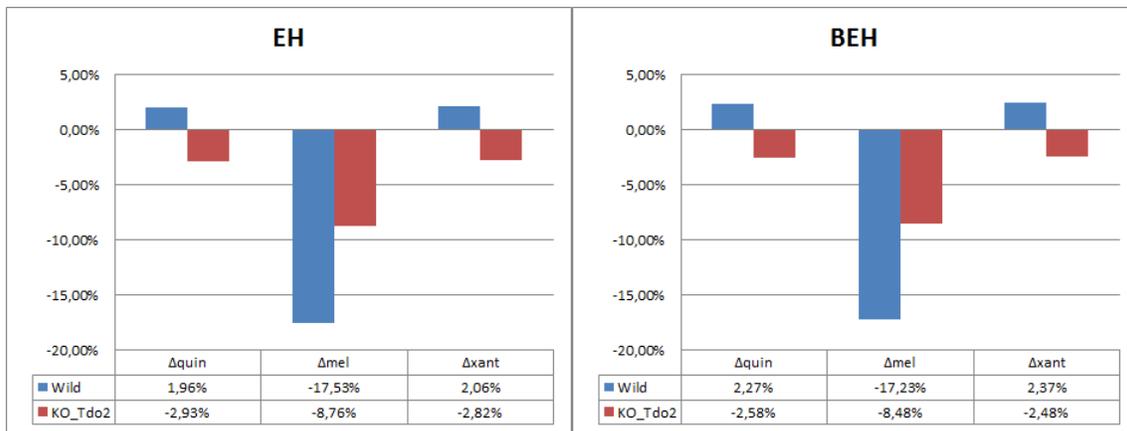


Figure 6.10: Risultati ottenuti con il KO del gene *TDO2* nel modello cinetico (nel muscolo) dei pazienti diabetici.  $\Delta$  = differenza percentuale dei diabetici rispetto ai pazienti sani, nel caso di modello wild-type e con KO del gene *TDO2* applicato al modello di paziente malato.

KO, non ci sono variazioni sensibili nella produzione di melatonina (questo fatto era prevedibile dall'analisi di sensitività fatta in precedenza: dai risultati infatti si poteva già osservare che il gene *KMO* non ha alcuna influenza relativamente al flusso di produzione di melatonina).

Dopo numerosi esperimenti, si è verificato che la soluzione migliore sembra essere quella di spegnere il gene *TDO2*. Come si può osservare in Figura 6.10, che riporta alcuni esempi di soluzioni possibili, tale KO effettuato nel modello dei pazienti diabetici:

1. diminuisce la sovrapproduzione di acido quinolinico;
2. diminuisce la sovrapproduzione di acido xanturenico;
3. diminuisce la sottoproduzione di melatonina.

In questo modo fa sì che il comportamento del metabolismo dei pazienti diabetici si avvicini al comportamento osservato nei pazienti sani. Occorre tuttavia osservare che non sempre l'effetto del KO è quello di ridurre la differenza tra i flussi nel paziente malato e sano: talvolta tale differenza percentuale può aumentare. Nonostante ciò, nella maggior parte dei casi in cui si verifica questo, il KO porta ad una sovrapproduzione di melatonina e ad una sottoproduzione degli acidi nel caso dei pazienti diabetici rispetto ai sani. Questa situazione ottenuta, pur non essendo quella sperata, è meno dannosa di quella di partenza.

## Chapter 7

# Conclusioni

Per arrivare alle conclusioni della tesi, è interessante ripercorrere tutta la storia inerente alla sua stesura, poichè essa è significativa nella comprensione di quali siano i risultati e gli obiettivi raggiunti.

Inizialmente, incuriositi da uno strumento di analisi di recente sviluppo (la Flux Balance Analysis), abbiamo pensato di approfondirne la conoscenza, al fine di studiare il comportamento del metabolismo del triptofano nell'essere umano. Poichè tale amminoacido viene spesso utilizzato come antidepressivo per i pazienti diabetici, la sua influenza nell'organismo inerente ad alcuni importanti metaboliti (quali la serotonina, la melatonina e gli acidi quinolinico e xanturenico) risulta essere di grande interesse. In particolare, come abbiamo riportato nell'introduzione della tesi, era nostro obiettivo modellizzare il pathway del metabolismo del triptofano, per osservare:

- se una diminuzione di produzione di serotonina e melatonina può portare ad un aumento della produzione degli acidi quinolinico e xanturenico;
- se nei pazienti diabetici si osserva effettivamente una sottoproduzione di melatonina e una sovrapproduzione degli acidi;
- nell'eventualità del verificarsi dei casi precedenti, se e come si può agire a livello genico per evitare tali eventuali sovrapproduzioni e cercare di aumentare il livello di melatonina nel sangue, così da riportarsi ad una situazione di normalità.

L'essere umano però è un organismo molto complesso e di cui si ha una conoscenza ancora solamente parziale. Riuscire a realizzare un modello cinetico del metabolismo del triptofano nell'uomo, completo di tutti i parametri cinetici necessari, sembrava quindi impossibile. Perciò questa nuova metodologia di modellizzazione, basata solamente sulla conoscenza dei vincoli stechiometrici, dettati dalle equazioni di bilanciamento di massa delle reazioni, ci è sembrata un'ottima opportunità per poter effettuare diversi studi.

La FBA però viene solitamente utilizzata nello studio di organismi unicellulari, i quali, in stato di benessere, cercano di massimizzare la biomassa. Per tali organismi quindi viene naturale scegliere come funzione obiettivo la reazione rappresentante la biomassa. È questo l'ambito in cui la Dott.ssa Costanza e il Prof. Nicosia ci hanno introdotti a questa nuova tecnica di analisi.

Un altro ambito nel quale viene utilizzata largamente la Flux Balance Analysis è la farmacocinetica, allo scopo di selezionare quali farmaci è bene utilizzare per attaccare determinati target senza alterare altre funzionalità dell'essere umano.

Discostandoci da questi ambiti quindi eravamo interessati anche a capire se la FBA potesse in qualche modo aiutarci anche nei nostri studi, in cui si analizzano cellule di un organismo molto complesso (l'uomo) in situazione di salute e di malattia del diabete.

Inizialmente quindi la tesi era volta ad uno studio teorico di questa nuova metodologia di analisi e alla ricerca di modelli realizzabili e utilizzabili inerenti al metabolismo del triptofano nelle cellule umane. L'interesse è stato rivolto anche agli strumenti utilizzabili a tale scopo, in particolare COBRA Toolbox, utilizzabile in ambiente Matlab. Tramite l'utilizzo di questo software, abbiamo quindi realizzato un modello FBA completo del pathway del metabolismo del triptofano nell'essere umano, in formato Excel.

Il 15 Ottobre 2013 però è stato pubblicato online un nuovo articolo [2] relativo al modello cinetico del metabolismo del triptofano nell'uomo. Ovviamente tale modello non può essere completo e si basa su alcune ipotesi e condizioni iniziali che non rispecchiano necessariamente la realtà; tuttavia esso è uno strumento molto utile come punto di partenza per ottenere dei primi risultati spendibili in analisi successive.

A partire da questo modello quindi il nostro interesse si è rivolto anche ad un'analisi approfondita della sua costruzione e di strumenti necessari per la sua analisi (a tale scopo ci siamo serviti del software COPASI). Poiché il triptofano viene metabolizzato in tutte le cellule del nostro organismo, tale modello è tessuto-dipendente e deve quindi contenere parametri differenti a seconda del tessuto che si vuole analizzare. Gli autori del paper presentano i parametri relativi al fegato e al cervello in pazienti sani. Da database disponibili online, siamo riusciti a reperire inoltre dati relativi al muscolo scheletrico di pazienti sani e di pazienti soggetti al diabete di tipo due. Sarebbe stato ancora più interessante poter analizzare il modello inerente all'intestino, dove viene secreta la maggior parte di serotonina nel nostro organismo. Purtroppo non è stato possibile reperire questi dati. Tale modello e tali strumenti ci hanno permesso di effettuare analisi nel tempo dei flussi relativi che metabolizzano il triptofano nella cellula e analisi di sensitività inerenti alle reazioni di nostro interesse, ovvero la produzione di melatonina (e quindi serotonina) e di acidi quinolinico e xanturenico. Ricordiamo che con flusso relativo si intende la percentuale di triptofano entrante nella cellula che viene metabolizzato da ciascuna reazione: esso è quindi indice di come si distribuisca il triptofano nella rete metabolica, determinando una conseguente diversa concentrazione dei metaboliti prodotti. L'analisi di sensitività invece ci ha aiutati ad identificare quali siano i geni più influenti in queste reazioni. La tessuto-dipendenza dei modelli ci ha permesso inoltre di confrontare i risultati ottenuti in tessuti differenti e in situazione di malattia o salute.

Dalle informazioni stechiometriche reperibili dal modello cinetico (ovvero dalle equazioni di bilanciamento di massa che descrivono le reazioni coinvolte nel pathway, ottenibili tramite l'utilizzo del software Copasi), è stato possibile realizzare in formato Excel anche un ulteriore modello FBA (ridotto). Questo modello FBA, analizzabile quindi tramite gli strumenti messi a disposizione dalle tecniche inerenti alla Flux Balance Analysis (uso di COBRA Toolbox) può essere considerato, in un certo senso, comparabile al modello cinetico di partenza, poiché esso è una diversa rappresentazione della stessa rete: il modello cinetico ne fa una descrizione tramite equazioni differenziali, mentre il modello FBA ridotto la descrive da un punto di vista strutturale, basandosi sui vincoli dettati dal bilanciamento di massa.

Utilizzando quindi questo modello FBA così realizzato e quello ottenuto in precedenza, quando ancora non eravamo a conoscenza dell'esistenza di un modello cinetico utile ai nostri scopi, abbiamo analizzato il metabolismo del triptofano anche con gli strumenti inerenti alla Flux Balance Analysis. In questo caso abbiamo simulato una situazione di malattia limitando la possibilità della cellula di produrre melatonina. Tramite queste analisi abbiamo potuto classificare i geni e le reazioni del pathway in situazione di ottimizzazione di produzione di melatonina nel caso di normalità o di simulata malattia. Inoltre abbiamo osservato i risultati ottenuti tramite FBA e la sua variante (più significativa) FVA (Flux Variability Analysis), che ci hanno rivelato come una limitata possibilità di produzione di melatonina possa produrre effettivamente un aumento della produzione degli acidi durante la metabolizzazione dell'amminoacido.

Allo scopo di riportare il comportamento del modello 'malato' ad andamenti simili a quelli del modello 'sano', si sono osservati i risultati di FBA e FVA ottenuti in seguito al KnockOut (KO) di diversi geni, basandosi sulle precedenti classificazioni. Dopo aver individuato quali KO potessero essere più utili per ristabilire una situazione di normalità, la nostra attenzione si è focalizzata nuovamente sul modello cinetico inerente ai dati reali di pazienti sani e diabetici, dove abbiamo potuto simulare effettivamente il comportamento del modello relativo ai diabetici in situazione di KO dei geni selezionati.

In seguito a tutte queste diverse analisi, possiamo riassumere in breve i seguenti risultati ottenuti tramite le simulazioni:

- nei pazienti diabetici si osserva effettivamente una sottoproduzione di melatonina e serotonina e una leggera sovrapproduzione di acidi quinolinico e xanturenico;
- una situazione di limitata possibilità di produzione di melatonina può comportare una sovrapproduzione di acidi quinolinico e xanturenico. Questa affermazione va interpretata come risultato ottenuto tramite l'analisi FVA effettuata nei modelli FBA. Supponendo che lo scopo sia l'ottimizzazione del flusso di produzione di melatonina, la Flux Variability Analysis (che analizza l'intero span delle distribuzioni di flussi ottime possibili), evidenzia che, in situazione di normalità, il valore massimo dei flussi relativi alla produzione degli acidi è nulla. Se invece si pone un limite nell'upper bound della produzione di melatonina (situazione utilizzata per simulare una sottoproduzione di melatonina), la stessa analisi mostra dei flussi massimi non nulli relativi alla produzione di entrambi gli acidi.
- il KnockOut di alcuni geni può favorire il ristabilirsi di una situazione simile a quella di un paziente sano anche nel caso di organismo diabetico, limitando il più possibile il numero di variazioni causate da tali KO non inerenti agli scopi dello studio. In una prima analisi, i geni KMO e TDO2 sembravano

entrambi dei validi candidati per essere bloccati; tuttavia, un'analisi più approfondita, che ha permesso di sfruttare entrambe le tipologie di modello utilizzate, ha rivelato che il KO del gene KMO può causare un'eliminazione completa della produzione degli acidi. Poiché ciò che si vuole ottenere non è la scomparsa di tali acidi nell'organismo, bensì una situazione simile a quella dei soggetti sani, i risultati ottenuti hanno identificato il gene TDO2 come miglior candidato ad essere posto in KnockOut.

Osservando tali risultati possiamo dirci soddisfatti: sono stati raggiunti gli obiettivi che ci eravamo posti in origine, infatti siamo stati in grado di modellizzare il pathway del metabolismo del triptofano mediante l'utilizzo di diverse tipologie di strumenti, al fine di rispondere alle nostre domande iniziali. Inoltre questo elaborato comprende una trattazione esauriente della nuova metodologia di analisi FBA, che si è rivelata essere un utile strumento di partenza anche nello studio di organismi più complessi, come l'essere umano.

# Appendix A

## Descrizione dei tasks del software COPASI

Il software COPASI permette di eseguire diverse analisi utilizzando un modello cinetico. Per questo motivo è stato utilizzato dagli autori del paper [2] per analizzare il modello che hanno realizzato. Al fine di perseguire i nostri scopi, abbiamo utilizzato anche noi questi strumenti; è bene perciò analizzare quali siano le funzioni offerte da tale software, per poter capire in modo completo i risultati presentati.

### A.1 Steady-State Analysis

Il programma permette di effettuare, a partire dalle condizioni iniziali fissate in precedenza dall'utente, un'analisi in stato stazionario, che corrisponde allo stato in cui le variabili di stato del modello (ad esempio le concentrazioni dei diversi metaboliti) non cambiano nel tempo: questo si esprime ponendo uguali a zero tutte le equazioni differenziali, che descrivono l'andamento temporale del sistema metabolico. Il sistema risultante è quindi un sistema di equazioni algebriche non lineari, che COPASI risolve attraverso più metodi numerici, basandosi sul numero di particelle e sul rate del numero di particelle, anziché sulle concentrazioni. Inoltre il programma analizza lo stato stazionario utilizzando il modello ridotto.

Quando si esegue tale analisi, è possibile settare diversi parametri:

*Use Newton:* valore booleano, che indica se utilizzare il metodo 'damped' di Newton (1, default) nella risoluzione del sistema non lineare, utilizzando le condizioni iniziali definite dall'utente. Tale metodo è una variante del metodo classico di Newton, in cui la soluzione viene ottenuta attraverso una procedura iterativa che raffina una condizione iniziale fino a che l'errore residuo non diventa minore di quanto richiesto. L'algoritmo può terminare anche senza il raggiungimento di una soluzione, nel caso sia settato un numero massimo di iterazioni. Il metodo originale applica sempre il seguente aggiornamento durante la procedura iterativa:

$$x_i = x_{i-1} - \frac{f(x_{i-1})}{f'(x_{i-1})},$$

mentre nella versione 'damped', se  $x_{i-1}$  ha un errore residuo maggiore di  $x_i$ , si valuta:

$$x_i = x_{i-1} - \frac{f(x_{i-1})}{f'(x_{i-1})} \cdot 2^{-n}, \quad \text{dove } n = 0, \dots, 32$$

e si accetta il primo valore che ha errore residuo minore di  $x_i$ . Se tale valore non viene trovato, la procedura termina senza una soluzione (perché ci si trova in un minimo locale).

*Use Integration:* tale parametro booleano, se settato ad 1, impone di utilizzare un risolutore deterministico delle equazioni differenziali nel tempo, fino a che non viene raggiunto uno stato stazionario. Se alla  $10^{10}$  unità temporale non è ancora stata raggiunta una condizione stazionaria, il metodo termina senza una soluzione.

*Use Back Integration:* tale parametro booleano, se settato ad 1, impone di utilizzare un risolutore deterministico delle equazioni differenziali ribaltando l'asse temporale, ossia andando indietro nel tempo,

fino a che non viene raggiunto lo stato stazionario. Se alla  $10^{10}$  unità temporale non è ancora stato raggiunta una condizione stazionaria, il metodo termina senza una soluzione.

*Accept Negative Concentrations:* questo parametro booleano indica se accettare uno stato stazionario contenente delle concentrazioni negative (1) oppure se non accettare alcuna concentrazione minore di zero (0, default).

*Iteration Limit:* questo parametro è un numero intero positivo, utilizzato per determinare il numero massimo di iterazioni eseguite dal metodo di Newton. Di default vale 50.

*Derivation Factor:* tale parametro numerico indica la dimensione del passo utilizzato per calcolare  $f'(x_{i-1})$ . Di default vale 0.001.

*Resolution:* è un parametro numerico positivo utilizzato per determinare la risoluzione con cui decidere se lo stato corrente è accettabile o meno come stato stazionario. Se la differenza (in valore assoluto) di ogni variabile di stato, rispetto allo stato precedente, è minore di tale valore, lo stato corrente viene accettato come soluzione. Di default vale  $10^{-9}$ .

L'esecuzione di tale analisi, restituisce diverse informazioni inerenti le concentrazioni e i rate di transizione dei metaboliti, i flussi delle reazioni, la matrici Jacobiane completa e ridotta, informazioni sulla stabilità del sistema e sul protocollo utilizzato. Inoltre possono essere fornite anche informazioni sui diversi compartimenti e sui parametri del modello.

## A.2 Stoichiometric State Analysis

### A.2.1 Elementary Flux Modes

Tale funzione permette di visualizzare una lista di tutte le reazioni del modello assieme alla propria equazione chimica e alla condizione di reversibilità o irreversibilità.

### A.2.2 Mass Conservations

Questa funzione permette di analizzare il modello dal punto di vista stechiometrico: restituisce infatti la matrice stechiometrica del sistema completo e della rete ridotta (il metodo di riduzione utilizzato da COPASI è descritto in R. R. Vallabhajosyula, V. Chickarmane, and H. M. Sauro<sup>1</sup>).

## A.3 Time Course Analysis

Attraverso delle simulazioni che tengono conto dello scorrere del tempo, il programma permette di calcolare le traiettorie delle concentrazioni dei metaboliti presenti nel modello in un dato intervallo di tempo. Il software mette a disposizione diversi metodi per effettuare tale analisi, che l'utente può scegliere in base alle finalità dei suoi studi.

Il metodo che COPASI implementa di default è un metodo deterministico LSODA<sup>2</sup>; per sistemi con un numero limitato di compartimenti però può essere più utile utilizzare uno dei metodi stocastici che il programma mette a disposizione, che si basano sul metodo descritto da Gibson e Bruck<sup>3</sup>.

Poiché le simulazioni deterministiche sono inappropriate per alcuni sistemi, ma, allo stesso tempo quelle stocastiche possono impiegare molto più tempo, COPASI offre alcuni metodi ibridi, che combinano i vantaggi di ciascuna delle due tipologie di simulazione.

---

<sup>1</sup>R. R. Vallabhajosyula, V. Chickarmane, and H. M. Sauro. "Conservation analysis of large biochemical networks". *Bioinformatics*. 22(3). 346 - 353. 2006

<sup>2</sup>L. Petzold. "Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations". *SIAM J. Sci. Stat. Comput.* 4. 136 - 148. 1983

<sup>3</sup>M.A. Gibson and J. Bruck. "Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels". *J. Phys. Chem.* A104(9). 1876-1889. 2000

### A.3.1 Deterministic Simulation (LSODA)

Il metodo deterministico LSODA, che COPASI implementa di default per le simulazioni temporali, risolve i sistemi  $\frac{dv}{dt} = f(t, y)$  utilizzando la matrice Jacobiana se il sistema è rigido, ma selezionando automaticamente tra metodi rigidi (BDF) e non rigidi (Adams). Inizialmente utilizza un metodo non rigido e in seguito monitora i dati per decidere quale metodo utilizzare.

Il metodo permette di selezionare diverse opzioni:

*Integrate Reduced Model:* questo parametro booleano, se posto ad 1 (default) determina l'utilizzo dell'integrazione utilizzando le leggi di conservazioni di massa. In altre parole l'opzione permette di scegliere se utilizzare un modello ridotto (e quindi le leggi di conservazioni di massa) oppure il modello completo (determinando tutte le variabili tramite le equazioni differenziali) per effettuare l'integrazione.

*Relative Tolerance:* parametro numerico che specifica la tolleranza relativa che l'utente vuole raggiungere (un valore piccolo indica che le traiettorie vengono calcolate in modo molto accurato). Di default vale  $1.0 \cdot 10^{-6}$ .

*Absolute Tolerance:* parametro numerico positivo che indica la tolleranza assoluta (applicata alle concentrazioni) che l'utente vuole raggiungere. Di default vale  $1.0 \cdot 10^{-12}$ .

*Adams Max Order:* parametro intero positivo che indica il massimo ordine che deve raggiungere il metodo non rigido Adams prima che il programma cambi metodo ed inizi ad utilizzare il metodo rigido BDF. Di default vale 12.

*BDF Max Order:* parametro intero positivo che indica l'ordine massimo che il metodo BDF deve raggiungere prima di diminuire le dimensioni degli step interni. Di default vale 5.

*Max Internal Steps:* parametro intero positivo che indica il numero massimo di step interni che l'integratore può effettuare prima dell'unità di tempo successiva. Di default vale 10000.

### A.3.2 Stochastic Simulation: The Next-Reaction-Method

Questo metodo di simulazione stocastica utilizza l'algoritmo sviluppato da Gibson e Bruck, secondo il quale per ciascuna reazione viene calcolato un tempo stocastico presunto e viene realizzata la reazione il cui tempo calcolato risulta essere il più piccolo. L'insieme delle reazioni viene organizzato in una priority queue, così da permettere una ricerca efficiente delle reazioni più veloci. Inoltre, ad ogni passo, vengono ricalcolati i tempi stocastici solo per quelle reazioni che sono dipendenti da quella che è stata realizzata. Tale metodo richiede però che tutte le reazioni del modello siano irreversibili: in ogni caso, COPASI mette a disposizione uno strumento per rendere irreversibili tutte le reazioni che non lo sono.

Poiché l'algoritmo lavora internamente utilizzando il numero di particelle anziché le concentrazioni, tale numero non deve eccedere un valore di circa 264, affinché il metodo possa essere implementato. È questo uno dei motivi per cui le simulazioni stocastiche possono avere luogo solo in sistemi con un ridotto numero di compartimenti.

Tale implementazione del Next-Reaction-Method è però piuttosto inefficiente se il modello contiene delle regole assegnate, che allungano di molto i tempi computazionali.

Inoltre occorre considerare che ci sono delle restrizioni inerenti le 'quantità globali': se una quantità globale è di tipo "ode" (ossia se ad un'equazione differenziale è assegnata una quantità globale), tale versione del programma non è in grado di effettuare una simulazione stocastica; se la quantità globale è di tipo "assignment", tale simulazione è possibile ma non molto efficiente. Non ci sono invece restrizioni per quanto riguarda le quantità globali di tipo "fixed":

Le opzioni di tale metodo sono le seguenti:

*Max Internal Step:* parametro intero positivo che indica il numero massimo di step interni che l'integratore può effettuare prima dell'unità di tempo successiva. Di default vale 10000000.

*Subtype:* parametro ignorato da questa versione di COPASI.

*Use Random Seed:* di default tale parametro booleano vale 0 e significa che il 'random seed' è settato ad un valore random prima di ogni esecuzione: di conseguenza esecuzioni diverse possono restituire traiettorie differenti. Se impostato ad 1, il parametro indica che il 'random seed' viene settato al valore impostato dall'utente.

*Random Seed*: valore intero privo di segno utilizzato come 'random seed' nell'esecuzione (viene utilizzato solo se il parametro precedente è stato settato ad 1). Di default vale 1.

### A.3.3 Hybrid Simulation: Runge-Kutta

Questo metodo di simulazione ibrida combina un'integrazione numerica deterministica delle equazioni differenziali con un algoritmo di simulazione stocastica. Internamente, il sistema viene partizionato in un sottosistema deterministico e in un sottosistema stocastico: quello deterministico contiene tutte le reazioni a cui prendono parte tutti i compartimenti con un alto numero di particelle; le reazioni che comprendono solo compartimenti con un numero limitato di particelle appartengono al sottosistema stocastico. La partizione delle rete biochimica può cambiare dinamicamente durante la simulazione: dopo un determinato intervallo di tempo, viene ridefinita la partizione in base al numero di particelle corrente. Durante un'esecuzione, i due sottosistemi vengono simulati in parallelo (per la parte deterministica viene utilizzato il metodo Runge-Kutta del 4° ordine; per la simulazione stocastica si utilizza il metodo di Gibson e Bruck). Le probabilità delle reazioni nel sottosistema stocastico durante uno step di simulazione vengono approssimate a delle costanti, anche se nella teoria potrebbero cambiare per effetto del sottosistema deterministico.

I parametri di tale metodo sono descritto qui di seguito:

*Max Internal Step*: parametro intero positivo che indica il numero massimo di step interni che l'integratore può effettuare prima dell'unità di tempo successiva. Di default vale 10000000.

*Lower Limit*: valore numerico double che specifica il limite inferiore del numero di particelle: le specie aventi un numero di particelle inferiore a tale valore sono considerate con un 'basso' numero di particelle. Di default vale 800. (Ovviamente questo parametro dovrà essere inferiore al seguente).

*Upper Limit*: valore numerico double che specifica il limite superiore del numero di particelle: le specie aventi un numero di particelle maggiore a tale valore sono considerate con un 'alto' numero di particelle. Di default vale 1000. Tutte le specie aventi un numero di particelle compreso tra il Lower e l'Upper Limit non cambiano il loro stato.

*Runge Kutta Stepsize*: parametro numerico positivo double che indica la dimensione dello step utilizzato dal metodo Runge-Kutta. Di default vale 0.001.

*Partitioning Interval*: valore intero positivo che specifica dopo quanti step interni deve essere ricalcolata la partizione del sistema.

*Use Random Seed*: di default tale parametro booleano vale 0 e significa che il 'random seed' è settata ad un valore random prima di ogni esecuzione: di conseguenza esecuzioni diverse possono restituire traiettorie differenti. Se impostato ad 1, il parametro indica che il 'random seed' viene settato al valore impostato dall'utente.

*Random Seed*: valore intero privo di segno utilizzato come 'random seed' nell'esecuzione (viene utilizzato solo se il parametro precedente è stato settato ad 1). Di default vale 1.

## A.4 Lyapunov Exponents

COPASI permette di calcolare gli esponenti di Lyapunov di una traiettoria e la divergenza media del sistema. Gli esponenti vengono calcolati per il sistema ridotto, in modo tale che il massimo numero di esponenti che possono essere calcolati è pari al numero di variabili indipendenti. Se viene richiesto di calcolare una quantità minore di esponenti, verranno calcolati soltanto i maggiori.

A tal fine, il programma utilizza l'algoritmo proposto da Wolf et al.<sup>4</sup>, secondo il quale vengono integrate contemporaneamente una traiettoria di riferimento e altre  $N$  traiettorie 'differenza' (dove  $N$  è il numero di esponenti richiesti) in un sistema linearizzando attorno alla traiettoria di riferimento.

- L'integrazione viene effettuata per un piccolo intervallo di tempo ('intervallo di ortonormalizzazione') e successivamente i vettori differenza vengono ri-ortonormalizzati. Gli esponenti per questo intervallo di tempo vengono calcolati in base a quante traiettorie differenza convergono o divergono da quella

---

<sup>4</sup>A. Wolf, J. B. Swift, H. Swinney, and J. A. Vastano. "Determining Lyapunov exponents from a time series". *Physica*. 16D. 285 - 317. 1985

di riferimento. Questo calcolo viene ripetuto e gli esponenti 'locali' vengono poi mediati su tutta la traiettoria.

- Se richiesta, la divergenza viene calcolata come la media della traccia dello Jacobiano. Nonostante non sia numericamente necessario, anche la divergenza viene calcolata negli stessi brevi intervalli di tempo utilizzati per il calcolo degli esponenti di Lyapunov. Ciò permette di comparare i valori locali della divergenza con gli esponenti locali.

Nel caso si sia interessati soltanto al risultato finale del calcolo degli esponenti di Lyapunov e della divergenza media, si può utilizzare visualizzare semplicemente il report che COPASI offre di default in seguito all'esecuzione di questa analisi. Se invece si desidera conoscere i risultati 'locali' per i singoli intervalli di ortonormalizzazione, è necessario definire manualmente dei report più completi.

La matrice Jacobiana utilizzata sia per il calcolo degli esponenti sia per il calcolo della divergenza è calcolata usando differenze finite: l'integrazione viene effettuata utilizzando l'algoritmo LSODA<sup>5</sup>.

Quando si esegue tale funzionalità di COPASI, è possibile settare i parametri:

*Orthonormalization interval:* intervallo temporale dopo il quale ha luogo l'ortonormalizzazione delle traiettorie differenza. Un valore piccolo di questo parametro porta ad avere un'accuratezza maggiore ma implica un maggiore tempo computazionale. Di default vale 1.0.

*Overall time:* specifica il tempo totale della computazione. L'integrazione verrà ripetuta negli intervalli di ortonormalizzazione fino a che non si raggiunge l'Overall time. Di default vale 1000.

I parametri rimanenti sono gli stessi definiti per le simulazioni deterministiche.

## A.5 Time Scale Separation Analysis

Questa funzione di COPASI utilizza due metodi di 'Time Scale Separation' per ridurre i sistemi analizzati. Entrambi questi metodi dipendono dalla presenza di una vasta gamma di caratteristiche dipendenti dal tempo nei sistemi biologici e si basano su un'analisi locale della matrice Jacobiana, che viene partizionata in componenti veloci e lenti nel punto iniziale dell'intervallo scelto dall'utente.

I parametri comuni ad entrambi i metodi sono i seguenti:

*Intervals:* numero di intervalli nei quali applicare il metodo. COPASI implementa l'analisi nel punto iniziale di ciascun intervallo.

*Deuffhard Tolerance:* valore numerico positivo che specifica il massimo errore tollerato per i modi lenti (per la definizione di 'modi lenti' si veda nel seguito). Di default vale  $1 \cdot 10^{-6}$ .

### A.5.1 Metodo ILDM (Deuffhard)

Il metodo ILDM utilizza l'algoritmo sviluppato da Deuffhard e Heroth, che si basa sulla decomposizione in modi 'lenti' e modi 'veloci'.

Tale decomposizione a blocchi della matrice Jacobiana viene effettuata in due step:

1. In primo luogo una decomposizione di Schur produce un blocco di matrice triangolare superiore.
2. In secondo luogo, si ottiene una struttura disaccoppiata della matrice Jacobiana così trasformata risolvendo un'equazione di Sylvester.

Il risultato di tale procedura è una trasformazione dei vettori di stato in nuovi modi, che poi vengono separati in 'lenti' e 'veloci'. Di conseguenza, la dinamica completa di un sistema di reazioni di  $n$  equazioni differenziali (ODE) viene ridotto in un sistema DAE contenente  $n_{slow}$  equazioni differenziali ordinarie ed  $n - n_{slow}$  equazioni algebriche. Il numero  $n_{slow}$  delle variabili lente è calcolato iterativamente utilizzando il criterio di tolleranza di Deuffhard e Heroth.

Poiché il metodo implementato da COPASI si focalizza sulla riduzione di reti biochimiche e non di semplici equazioni matematiche, COPASI esegue un'analisi delle matrici derivanti dalla risoluzione dell'equazione di Sylvester.

---

<sup>5</sup>A.C. Hindmarsh. ODEPACK, "A Systematized Collection of ODE Solvers". Scientific Computing, R. S. Stepleman et al. (eds.), North-Holland, Amsterdam, IMACS Transactions on Scientific Computation, 1. 55 - 64, 1983

## A.5.2 Metodo ILDM Modificato

Il secondo metodo utilizzabile, che consiste in una versione modificata del precedente, è stato sviluppato dagli autori del programma COPASI. Lo scopo principale di tale metodo è quello di riconoscere i metaboliti 'veloci' anziché i modi veloci. Ciò comporta che l'analisi può essere formulata senza una trasformazione lineare del sistema.

## A.6 Cross Section

COPASI mette a disposizione tale funzione per analizzare le 'sezioni trasversali', ovvero le curve di livello del sistema relative ad un parametro selezionato dall'utente.

## A.7 Parameter Scan

Questa funzione di COPASI permette di osservare i valori che possono essere assunti da uno o più parametri selezionati e in che modo tali variazioni influenzino le variabili di interesse. COPASI offre la visualizzazione di diversi output, che possono essere selezionati dall'utente, in base ai suoi interessi specifici.

## A.8 Parameter Estimation

COPASI offre una funzione che permette di stimare il valore dei parametri del modello basandosi su un dataset, che può essere il risultato di esperimenti eseguiti allo scorrere del tempo, in stato stazionario o in entrambe le situazioni. Il software legge un dataset, che può essere composto da diversi file, ciascuno contenente esperimenti multipli, e lo carica. Dopodiché cerca di fittare uno o più parametri, stabiliti dall'utente, utilizzando tali risultati sperimentali. I metodi utilizzati da questa funzione, sono gli stessi utilizzati per l'Ottimizzazione.

### A.8.1 Dati sperimentali

Per poter eseguire la stima dei parametri, è necessario caricare il dataset dei risultati sperimentali. Ciascun esperimento del dataset contribuisce alla funzione obiettivo attraverso la seguente somma pesata di quadrati:

$$E(P) = \sum_{i,j} \omega_j \cdot (x_{i,j} - y_{i,j}(P))^2,$$

dove  $P$  è l'insieme corrente dei parametri da testare,  $x_{i,j}$  è un punto nel dataset e  $y_{i,j}(P)$  è il corrispondente valore simulato. Gli indici  $i$  e  $j$  identificano la riga e la colonna del dataset.  $\omega_j$  è il peso assegnato a ciascuna colonna. Tale peso può essere calcolato in tre metodi differenti, che l'utente può scegliere:

1. media:  $\omega_j = \frac{1}{|\langle x_j \rangle|}$ ;
2. media quadratica:  $\omega_j = \frac{1}{\sqrt{\langle x_j^2 \rangle}}$ ;
3. deviazione standard:  $\omega_j = \frac{1}{\langle x_j^2 \rangle - \langle x_j \rangle^2}$ .

In alternativa, l'utente può modificare manualmente i pesi.

## A.9 Ottimizzazione

Oltre agli strumenti appena descritti, che permettono di eseguire molto dettagliatamente un'analisi del modello, COPASI mette a disposizione anche dei tasks di ottimizzazione, di Metabolic Control Analysis (MCA) e di sensitività.

Una delle funzioni più interessanti messe a disposizione del software è quindi quella di ottimizzazione, che permette di minimizzare (o massimizzare) una funzione obiettivo definita dall'utente al variare di uno o più parametri all'interno di range definiti.

Per minimizzare una data funzione, COPASI mette a disposizione diversi metodi tra cui scegliere, che descriveremo brevemente qui di seguito.

### A.9.1 Evolutionary Programming (EP)

Il metodo EP è una tecnica che vuole mimare l'evoluzione e si basa sulla riproduzione e sulla selezione. Un algoritmo EP è composto da individui che si riproducono e competono tra di loro: ciascuno è una potenziale soluzione al problema di ottimizzazione ed è rappresentato da un cosiddetto "genoma", all'interno del quale ogni gene corrisponde ad un parametro tarabile. Ad ogni generazione dell'algoritmo, ciascun individuo si riproduce, dividendosi in due individui: uno dei due contiene esattamente lo stesso "genoma" del genitore, mentre l'altro contiene alcune mutazioni (i valori dei parametri di ciascun gene vengono leggermente modificati). Alla fine di ciascuna generazione perciò, sono presenti il doppio degli algoritmi della generazione precedente. Successivamente ogni individuo viene confrontato con un certo numero di altri individui, per valutare il numero di 'vittorie', ossia il numero di altri individui rispetto ai quali quello in esame permette di ottenere delle performance migliori.

A tutti gli individui viene quindi assegnato un punteggio in base al numero di vittorie e la popolazione viene quindi ridotta al numero originale di individui eliminando quelli con un punteggio inferiore.

I parametri da settare per implementare tale metodo sono i seguenti:

*Number of Generations*: numero di generazioni che devono essere implementate dall'algoritmo. Di default vale 200.

*Population Size*: dimensione della popolazione, cioè il numero di individui che sopravvivono al termine di ciascuna generazione. Di default vale 20.

*Random Number Generator*: valore booleano che indica quale generatore di numeri casuali utilizzare (se settato a 0, viene utilizzato R250, se settato ad 1, di default, viene utilizzato il generatore Mersenne Twister).

*Seed*: indica il 'seed' della generazione dei numeri casuali. Se settato a 0 indica a COPASI di utilizzare un valore random.

### A.9.2 Evolutionary Strategy con Ranking Stocastico (SRES)

Il metodo SRES è simile al precedente, ma ciascun genitore ha molteplici figli durante ciascuna generazione. Ciascun figlio contiene una ricombinazione dei geni con un altro genitore, con l'aggiunta di alcune mutazioni. Il metodo si assicura che ciascun parametro stia all'interno dei suoi limiti, ma i vincoli delle soluzioni possono essere violati: se ciò accade, viene calcolato il valore

$$\varphi = \sum_{c_j < l_{c_j}} (l_{c_j} - c_j)^2 + \sum_{c_j > u_{c_j}} (c_j - u_{c_j})^2,$$

dove i vincoli sono dati da  $c_j \in (l_{c_j}, u_{c_j})$ . Il valore  $\varphi$  così calcolato verrà utilizzato all'interno della selezione, che viene effettuata seguendo una procedura simile a quella che verrà descritta in seguito per il Genetic Algorithm SR.

I parametri relativi a questo metodo sono gli stessi del metodo precedente, con l'aggiunta di:

*Pf*: valore numerico compreso tra (0, 1) che indica la probabilità che gli individui escano dai limiti del parametro o violino i vincoli durante la selezione. Di default vale 0.475.

### A.9.3 Genetic Algorithm (GA)

Analogamente all'algoritmo EP, anche l'algoritmo GA mima l'evoluzione e si basa sulla riproduzione e sulla selezione. È composto anch'esso da individui che si riproducono e competono tra di loro: ciascuno è una potenziale soluzione al problema di ottimizzazione ed è rappresentato da un "genoma", all'interno del quale ogni gene corrisponde ad un parametro tarabile. Ad ogni generazione ciascun individuo viene accoppiato con un altro per produrre due figli, tramite la combinazione dei geni di entrambi i genitori (ciascun figlio conterrà geni provenienti da entrambi i genitori). Ogni nuovo gene però deve essere leggermente modificato rispetto ai geni dei genitori. Alla fine di ciascuna generazione viene attribuito un punteggio in base alle proprie performance e vengono eliminati gli individui con i punteggi più bassi, così da riportare la popolazione alla sua dimensione iniziale.

Poiché COPASI implementa l'ottimizzazione di una rete biochimica, sono state effettuate dagli autori alcune variazioni all'interno dell'algoritmo, le cui caratteristiche principali si possono così riassumere:

- i parametri (geni) sono rappresentati floating-point, anziché nella rappresentazione binaria (più usuale);
- la mutazione consiste nell'aggiunta al valore di ciascun gene di un numero random preso da una distribuzione normale con media nulla e deviazione standard pari al 10% del valore del parametro. Se in questo modo il gene eccede i suoi limiti, viene posto uguale al suo valore limite.
- l'incrocio tra i geni nel momento della riproduzione è implementata in modo tale che nessun gene venga eliminato (i geni vengono mantenuti all'interno dei loro limiti). Il numero dei punti di cross-over è un numero casuale estratto da una distribuzione uniforme nell'intervallo che va da zero alla metà del numero di parametri modificabili.
- Durante la selezione, ciascun individuo compete con un numero di individui pari al 20% della dimensione della popolazione. I competitori vengono scelti casualmente.
- La popolazione iniziale contiene un individuo i cui geni hanno i valori iniziali dei parametri, mentre i geni degli altri individui sono inizializzati ad un valore random compreso all'interno dei loro limiti.
- Nel caso in cui l'individuo con un punteggio più alto non cambi per 10 generazioni, il 10% degli individui con punteggio inferiore vengono rimpiazzati da individui con geni randomici. Se l'individuo 'migliore' non cambia per 30 (50) generazioni, il 30% (50%) degli individui (peggiori) viene sostituito con individui con parametri casuali. Tale procedura permette all'algoritmo di non fermarsi ad un minimo locale e, in un certo senso, realizza un aumento in termini di mutazioni quando la popolazione diventa uniforme.

I parametri di tale metodo sono analoghi a quelli del metodo precedente.

#### A.9.4 Genetic Algorithm SR

Questo algoritmo è molto simile al precedente, con due eccezioni secondo le quali le mutazioni non sono forzate a rimanere entro ai limiti fissati e la selezione viene effettuata tramite un metodo di Bubble-Sort con un fattore random, come viene descritto da Runarsson e Yao<sup>6</sup>.

Se, durante la mutazione, i parametri eccedono i limiti, viene calcolato:

$$\varphi = \sum_{p_i < l_{p_i}} (l_{p_i} - p_i)^2 + \sum_{p_i > u_{p_i}} (p_i - u_{p_i})^2 + \sum_{c_j < l_{c_j}} (l_{c_j} - c_j)^2 + \sum_{c_j > u_{c_j}} (c_j - u_{c_j})^2,$$

dove i parametri sono soggetti a  $p_i \in (l_{p_i}, u_{p_i})$  e i vincoli sono descritti da  $c_j \in (l_{c_j}, u_{c_j})$ . Tale valore di  $\varphi$  viene utilizzato all'interno della selezione, che implementa un Bubble-Sort, il quale incorpora una probabilità per confrontare i valori dell'obiettivo per gli individui con  $\varphi = 0$ .

#### A.9.5 Hooke & Jeeves

Il metodo Hooke & Jeeves è un metodo diretto di ricerca che cerca il minimo di una funzione lineare senza richiedere né calcolare le derivate della funzione. Esso è basato su un'euristica che suggerisce la direzione decrescente usando i valori che la funzione assume in un numero di iterazioni calcolate precedentemente.

I parametri da settare per tale metodo sono:

*Iteration Limit:* massimo numero di iterazioni da effettuare (default: 50).

*Tolerance:* tolleranza con cui determinare la soluzione. Se la diminuzione tra due passi successivi è minore della tolleranza, l'algoritmo si ferma. Di default vale  $10^{-5}$ .

*Rho:* valore compreso tra (0, 1), che determina il fattore di cui ridurre la dimensione dello step ad ogni iterazione. Di default vale 0.2.

---

<sup>6</sup>T. Runarsson and X. Yao. "Stochastic ranking for constrained evolutionary optimization". *EEE Transactions on Evolutionary Computation*. 4. 284 - 294. 2000

### A.9.6 Levenberg - Marquardt

Tale metodo è un metodo di Gradient descent, che implementa un ibrido tra il metodo dello Steepest descent e il metodo di Newton.

- Il metodo di Newton ricerca il minimo di una funzione non lineare seguendo la direzione di discesa del gradiente determinata dalle derivate prime e seconde della funzione. Tale metodo può convergere quadraticamente se la condizione iniziale è vicina al punto di minimo, ma può anche non convergere affatto.
- Il metodo dello Steepest descent esegue una ricerca del minimo basandosi soltanto sulle derivate prime. Anche se la convergenza in questo caso è solamente lineare, essa è sempre garantita.

Le novità riportate da Levenberg e Marquardt consiste nell'aggiunta di un fattore moltiplicativo agli elementi diagonali della matrice Hessiana utilizzata nel metodo di Newton, per evitare la possibilità di mancata convergenza: se tale valore diventa molto elevato il metodo diventa equivalente allo Steepest descent: per questo motivo tale metodo si può considerare a cavallo tra i due.

I parametri relativi a tale metodo sono:

*Iteration Limit:* massimo numero di iterazioni da effettuare (default: 200).

*Tolerance:* tolleranza con cui determinare la soluzione. Se la diminuzione tra due passi successivi è minore della tolleranza, l'algoritmo si ferma. Di default vale  $10^{-5}$ .

### A.9.7 Nelder - Mead

Il metodo Nelder - Mead è conosciuto anche come metodo del Simplex. Il Simplex è un politopo di  $N + 1$  vertici in  $N$  dimensioni; la funzione obiettivo viene valutata ad ogni vertice. In seguito a questi valori calcolati, viene costruito un nuovo politopo. Il passo più semplice consiste nel rimpiazzare il punto peggiore con un punto riflesso attraverso il baricentro dei rimanenti  $N$  punti. Se questo punto è migliore di quello corrente, si può provare a muoversi esponenzialmente lungo questa linea; se invece questo punto non risulta essere molto migliore del precedente, significa che ci si sta muovendo lungo una valle, in modo da ridurre il simplex verso il punto migliore.

I parametri di questo metodo sono gli stessi del metodo precedente, con l'aggiunta di:

*Scale:* numero positivo che determina la dimensione del simplex iniziale. Gli archi del politopo sono inversamente proporzionali a tale parametro, cioè un valore elevato realizza un simplex iniziale piccolo. Di default vale 10.

### A.9.8 Particle Swarm

Il metodo di ottimizzazione Particle Swarm, sviluppato da Kennedy ed Eberhart<sup>7</sup>, prende ispirazione da uno stormo di uccelli o di un gruppo di pesci che sono alla ricerca di cibo. Ciascun elemento partecipante ha una posizione  $X_i$  e una velocità  $V_i$  nello spazio dei parametri. In aggiunta ciascuno tiene in memoria il suo miglior valore ottenuto della funzione obiettivo ( $O$ ) e la posizione  $M_i$ . Dalle proprie informazioni e da quelle del suo miglior vicino, calcola una nuova velocità: con questa informazione viene quindi aggiornata anche la posizione.

I parametri del metodo sono i seguenti:

*Iteration Limit:* massimo numero di iterazioni da effettuare (default: 200).

*Swarm Size:* numero di elementi nello sciame.

*Std. Deviation:* parametro che determina un metodo alternativo di terminazione. Se la deviazione standard del valore della funzione obiettivo di ciascun elemento e delle posizioni migliori è più piccola di tale valore settato, l'algoritmo si ferma. Di default vale  $10^{-6}$ .

*Random Number Generator:* valore booleano che indica quale generatore di numeri casuali utilizzare (se settato a 0, viene utilizzato R250, se settato ad 1, di default, viene utilizzato il generatore Mersenne Twister).

---

<sup>7</sup>J. Kennedy and R. Eberhart. "Particle Swarm Optimization". Proceedings of the Fourth IEEE International Conference on Neural Networks, Perth, Australia. 1942 - 1948. 1995

*Seed*: indica il 'seed' della generazione dei numeri casuali. Se settato a 0 indica a COPASI di utilizzare un valore random.

### A.9.9 Praxis

Tale metodo è un metodo di ricerca diretto, che ricerca il minimo di una funzione non lineare senza la necessità di conoscere o calcolare le derivate di tale funzione: il metodo prende spunto dal metodo degli assi principali, secondo il quale viene minimizzato ciascun parametro modificabile (direzione) uno per volta. In Praxis le direzioni vengono scelte in modo da non coincidere con gli assi principali.

Il parametro da fissare è:

*Tolerance*: se due stime consecutive si differenziano per un valore inferiore alla tolleranza fissata, il metodo termina. Di default vale  $10^{-5}$ .

### A.9.10 Random Search

Il metodo Random Search cerca di raggiungere l'ottimo testando il valore della funzione obiettivo su una serie di combinazioni di valori casuali dei parametri modificabili. Tali valori casuali sono generati compatibilmente ai limiti imposti dall'utente, inoltre tutte le combinazioni che non soddisfano pienamente i vincoli vengono eliminate.

Il metodo garantisce di raggiungere l'ottimo globale in un numero infinito di iterazioni, perciò solitamente viene eseguito impostando un alto numero di iterazioni.

I parametri da impostare per l'esecuzione del Random Search sono i seguenti:

*Number of Iterations*: numero di set di parametri da calcolare prima che l'algoritmo termini. Di default vale 100000.

*Random Number Generator*: valore booleano che indica quale generatore di numeri casuali utilizzare (se settato a 0, viene utilizzato R250, se settato ad 1, di default, viene utilizzato il generatore Mersenne Twister).

*Seed*: indica il 'seed' della generazione dei numeri casuali. Se settato a 0 indica a COPASI di utilizzare un valore random.

### A.9.11 Simulated Annealing

Il metodo Simulated Annealing, sviluppato da Kirkpatrick et al.<sup>8</sup>, prende ispirazione da meccanismi statistici e dalla metodologia secondo cui si formano i cristalli perfetti. I cristalli perfetti si formano attraverso la fusione delle sostanze di interesse e ad un raffreddamento molto lento. Alle alte temperature le particelle vibrano con ampia ampiezza e ciò permette la ricerca di ottimo globale. Quando la temperatura diminuisce, diminuiscono anche le vibrazioni fino a che il sistema si assesta ad un ottimo globale (cristallo perfetto).

Analogamente, nell'implementazione di tale metodo, la funzione obiettivo viene considerata come una misura dell'energia del sistema e viene mantenuta costante per un certo numero di iterazioni (un ciclo di temperatura). Ad ogni iterazione i parametri vengono modificati ad un valore vicino nello spazio parametrico e viene ricalcolato il valore della funzione obiettivo: se tale valore decresce, il nuovo stato viene accettato, altrimenti viene accettato con una probabilità che segue la distribuzione di Boltzmann (una temperatura più alta corrisponde ad una maggiore probabilità di accettare il nuovo stato). Dopo un numero fissato di temperature, se non è ancora stato raggiunto il tempo massimo fissato, la temperatura del sistema viene ridotta e l'algoritmo prosegue.

Tale algoritmo è un algoritmo stocastico che garantisce la convergenza in un numero infinito di iterazioni: è uno dei metodi di ottimizzazione più robusti, ma, allo stesso tempo è anche uno dei più lenti.

COPASI implementa questo algoritmo basandosi sul codice di Corana et al.<sup>9</sup>, che setta i parametri seguendo alcune regole:

- Oggi ciclo di temperatura comprende  $10 \cdot p \cdot \max(5 \cdot p, 100)$  step randomici, dove  $p$  è il numero di parametri da ottimizzare;

<sup>8</sup>S. Kirkpatrick, J., C.D. Gelatt, and M. P. Vecchi. "Optimization by simulated annealing". Science. 220. 671 - 680. 1983

<sup>9</sup>A. Corana, M. Marchesi, C. Martini, and S. Ridella. "Minimizing multimodal functions of continuous variables with the "simulated annealing" algorithm". ACM Trans. Math. Softw.. 13. 262 - 280. 1987

- ad ogni step, una nuova soluzione candidata è accettata :
- se riduce il valore della funzione obiettivo, oppure
- con una probabilità pari a  $e^{-\frac{\Delta f}{T}}$ , dove  $\Delta f$  è l'incremento del valore della funzione obiettivo e  $T$  la temperatura corrente.
- Il criterio di terminazione viene applicato agli ultimi due cicli di temperatura (l'algoritmo si ferma se il cambiamento del valore della funzione obiettivo alla temperatura corrente e a quella precedente è minore della tolleranza fissata).

In accordo con la procedura seguita dall'algoritmo, i parametri da fissare sono i seguenti:

*Start Temperature:* temperatura iniziale del sistema (più alto è tale valore, più è alta la probabilità di raggiungere un ottimo globale). Di default vale 1.

*Cooling Factor:* rate di riduzione della temperatura tra un ciclo e il successivo. L'aggiornamento della temperatura segue la seguente formula:

$$T_{new} = T_{old} \cdot Cooling\ Factor.$$

Affinchè l'algoritmo funzioni bene, tale fattore deve mantenersi abbastanza vicino all'unità (di default vale 0.85).

*Tolerance:* se due stime consecutive si differenziano per un valore inferiore alla tolleranza fissata, il metodo termina. Di default vale  $10^{-6}$ .

*Random Number Generator:* valore booleano che indica quale generatore di numeri casuali utilizzare (se settato a 0, viene utilizzato R250, se settato ad 1, di default, viene utilizzato il generatore Mersenne Twister).

*Seed:* indica il 'seed' della generazione dei numeri casuali. Se settato a 0 indica a COPASI di utilizzare un valore random.

### A.9.12 Steepest Descent

Come già precedentemente annunciato, tale metodo, suggerito da D. B. Fogel, L. J. Fogel ed Atmar<sup>10</sup>, segue la direzione di discesa (definita dal gradiente negativo della funzione obiettivo) su un'iper-superficie della funzione obiettivo, al fine di raggiungere un minimo locale.

I due parametri da fissare sono:

*Iteration Limit:* massimo numero di iterazioni da effettuare (default: 100).

*Tolerance:* se due stime consecutive si differenziano per un valore inferiore alla tolleranza fissata, il metodo termina. Di default vale  $10^{-6}$ .

### A.9.13 Truncated Newton

Il metodo Truncated Newton è una variante più sofisticata del metodo di Newton, che, per calcolare la direzione di Newton, computa una soluzione incompleta (troncata) del sistema di equazioni. Analogamente al metodo Levenberg - Marquardt, tale algoritmo si pone tra lo Steepest descent e l'algoritmo originale di Newton.

In questo caso non c'è alcun parametro da fissare.

---

<sup>10</sup>D.B. Fogel, L.J. Fogel, and J.W. Atmar. "Meta-evolutionary programming". 25th Asiloma Conference on Signals, Systems and Computers. IEEE Computer Society, Asilomar . 540 - 545. 1992

## A.10 Metabolic Control Analysis (MCA)

La Metabolic Control Analysis è un'analisi di sensitività, che permette di calcolare l'effetto relativo di un determinato controllo sulle variabili del sistema (flussi e concentrazioni), ad ogni passo. Questo controllo è misurato applicando una perturbazione nello step analizzato e calcolando il valore della variabile di interesse, dopo che il sistema si è stabilizzato in un nuovo stato stazionario.

Le opzioni di tale metodo sono le stesse coinvolte nell'analisi di Stato Stazionario e di simulazione deterministica.

### A.10.1 Control Coefficients

L'analisi inerente al controllo dei coefficienti consiste in una misura relativa di come le variabili del sistema vengano variate in seguito alla perturbazione di un determinato parametro. Viene definito nel modo seguente:

$$C_{v_i}^A = \frac{\partial A}{\partial v_i} \frac{v_i}{A};$$

dove  $A$  è la variabile,  $i$  è lo step (enzima) e  $v$  è il rate in stato stazionario dello step perturbato. Anche se il controllo dei coefficienti più comunemente analizzato è quello inerente ai flussi e alle concentrazioni delle specie, ciascuna variabile del sistema può essere analizzata tramite la MCA.

### A.10.2 Summation Theorem

Una delle più importanti proprietà dello stato stazionario è stata scoperta tramite la MCA: essa riguarda la somma di tutti i coefficienti del flusso di controllo del pathway. Si può dimostrare infatti che per un dato flusso di riferimento, la somma di tutti i coefficienti del flusso di controllo è pari all'unità:

$$\sum_i C_{v_i}^J = 1.$$

Inoltre, per una concentrazione di riferimento, la somma di tutti i coefficienti della concentrazione di controllo è pari a zero:

$$\sum_i C_{v_i}^{[M]} = 0,$$

dove la somma è effettuata su tutti gli step del sistema.

Una conseguenza della prima uguaglianza consiste nel fatto che, aumentando alcuni coefficienti del controllo di flusso, gli altri debbano diminuire (per mantenere la somma unitaria). Ovviamente, più il valore relativo ad un enzima o ad un gruppo di enzimi si avvicina a 1, maggiore sarà l'influenza di quell'enzima o di quel gruppo di enzimi sul controllo di un determinato pathway metabolico.

Da entrambe le uguaglianze si deduce che i coefficienti di controllo sono una proprietà globale e che nei sistemi meyalobici, il controllo è una proprietà del sistema che dipende da tutti gli elementi (step) della rete.

### A.10.3 Enzyme Kinetics ed Elasticity Coefficients

Nell'analisi relativa alle cinetiche degli enzimi viene studiato il comportamento di enzimi isolati attraverso la dipendenza dei rate iniziali delle reazioni dalle concentrazioni dei substrati.

Nella MCA, le proprietà di ciascun enzima (isolato) vengono misurate in un modo molto simile a quello inerente alle proprietà del flusso di controllo, utilizzando un coefficiente di elasticità. In questo caso si considera l'effetto delle perturbazioni di un parametro della reazione sul rate della reazione 'locale'. Con il termine 'locale' si intende che tale misura di sensitività si riferisce alla reazione isolata che ha le stesse caratteristiche (concentrazione enzimatica, temperatura, ecc.) del sistema completo nello stato stazionario di interesse.

I coefficienti di elasticità sono definiti come il rapporto tra la variazione relativa del rate locale e la variazione relativa di un parametro, che si può descrivere in termini infinitesimi nel modo seguente:

$$\varepsilon_p^{v_i} = \frac{\partial v_i}{\partial p} \frac{p}{v_i};$$

dove  $v$  è il rate dell'enzima in questione e  $p$  è il parametro perturbato. Normalmente il parametro  $p$  rappresenta la concentrazione di un metabolita (ad esempio il substrato o il prodotto). Le elasticità possono essere positive nel caso del substrato e degli attivatori, negative nel caso dei prodotti e degli inibitori. Ciascun enzima ha quindi un numero di coefficienti di elasticità pari al numero di parametri che lo influenzano.

A differenza dei coefficienti di controllo, che sono una proprietà del sistema, i coefficienti di elasticità misurano quanto un enzima isolato è sensibile alle variazioni dei parametri che lo interessano. Tali coefficienti possono essere ottenuti attraverso la derivazione parziale delle funzioni cinetiche. Similmente ai coefficienti di controllo, anche i coefficienti di elasticità non sono costanti: dipendono dal valore dei parametri rilevanti e differiscono per ciascuno stato stazionario.

#### A.10.4 Relazioni di connettività

La MCA può essere utilizzata anche per collegare tra loro le proprietà delle reazioni individuali (proprietà locali) e le proprietà globali dell'intero pathway. Ciò è possibile utilizzando il teorema della connettività<sup>11</sup>, che lega i coefficienti di controllo con i coefficienti di elasticità delle reazioni con specie intermedie in comune.

Tale teorema afferma che, per quanto riguarda una specie comune, la somma dei prodotti dei coefficienti del flusso di controllo di tutti gli step ( $i$ ) affetti da  $S$  e i coefficienti di elasticità riguardanti  $S$  è zero:

$$\sum_i C_{v_i}^J \varepsilon_{[S]}^{v_i} = 0.$$

Per quanto riguarda i coefficienti di controllo delle concentrazioni si applicano le due equazioni riportate da Westerhoff e Chen<sup>12</sup>:

$$\sum_i C_{v_i}^{[A]} \varepsilon_{[S]}^{v_i} = 0, \quad \text{se } A \neq S$$

$$\sum_i C_{v_i}^{[A]} \varepsilon_{[S]}^{v_i} = -1.$$

La prima equazione si applica se le specie di riferimento  $A$  sono diverse dalle specie perturbate  $S$ ; la seconda invece si applica se le specie di riferimento sono le stesse che sono state perturbate.

I teoremi della connettività permettono alla MCA di descrivere come la perturbazione di alcune specie in un pathway si propaghi attraverso la catena enzimatica. Le proprietà cinetiche locali di ciascun enzima fanno sì che le perturbazioni si propaghino ai vicini più prossimi.

#### A.10.5 Scaling

Il metodo restituisce i valori di sensitività calcolati sia scalati sia non scalati. Questo deriva dal fatto che il metodo implementato calcola i valori indipendentemente dallo stato stazionario adeguato. Tali valori vengono poi scalati in base alle concentrazioni e ai flussi dello stato stazionario appropriato.

#### A.10.6 Sensitivity Analysis

COPASI mette a disposizione un'analisi di sensitività rispetto a diversi parametri. Generalmente la sensitività è una misura di quanto viene modificato il valore di una grandezza 'osservabile' in relazione alla variazione di un determinato parametro.

Il programma permette di calcolare un array completo di sensitività per tutte le variabili 'osservabili' del modello in relazione alla lista dei parametri.

L'implementazione di tale analisi è effettuata in COPASI mediante la differenziazioni numerica utilizzando differenze finite.

I parametri che si possono settare nel calcolo della sensitività sono i seguenti:

**Delta factor:** valore utilizzato per il calcolo del delta per la differenziazione numerica. Il valore del delta è dato dal prodotto di tale fattore e del valore assoluto corrente della variabile. Di default vale  $10^{-6}$ .

Se il delta così calcolato è più piccolo del valore minimo accettabile, viene utilizzato il valore 'Delta minimum' settato dall'utente.

<sup>11</sup>H. Kacser and J.A. Burns. "The control of flux". Symp. Soc. Exp. Biol., 27, 65 - 104, 1973

<sup>12</sup>H.V. Westerhoff and Y.-D. Chen. "How do enzyme activities control species concentrations? An additional theorem in the theory of metabolic control." Eur. J. Biochem. 142, 425 - 430, 1984

*Delta minimum:* valore minimo che può essere assunto dal delta nella differenziazione numerica. Di default vale  $10^{-12}$ .

# Appendix B

## Utilizzo e funzionalità di COBRA Toolbox

### B.1 Materiali

#### B.1.1 Equipment

- La versione COBRA Toolbox 2.0 o superiore (<http://www.COBRAToolbox.org>);
- Un computer in grado di eseguire MATLAB;
- La versione 7.0 o superiore di MATLAB (MathWorks): un software per il calcolo numerico e la visualizzazione (<http://www.mathworks.com>);
- libSBML programming library 4.0.1 o superiore (<http://www.sbml.org>)
- SBMLToolbox version 3.1.1 o superiore per MATLAB per poter caricare e scrivere modelli in formato SBML (<http://www.sbml.org>) ;
- Un risolutore per problemi LP (linear programming). Al momento, COBRA Toolbox supporta i seguenti risolutori:
  1. Gurobi (Gurobi Optimization, <http://www.gurobi.com>) mediante Gurobi Mex ([http://www.convexoptimization.com/wikimization/index.php/Gurobi\\_mex](http://www.convexoptimization.com/wikimization/index.php/Gurobi_mex))
  2. CPLEX (ILOG) mediante Tomlab (Tomlab Optimization, <http://tomopt.com>)
  3. GLPK (<http://www.gnu.org/software/glpk>) mediante glpk mex (<http://glpkmex.sourceforge.net>). Occorre però fare attenzione poichè GLPK non fornisce delle soluzioni accurate per i metodi OptKnock e Genetic Design Local Search (GDLS), che verranno descritti nel seguito.
- Un risolutore per problemi QP (quadratic programming) (opzionale). Al momento COBRA Toolbox supporta i seguenti:
  1. CPLEX (ILOG) mediante Tomlab;
  2. QPNG (parte di GLPK). Questo risolutore però non fornisce soluzioni accurate per il metodo MOMA, implementato nel Toolbox.

NOTA: È possibile che altri risolutori possano fornire delle soluzioni corrette (come, ad esempio Mosek, <http://www.mosek.com>; LINDO, <http://www.lindo.com>; and PDCO, <http://www.stanford.edu/group/SOL/software/pdco.html>), ma non sono stati validati per i metodi implementati in COBRA Toolbox.

Ad esempio, per problemi LP esistono altri solver, quali Mosek (<http://www.mosek.com>), LINDO (<http://www.lindo.com>) e PDCO (<http://www.stanford.edu/group/SOL/software/pdco.html>); Mosek e PDCO possono essere utilizzati anche per problemi QP.

- Un risolutore per problemi NLP (nonlinear programming) (opzionale). Al momento COBRA Toolbox supporta SNOPT mediante Tomlab.

Per ottenere delle performance migliori, può rendersi necessario aggiustare i parametri dei risolutori utilizzati.

## B.1.2 Equipment setup

Per installare COBRA Toolbox ci sono due possibilità:

1. 'à la carte', che contiene solamente COBRA Toolbox;
2. pacchetto completo, che include COBRA Toolbox, libSBML, SBMLToolbox, il kit GNU per problemi LP (GLPK) e gplkmex.

A seconda della scelta, l'installazione deve essere effettuata in modo differente:

- Installazione della versione 'à la carte':

1. Installare MATLAB;
2. installare libSBML, SBML Toolbox e i risolutori adeguati;
3. estrarre l'archivio COBRA 2.0.

- Installazione della versione in pacchetto completo:

1. Installare MATLAB;
2. estrarre l'archivio COBRA 2.0. `Cobra_Install_Path` è il percorso per la cartella di livello superiore del Toolbox;
3. aggiornare il path delle librerie, se necessario.

Per le analisi effettuate in questa tesi sono stati utilizzati i seguenti strumenti:

- versione Matlab R2009;
- COBRA Toolbox 2.0.5;
- libSBML-5.8.0-libxml2-x86;
- SBMLToolbox-4.1.0;
- solver LP: glpk;
- solver MILP: gurobi560;
- Microsoft Excel 2010.

## B.2 Descrizione delle procedure e dei metodi utilizzati nell'analisi

### B.2.1 Procedure preliminari

#### B.2.1.1 Inizializzazione del Toolbox

Per poter utilizzare i metodi contenuti nella cartella del Toolbox, è necessario inizializzare lo stesso, eseguendo il comando matlab:

*initCobraToolbox ()*

In questo modo tutti i metodi implementati possono essere invocati senza dover necessariamente posizionarsi nella cartella in cui sono stati salvati i file.

### B.2.1.2 Cambiare il risolutore in uso

È possibile settare i risolutori desiderati che vengono utilizzati da COBRA. Il comando da utilizzare a tale scopo è:

```
changeCobraSolver (solverName, [solverType]);
```

dove:

- *solverName* specifica il pacchetto del risolutore da utilizzare ('gurobi', 'tomlab\_cplex', 'glpk', o 'qpng');
- *solverType* indica il tipo di problema ('lp', 'milp', 'qp', 'miqp', o 'nlp'). Se non viene inserito alcun secondo parametro, di default viene assunto 'lp'.

Per verificare i risolutori settati, è sufficiente eseguire il comando privandolo degli argomenti.

### B.2.1.3 Test delle funzionalità del Toolbox

La versione 2.0 di COBRA permette di verificare se l'installazione è stata eseguita correttamente, testando tutte le funzionalità del Toolbox. A tal proposito sono stati implementati diversi esempi, necessari per eseguire delle analisi e verificare la correttezza del risultato.

Per eseguire il test su tutte le funzionalità, è sufficiente invocare il metodo:

```
testAll ();
```

se invece si è interessati a testare solo alcune specifiche funzioni, si può seguire il test corrispondente, scegliendo il metodo appropriato dalla cartella 'testing'. Infatti, l'esecuzione di tutti i test può richiedere diverso tempo e può essere superfluo: a seconda dello scopo per cui si utilizza il Toolbox, si può quindi verificare il funzionamento solo dei metodi necessari.

Va sottolineato inoltre che non è necessario che tutti i test restituiscano un responso positivo: è sufficiente che funzionino correttamente i metodi legati al tipo di analisi che si vuole effettuare. Inoltre, per quanto riguarda i solver Gurobi e Tomlab, potrebbero verificarsi degli errori legati ad una sbagliata parametrizzazione dei risolutori.

## B.2.2 Caricare un modello in Matlab

Se si desidera caricare un modello di cui si ha a disposizione un file in formato SBML, è sufficiente spostarsi nella cartella contenente il file e invocare il metodo:

```
model = readCbModel ([filename]),
```

che restituisce un modello sottoforma di struttura, contenente tutti i campi necessari per descrivere la rete metabolica.

Se non viene esplicitato nessun argomento, il metodo invita l'utente a scegliere il file utilizzando un box di dialogo.

Alternativamente è possibile caricare un modello descritto in formato Excel. A tal proposito è possibile invocare il metodo:

```
model = xls2model ([filename], [biomassRxnEquation]),
```

in cui è necessario esplicitare almeno il primo argomento. Va evidenziato tuttavia che l'esecuzione di tale comando causa un warning:

*"Warning: xls2model IS NOT SUPPORTED BY THE openCOBRA CORE TEAM AND WILL BE MOVED FROM THE CORE IN THE NEAR FUTURE"*

Quindi in futuro sarà preferibile utilizzare il formato SBML.

Nel nostro caso però, poiché non è stato possibile recuperare un file SBML completo per la rete di nostro interesse, è stato creato il modello in formato 'xlsx', caricato in Matlab tramite il comando sopraccitato.

### B.2.3 Salvare un modello in un file

COBRA Toolbox permette di salvare la struttura corrispondente ad un modello in un file di testo o in formato SBML. Nel caso si utilizzi Microsoft Windows è possibile salvare il modello anche in formato Excel.

Il comando da utilizzare a tale scopo è

```
writeCbModel(model, format, [fileName], [compSymbolList], [compNameList], [SBMLLevel], [SBMLVersion]);
```

Ricordiamo che il Toolbox è in grado di generare file SBML di livello 2, versione 1 oppure 4.

Va inoltre evidenziato che questo non è il solo comando disponibile per ottenere questo risultato: metodi simili possono essere selezionati all'interno della cartella 'io' del Toolbox.

### B.2.4 Apportare delle modifiche ai modelli COBRA Toolbox

Dopo aver caricato un modello in Matlab, questo viene salvato come variabile di tipo 'struc' ed è ovviamente modificabile: apportando le dovute modifiche, si possono simulare condizioni diverse per eseguire differenti tipi di analisi. Le principali modifiche di tipo strutturale che si possono apportare ad un modello sono le seguenti:

#### B.2.4.1 Modificare i limiti delle reazioni

Questa funzionalità è molto utile se si considera che assegnare un valore negativo come lower bound ad una reazione significa assegnare un valore massimo alla quantità di flusso che può entrare nella rete. Di conseguenza, soprattutto per quanto riguarda i flussi di scambio, la scelta del lower bound è molto importante. Viceversa, l'upper bound diventa una grandezza importante per i flussi di scambio riferiti ai prodotti della rete metabolica. Inoltre, i lower ed upper bound sono molto utili per bloccare determinate reazioni (in tal caso vanno settati entrambi a 0) o per vincolare alcuni flussi ad essere maggiori di zero (in tal caso il lower bound o l'upper bound saranno diversi da zero).

Per effettuare tali modifiche si può operare direttamente sulla variabile 'model':

```
model.lb(indiceReazione) = [valore];    model.up(indiceReazione) = [valore],
```

oppure si può utilizzare la funzione disposta dal Toolbox:

```
model = changeRxnBounds(model, rxnNameList, value, boundType);
```

dove:

*rxnNameList* è un lista di IDs delle reazioni, che corrispondono agli identificativi salvati in *model.rxns*;

*value* è un numero floating point, che rappresenta il valore che si vuole assegnare al bound;

*boundType* specifica il tipo di bound che si vuole modificare: lower ('l'), upper ('u') o entrambi ('b').

#### B.2.4.2 Aggiungere nuove reazioni

COBRA Toolbox permette di aggiungere facilmente nuove reazioni all'interno della rete considerata, utilizzando la specifica funzione:

```
[model] = addReaction(model, rxnName, metaboliteList, stoichCoeffList, [revFlag], [lowerBound], [upperBound],
```

```
[objCoeff], [subsystem], [grRule], [geneNameList], [systNameList], [checkDuplicate]);
```

```
model = addReaction(model, rxnName, rxnFormula);
```

Nel primo caso, gli argomenti obbligatori sono:

*model* modello su cui si vuole intervenire;

*rxnName* nome abbreviato della reazione che si vuole inserire (ad esempio 'ACALD'). Altrimenti si può inserire una lista {'abbr', 'name'};

*metaboliteList* lista dei metaboliti che partecipano alla reazione;

*stoichCoeffList* lista dei coefficienti stechiometrici relativi a ciascun metabolita (i reagenti avranno un coefficiente negativo ('-') e i prodotti uno positivo ('+'));

Si possono eventualmente aggiungere degli input non obbligatori:

*revFlag* booleano che indica se la reazione è reversibile (*true*, default) o irreversibile (*false*);

*lowerBound* valore del lower bound (default: 0 oppure  $-vMax$ );

*upperBound* valore dell'upper bound (default:  $vMax$ );

*objCoeff* (default: 0) se impostato a 1 indica che tale reazione è una reazione obiettivo;

*subsystem* sottosistema a cui appartiene la reazione (default: '.');

*grRule* regole booleane che descrivono la relazione tra la reazione inserita e i geni corrispondenti (default: '.');

*geneNameList* lista dei nomi dei geni;

*systNameList* lista di nomi sistematici;

*checkDuplicate* booleano che indica se verificare l'eventuale presenza di un duplicato nella matrice stechiometrica  $S$  (default: *true*);

Nel secondo caso è sufficiente inserire la formula stechiometrica.  
 (Esempio: `model = addReaction(model, 'newRxn1', 'A -> B + 2C')`). Si possono in ogni caso inserire gli input non obbligatori.

### B.2.4.3 Rimuovere alcune reazioni

Allo stesso modo, COBRA Toolbox permette di rimuovere alcune reazioni dalla rete. A tale scopo è sufficiente eseguire la funzione

$$model = removeRxns(model, rxnRemoveList, irrevFlag, metFlag);$$

dove gli argomenti indicano:

*model* il modello su cui si vuole agire;

*rxnRemoveList* una lista contenente i nomi delle reazioni che si vogliono eliminare;

*irrevFlag* booleano che indica se la reazione è reversibile (*true*) o irreversibile (*false*, default);

*metFlag* booleano che indica se eliminare (*true*, default) oppure no (*false*) i metaboliti che non verrebbero più utilizzati.

### B.2.4.4 Modificare il flusso obiettivo

COBRA Toolbox permette di effettuare diversi analisi, calcolando una distribuzione ottima dei flussi per massimizzare il flusso attraverso determinate reazioni. Per far questo è necessario definire una funzione obiettivo, che consiste in una funzione lineare delle reazioni presenti nella rete analizzata. Per definire tale funzione obiettivo, è necessario utilizzare la funzione offerta dal Toolbox:

$$model = changeObjective(model, rxnNameList, [objectiveCoeff]);$$

in cui:

*rxnNameList* è una stringa oppure una lista contenente le IDs delle reazioni da ottimizzare;

[*objectiveCoeff*] specifica il peso dato a ciascuna reazione da ottimizzare all'interno della funzione obiettivo (default: 1).

Per visualizzare la funzione obiettivo che è stata impostata, è sufficiente eseguire il metodo:

$$objectiveAbbr = checkObjective(model);$$

che restituisce tutte le reazioni (con i pesi corrispondenti) che partecipano alla funzione obiettivo.

### B.2.4.5 Stampare la formula di una reazione

Il Toolbox offre la possibilità di visualizzare la formula stechiometrica di una o più reazioni presenti nella rete: è sufficiente richiamare il metodo:

```
printRxnFormula(model, rxnNameList);
```

dove l'argomento *rxnNameList* indica la o le reazioni di cui si vuole conoscere la formula stechiometrica.

### B.2.5 Visualizzazione del modello

A volte può essere molto utile poter visualizzare la rete metabolica, per avere una comprensione più chiara dei risultati dell'analisi che è stata effettuata. Per poter visualizzare il modello, è necessario fornirsi della rete sottoforma di mappa. Ancora una volta il database di BiGG ci può essere utile, in quanto permette di scaricare informato .txt la mappa di diversi sistemi metabolici. In Figura B.1 possiamo visualizzare tale funzionalità di BiGG.

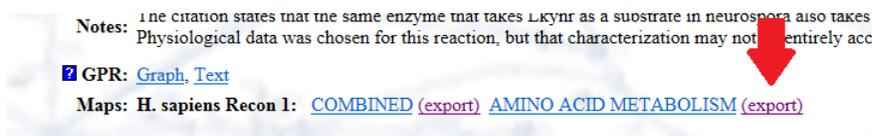


Figure B.1: Esportazione di un rete in formato .txt dal database BiGG

Tale file deve essere salvato nella cartella di lavoro utilizzata in Matlab: in questo modo, per caricare l'oggetto è sufficiente richiamare il comando:

```
map = readCbMap([fileName]);
```

Analogamente al caricamento di un modello, se non viene esplicitato alcun argomento, il Toolbox invita l'utente a selezionare il file map utilizzando un box di dialogo.

Questo comando crea l'oggetto map sotto forma di struttura ('*struc*'), che può essere visualizzato tramite la funzione:

```
drawCbMap(map, [options], [verargin]);
```

Si può scegliere se visualizzare la mappa sottoforma di Figure in ambiente Matlab oppure se salvarla in formato '.svg' (default). Per cambiare il formato di output, è disponibile il metodo:

```
outputFormatOK = changeCbMapOutput(outputFormat),
```

in cui *outputFormat* può essere '*matlab*' oppure '*svg*'.

Di default il metodo *drawCbMap* restituisce l'oggetto con il nome '*target.svg*': tale nome può ovviamente essere modificato, inserendo l'opzione '*FileName*' come argomento.

## B.3 Gap filling

Poichè non si ha ancora una conoscenza completa delle reti metaboliche, spesso può capitare che i modelli contengano dei gap, cioè delle parti mancanti. Più precisamente, con il termine 'gap' si indica la mancanza di informazioni biochimiche che possono spiegare eventuali differenze tra le predizioni che si possono effettuare tramite il modello in silico e i dati sperimentali. Solitamente questi gap sono costituiti da reazioni che facilitano la conversione di un metabolita disponibile nel modello in un altro necessario per raggiungere un certo obiettivo.

COBRA Toolbox mette a disposizione due metodi per cercare di identificare eventuali gap della rete:

1. *detectDeadEnds*;
2. *gapFind*.

### B.3.1 Ricerca di metaboliti che partecipano ad un'unica reazione:

Applicando il primo dei metodi sopracitati:

$$outputMets = detectDeadEnds(model, removeExternalMets);$$

la funzione va alla ricerca (valutando la matrice stechiometrica  $S$ ) dei metaboliti che partecipano ad un'unica reazione. In output infatti *outputMets* contiene gli indici di tali metaboliti.

L'input opzionale *removeExternalMets*, se settato a *'true'* rimuove quei metaboliti che riguardano le reazioni di scambio (cioè quei metaboliti che partecipano nelle reazioni solo con se stessi). Di default questa opzione è settata a *'false'*.

Va sottolineato però che, ispezionando la sola matrice  $S$ , non è garantito che vengano individuati tutti i gap presenti nella rete.

### B.3.2 Individuazione di tutti i gap del modello:

Poichè il metodo precedente non garantisce l'individuazione di tutti i gap, è possibile utilizzare il metodo:

$$[allGaps, rootGaps, downstreamGaps] = gapFind(model, findNCgaps, verbFlag);$$

per individuare tutte le informazioni mancanti. Tale metodo infatti restituisce:

*allGaps*: gli indici dei metaboliti riguardanti un gap;

*rootGaps*: lista di metaboliti che non possono essere prodotti;

*downstreamGaps*: lista di metaboliti che vengono prodotti in una reazione che richiede altri metaboliti che non possono essere prodotti.

L'ingresso opzionale *findNCgaps*, che di default vale *'false'*, se settato a *'true'* permette di trovare anche i metaboliti che non vengono mai consumati.

### B.3.3 Identificazione di candidati per colmare i gap:

Oltre ai metodi appena descritti, che permettono solo di individuare eventuali gap del modello, COBRA Toolbox mette a disposizione un algoritmo per identificare le reazioni candidate a colmare questi gap individuati. Il metodo proposto dal Toolbox individua il numero minimo di reazioni, appartenenti ad un database universale, che sono necessarie affinché il modello metabolico possa crescere in un substarto specifico. Il metodo in esame è il seguente:

$$[solution] = growthExpMatch(model, KEGGFilename, compartment, iterations, dictionary, logFile, threshold);$$

dove gli input obbligatori sono:

*KEGGFilename*: nome del file contenente il database di Kegg (il file dev'essere in formato '.lst' e contenere una lista di reazioni così strutturata: ogni reazione della lista dev'essere seguita da una colonna contenente la formula della reazione stessa). Tale file si può scaricare dal database disponibile al sito: <http://www.genome.jp/kegg>;

*compartment*: stringa che rappresenta il compartimento da cui generare le reazioni di scambio;

*iterations*: numero di iterazioni da effettuare;

*dictionary*: matrice di dimensione  $n \times 2$  che mappa ciascun metabolita con la sua Kegg ID (prima colonna: abbreviazione del nome del metabolita riferito al database di BiGG, seconda colonna: Kegg ID del metabolita).

Gli input opzionali permettono di settare altri parametri:

*logFile*: nome del file in cui viene stampata la soluzione (id est il nome della reazione aggiunta e valore del flusso attraverso tale reazione). Di default vale '*GEMLog.txt*';

*threshold*: minimo valore che può assumere la funzione biomassa affinché si possa considerare che il modello stia crescendo.

Il metodo restituisce come soluzione (*solution*) quella ottenuta tramite un problema MILP. Vengono riportati la soluzione continua, la soluzione intera, il valore dell'obiettivo, lo stato, la soluzione completa e le reazioni importate.

Per visualizzare il risultato appena ottenuto, stampando il file generato, si può usare il metodo:

```
printSolutionGEM(MatricesSUX, solution, logFile, itNum);
```

dopo aver precedentemente creato la matrice *MatricesSUX* attraverso il metodo

```
MatricesSUX = generateSUXMatrix(model, dictionary, KEGGFilename, KEGGBlackList, compartment, addModel).
```

## B.4 Ottimizzazione di flussi obiettivo tramite Flux Balance Analysis (FBA)

Come è stato più volte ribadito, uno dei vantaggi maggiori di una rappresentazione di un sistema metabolico sottoforma di modello è quello di poter effettuare la Flux Balance Analysis (la cui descrizione è stata ampiamente trattata nel Capitolo 4). Ricordiamo che tale analisi viene effettuata in stato stazionario. Solitamente, essa viene utilizzata per ottimizzare la crescita di una cellula (massimizzare la biomassa) e poter calcolare una distribuzione di flussi attraverso le reazioni, tale da assicurare tale ottimizzazione. La FBA però può essere utilizzata per ottimizzare qualsiasi combinazione di reazioni presenti nella rete, senza che sia necessariamente richiesta la definizione della funzione biomassa. per poter effettuare tale analisi è necessario però aver definito:

1. la funzione obiettivo;
2. i limiti (upper e lower bound) delle reazioni di scambio, che definiscono il mezzo in cui si vuole testare il modello (ad esempio, settando a zero il lower bound della reazione corrispondente all'ingresso di ossigeno nella rete, si può eseguire l'analisi in assenza di ossigeno. In caso contrario, occorre verificare che tale limite abbia un valore inferiore allo zero. Analogamente, è importante assicurarsi che l'upper bound relativo al prodotto che si vuole massimizzare abbia un valore positivo, per renderne possibile la crescita e la fuoriuscita dalla rete).

Per implementare la Flux Balance Analysis, COBRA Toolbox mette a disposizione due metodi principali: la FBA standard e la geometric FBA, che verrà descritta appena sotto.

### B.4.1 FBA standard

La Flux Analysis Standard, punto focale di questa tesi, è stata già ampiamente descritta precedentemente: in questa sede perciò ci limitiamo ad osservare come possa essere implementata mediante il Toolbox a nostra disposizione.

Il comando da eseguire è il seguente:

```
[solution] = optimizeCbModel(model, [osenseStr], [minNorm], [allowLoops]);
```

in cui:

*osenseStr* può valere '*max*' o '*min*' a seconda che si voglia massimizzare o minimizzare il flusso obiettivo (default: '*max*');

*minNorm* (default: 0) se diverso da zero indica che si vuole ottenere una soluzione minimizzando la presenza dei loop;

*allowLoop* (default: 0) se diverso da zero, usa un algoritmo per eliminare i loop (va considerato che questa funzione può consumare più tempo).

Questo metodo restituisce il risultato dell'analisi FBA sottoforma di struttura, contenente i seguenti campi:

*solution.f*: il valore del flusso obiettivo;

*solution.x*: rappresenta la soluzione vera e propria, ovvero la distribuzione dei flussi all'interno della rete che massimizza l'obiettivo;

*solution.y*: rappresenta lo 'shadow price', per ciascun metabolita, ovvero un indice di quanto l'aggiunta di quel particolare metabolita farebbe aumentare o diminuire il flusso da ottimizzare;

*solution.w*: rappresenta il 'reduced cost' di ogni reazione, ovvero un indice di quanto ciascuna reazione incide nel risultato;

*solution.stat*: valore booleano che indica se è stata trovata una soluzione ammissibile. Se il suo valore è pari a 1, significa che l'ottimizzazione è andata a buon fine.

## B.4.2 Geometric FBA

Come descritto nel corpo della tesi, la Geometric FBA è un algoritmo alternativo alla standard FBA, che permette di ottenere una soluzione unica al problema di ottimizzazione soggetto ai vincoli stechiometrici. Tale soluzione può essere considerata in un certo senso minimale e centrale (da un punto di vista geometrico) rispetto a tutto lo spazio delle soluzioni ottime possibili, perciò essa si può considerare rappresentativa per tutte le soluzioni. Inoltre tale risultato è riproducibile, in quanto non dipende dal tipo di risolutore utilizzato.

COBRA Toolbox permette di implementare tale analisi, mediante la funzione:

$$flux = geometricFBA(model, [varargin]);$$

che restituisce il vettore '*flux*', contenente la distribuzione ottima (unica e centrale) dei flussi, che garantiscono la massimizzazione del flusso obiettivo.

## B.4.3 Visualizzazione dei risultati

Dopo aver creato l'oggetto *map*, come spiegato nei punti precedenti, il Toolbox mette a disposizione dei metodi per visualizzare i risultati ottenuti dalla standard FBA o dalla Geometric FBA: poter visualizzare i flussi direttamente nella rete permette una maggiore comprensione dei risultati. Il comando da eseguire a tal fine è il seguente:

$$drawFlux(map, model, [options], [varargin]);$$

dove *[options]* è una struttura contenente i seguenti parametri opzionali:

*lb*: limite inferiore a cui arrotondare i valori minori dei flussi;

*ub*: limite superiore a cui arrotondare i valori maggiori dei flussi;

*colorScale*: colori da utilizzare nella visualizzazione;

*zeroFluxWidth*: larghezza delle frecce che corrispondono ai flussi nulli;

*zeroFluxColor*: colore delle frecce che corrispondono ai flussi nulli;

*fileName*: nome del file in output;

*rxnDirMultiplier*: valore di scala delle frecce che indicano la direzione del flusso.

Il metodo, oltre a produrre il file contenente la mappa richiesta, restituisce una struttura '*options*', contenente i parametri opzionali utilizzati.

## B.5 Classificazione di geni e reazioni mediante pFBA

La parsimonious FBA (pFBA), come descritto ampiamente nel corpo della tesi, è un approccio che si utilizza per classificare le reazioni e i geni ad esse associati, basandosi su un’ottimizzazione a due livelli, che permette contemporaneamente di massimizzare il flusso obiettivo (ad esempio il rate di crescita della cellula) e di minimizzare la quantità di materiale enzimatico necessario per ottenere il risultato richiesto dall’ottimizzazione del flusso obiettivo.

Il metodo proposto da COBRA Toolbox permette di implementare la pFBA ottenendo una classificazione di geni e reazioni in 6 classi (la sesta classe non era stata descritta nella sezione introduttiva, ma la presentiamo in questa sede perchè viene implementata nel Toolbox):

1. Essential genes: geni necessari per il raggiungimento dell’obiettivo;
2. pFBA-optima genes: geni NON essenziali che contribuiscono all’ottimizzazione del flusso richiesto e alla minimizzazione degli enzimi associati ai flussi;
3. Geni enzimaticamente sotto-efficienti (enzymatically less-efficient genes: ELE), che richiedono una maggiore quantità di flusso attraverso gli step enzimatici della rete piuttosto che attraverso pathways alternative che permettono comunque il raggiungimento dello stesso obiettivo;
4. Geni metabolicamente sotto-efficienti (metabolically less-efficient genes: MLE), che richiederebbero una riduzione del flusso di ottimizzazione;
5. pFBA no flux genes, che sono incapaci di trasportare un flusso nelle condizioni sperimentali;
6. Blocked genes, che sono associati a reazioni che non possono trasportare un flusso in nessuna condizione (“blocked” reactions).

Il metodo volto ad ottenere tale classificazione è il seguente:

$$[GeneClasses, RxnClasses, modelIrrevFM] = pFBA(model, [varargin]);$$

che restituisce:

*GeneClasses*: una lista contenente tutti i geni classificati nelle classi sopra descritte;

*RxnClasses*: una lista contenente tutte le reazioni classificate nelle classi sopra descritte;

*modelIrrevFM*: un modello che contiene i vincoli utilizzati per la minimizzazione dei flussi.

Per quanto riguarda gli argomenti opzionali in ingresso, alcuni sono inerenti all’analisi, altri alla visualizzazione dei risultati (questi ultimi verranno presentati nel paragrafo seguente). Per quanto riguarda l’analisi, *[varargin]* contiene i seguenti parametri:

*'geneoption'*: se posto uguale a 1 (default) impone la minimizzazione della somma dei flussi attraverso le reazioni associate ai geni; se posto uguale a 0 impone la minimizzazione della somma di tutti i flussi della rete.

*'tol'*: tolleranza (default:  $1e - 6$ );

*'skipclass'*: se posto uguale ad 1 non classifica i geni e le reazioni e restituisce solamente il modello con il flusso minimo settato come upper bound.

### B.5.1 Visualizzazione dei risultati

Dopo aver creato l’oggetto mappa, come descritto precedentemente, è possibile visualizzare i risultati della classificazione direttamente nella mappa. Per poter fare questo è sufficiente inserire i parametri opzionali quando si richiama il metodo *'pFBA'*. In particolare, per quanto riguarda la visualizzazione, *[varargin]* contiene i seguenti parametri:

*'map'*: oggetto 'map' ottenuto dal metodo *readCbMap.m*;

*'mapoutname'*: nome del file da associare alla mappa.

## B.6 Robustness Analysis

Per studiare le proprietà di una rete metabolica, si può effettuare una Robustness Analysis, anch'essa basata sulla Flux Balance Analysis. Tale metodologia consente di variare il flusso attraverso una reazione specifica e calcolare il valore del flusso obiettivo come funzione di tale flusso variante. Questo metodo può essere utilizzato a vari scopi, ad esempio per esaminare l'effetto dei nutrienti nei confronti dell'obiettivo scelto da ottimizzare. Altrimenti può essere molto utile per capire come le reazioni si influenzino tra di loro.

Il metodo implementato da COBRA Toolbox per studiare l'andamento del flusso obiettivo in funzione di una singola reazione della rete è il seguente:

```
[controlFlux, objFlux] = robustnessAnalysis(model, controlRxn, nPoints,  
  
plotResFlag, objRxn, objType);
```

che richiede in ingresso obbligatoriamente:

*model*: il modello analizzato;

*controlRxn*: la reazione di interesse di cui si vuole far variare il flusso.

Gli argomenti opzionali presentati hanno il seguente significato:

*nPoints*: numero di punti utilizzati per mostrare il risultato (default: 20);

*plotResFlag*: booleano che indica se si vuole visualizzare o meno il risultato (default: *true*);

*objRxn*: reazione obiettivo da massimizzare (di default viene scelta quella definita dal modello);

*objType*: permettere di scegliere se massimizzare ('*max*', default) o minimizzare ('*min*') il flusso obiettivo.

Oltre a restituire un'immagine in cui è presentato il flusso obiettivo in funzione del flusso di controllo, il metodo restituisce anche i valori del flusso obiettivo (*objFlux*) in relazione ad ogni modifica del flusso di controllo (*controlFlux*), che viene fatto variare tra il suo valore minimo e massimo consentiti dalla topologia della rete e dai vincoli.

## B.7 Flux Variability Analysis (FVA)

Poiché la FBA restituisce una sola delle possibili distribuzioni dei flussi che corrispondono ad uno stesso valore massimo del flusso obiettivo, può essere interessante e molto utile analizzare l'intero range dei valori dei flussi che ciascuna reazione della rete può assumere, mantenendo il flusso obiettivo ad una certa percentuale del suo valore massimo (calcolato tramite FBA).

Per determinare i flussi minimo e massimo attraverso ciascuna reazione, nelle condizioni dettate dalla FBA, COBRA Toolbox offre la seguente funzione:

```
[minFlux, maxFlux, Vmin, Vmax] = fluxVariability(model, optPercentage, osenseStr,  
  
rxnNameList, verbFlag, allowLoops);
```

in cui l'unico argomento necessario è il modello di interesse, mentre gli altri argomenti (opzionali) permettono di gestire l'analisi nel modo desiderato:

*optPercentage*: valore percentuale del flusso obiettivo che si vuole considerare: si pone attenzione solo alle soluzioni che garantiscono un valore del flusso obiettivo pari a tale percentuale del flusso massimo. (Default: 100);

*osenseStr*: tipo di ottimizzazione ('*min*' o '*max*'). (Default: '*max*');

*rxnNameList*: lista di reazioni che vengono prese in analisi dalla FVA (di default si considerano tutte le reazioni della rete);

*verbFlag*: se posta uguale a *'true'* stampa il risultato (default: *'false'*);

*allowLoops*: indica se permettere la considerazione dei loop nella soluzione (default: *'true'*);

Il metodo restituisce il minimo (*minFlux*) e il massimo (*maxFlux*) valore dei flussi, sottoforma di vettore, e due output opzionali:

*Vmin*: una matrice di flussi (ogni flusso è una colonna), dove ogni colonna è una minimizzazione separata;

*Vmax*: una matrice di flussi (ogni flusso è una colonna), dove ogni colonna è una massimizzazione separata.

### B.7.1 Visualizzazione dei risultati

Il Toolbox permette di osservare i risultati della FVA direttamente sulla mappa (l'oggetto mappa dev'essere caricato in Matlab, come spiegato precedentemente) della rete metabolica. Il comando da utilizzare è il seguente:

```
drawFluxVariability(map, model, minFlux, maxFlux, [options]);
```

La struttura [*options*] permette di settare i parametri riguardanti i colori e le dimensioni dei nodi e degli archi. Di default, la funzione attribuisce i seguenti colori alle reazioni:

- verde: reazioni reversibili bidirezionali;
- magenta: reazioni reversibili unidirezionali, che trasportano il flusso solo nella direzione diretta;
- azzurro: reazioni reversibili unidirezionali, che trasportano il flusso solo nella direzione inversa;
- blu: reazioni irreversibili.

## B.8 Analisi delle caratteristiche topologiche della matrice stechiometrica S: metabolite connectivity e reaction participation

COBRA Toolbox mette a disposizione dei metodi per analizzare il modello dal punto di vista topologico, osservando le proprietà che si possono ricavare direttamente ispezionando la matrice stechiometrica *S*.

### B.8.1 Visualizzazione della matrice S

Per visualizzare la matrice stechiometrica *S*, il Toolbox mette a disposizione il metodo

```
spy(model.S);
```

che plotta tale matrice rappresentando ogni entry diversa da zero con un pallino.

### B.8.2 Determinazione del numero di reazioni a cui partecipa ciascun metabolita

Per determinare il numero di reazioni a cui partecipa ciascun metabolita è necessario convertire la matrice stechiometrica *S* in una matrice binaria *S<sub>bin</sub>* sostituendo tutti i valori diversi da zero con il valore '1', che indica che il metabolita corrispondente partecipa alla relativa reazione (viene meno quindi l'informazione relativa al valore esatto del coefficiente stechiometrico). I comandi da utilizzare a tale scopo sono i seguenti:

```
Sbin = zeros(size(model.S));  
Sbin(find(model.S)) = 1;
```

A questo punto, per determinare il numero di reazioni a cui partecipa ciascun metabolita è sufficiente sommare i valori di ciascuna riga di *S<sub>bin</sub>*:

```
for i = 1 : length(model.mets)  
    metConnectivity(i, 1) = sum(Sbin(i, :));  
end
```

Per visualizzare il risultato in dei grafici, si possono utilizzare i comandi matlab:

```
loglog(sort(metConnectivity,'descend'),'*')
xlabel('metabolite number (rank ordered) - log scale');
ylabel('number of reactions - log scale')
```

Può essere interessante a questo punto osservare il numero di reazioni comuni a diversi metaboliti della rete: a tale scopo occorre calcolare la matrice delle adiacenze dei composti ( $A_{comp}$ ):  $A_{comp} = S_{bin} * S_{bin}^T \in \mathbb{R}^{m \times m}$ , dove  $m$  è il numero dei metaboliti nella rete.

Gli elementi diagonali di  $A_{comp}$  corrispondono al numero di reazioni a cui partecipa ciascun metabolita, valore che abbiamo già calcolato precedentemente (questo è quindi un metodo alternativo per ricavare la *metConnectivity*):

```
A_comp = S_bin * S_bin';
metConnectivity = diag(A_comp);
```

Gli elementi al di fuori della diagonale invece rappresentano proprio il numero di reazioni a cui partecipano assieme i due diversi metaboliti (riga e colonna).

### B.8.3 Reaction Participation

In questo contesto, la Reaction Participation è definita come il numero di metaboliti che partecipa a ciascuna reazione. Per determinare tale valore, relativo a ciascuna reazione della rete, è sufficiente sommare gli elementi per colonna della matrice binaria  $S_{bin}$ :

```
for i = 1 : length(model.rxns)
    rxnParticipation(i,1) = sum(S_bin(:,i));
end
```

Se vogliamo ora osservare quanti metaboliti sono comuni a due diverse reazioni, calcoliamo la matrice delle adiacenze delle reazioni ( $A_{rxns}$ ):  $A_{rxns} = S_{bin}^T * S_{bin} \in \mathbb{R}^{n \times n}$ , dove  $n$  è il numero di reazioni della rete. Analogamente al caso dei metaboliti, gli elementi diagonali di  $A_{rxns}$  corrispondono alle Reaction Participation di ciascuna reazione, mentre gli elementi fuori diagonale indicano il numero di metaboliti comuni alle due reazioni di riga e colonna.

```
A_rxns = S_bin' * S_bin;
rxnParticipation = diag(A_rxns);
```

### B.8.4 Correlazione tra essenzialità delle reazioni e metabolite connectivity

Per verificare se esiste qualche correlazione tra la caratteristica di essenzialità di una reazione (intesa come necessità di attivazione di tale reazione per poter massimizzare il flusso obiettivo) e la connettività tra diversi metaboliti, è possibile osservare cosa succede al flusso obiettivo se si rimuovono tutte le reazioni a cui partecipa un determinato metabolita.

Per trovare tutte le reazioni associate ad un metabolita è sufficiente osservare le righe della matrice  $S$ : si bloccano quindi tutti i geni che corrispondono a tali reazioni e si esegue la FBA per predire se è possibile la massimizzazione del flusso obiettivo (es. la biomassa). Il rapporto tra la crescita e la connettività si può considerare come un indice che lega l'essenzialità delle reazioni con il valore di connettività che caratterizza i metaboliti che vi partecipano.

Il codice matlab per eseguire questo calcolo e plottare il risultato può essere il seguente:

```
growthDel = zeros(length(model.mets),1);
for i = 1 : length(model.mets)
    rxnID = find(model.S(i,:));
    for j = 1 : length(rxnID)
```

```

    modelDel = deleteModelRxn(model, model.rxns(rxnID(j)));
    FBAolutionDel = optimizeCbModel(modelDel);
    if FBAolutionDel.f <= 1e - 6
        growthDel(i, 1) = growthDel(i, 1) + 1;
    end
end
end
lethalityFraction = growthDel./metConnectivity;
semilogx(metConnectivity, lethalityFraction, '*')
xlabel('metabolite connectivity - log scale');
ylabel('average lethality fraction');

```

## B.9 Sampling delle distruzione di flusso

La Flux Balance Analysis può generare una singola distribuzione di flussi che corrisponde all'ottimizzazione di un obiettivo cellulare, sotto determinate condizioni. Un approccio alternativo che si può utilizzare consiste nel caratterizzare l'intero spazio delle distribuzioni di flusso che sono ammesse dai vincoli stechiometrici e dai lower ed upper bound dei diversi flussi. COBRA Toolbox implementa un metodo basato sul campionamento uniforme con il metodo di Monte Carlo. L'insieme delle distribuzioni di flusso ottenute con il metodo proposto dal Toolbox permette di analizzare vari aspetti del modello in esame, ad esempio quale sia il flusso più probabile attraverso ciascuna reazione sotto determinati vincoli e quanto siano correlate tra di loro le diverse reazioni della rete metabolica.

Uno dei metodi disponibili nel Toolbox è il seguente:

```
[modelSampling, samples] = sampleCbModel(model, sampleFile, samplerName, options);
```

i cui ingressi necessari sono:

*model*: il modello in analisi;

*sampleFile*: i nomi dei file da usare in output per il campionamento;

mentre gli input opzionali permettono di definire più nel dettaglio il campionamento che si vuole effettuare:

*samplerName*: nome dell'algoritmo di campionamento da utilizzare (default: 'ACHR');

*options.nWarmupPoints*: numero di 'warmup points' (default: 5000);

*options.nFiles*: numero di file in output (default: 10);

*options.nPointsPerFile*: numero di punti per ogni file (default: 1000);

*options.nStepsPerPoint*: numero di step di campionamento per ogni punto (default: 200);

*options.nFilesSkipped*: numero di file in output saltati quando vengono caricati i punti (default: 2);

*options.removeLoopsFlag*: se posta uguale a 'true', rimuove i loop dal modello (default: 'false');

*options.removeLoopSamplesFlag*: se posta uguale a 'true' (default) rimuove i dati campionati relativi alle reazioni coinvolte in loop.

Il metodo restituisce il modello ripulito, utilizzato nel campionamento (*modelSampling*) e i campioni dello spazio delle soluzioni ottenute mediante campionamento uniforme (*samples*), in forma matriciale.

A questo punto è possibile plottare gli istogrammi delle distribuzioni di flusso. Si possono visualizzare le distribuzioni di ciascuna reazione in modo indipendente, oppure le distribuzioni di coppie di reazioni, così da analizzarne la correlazione. Questa rappresentazione permette quindi di visualizzare le interazioni tra le reazioni a due a due: se tali reazioni fossero perfettamente correlate, i punti del plot si sistemerebbero lungo una retta. Al contrario, due flussi sono pressoché indipendenti l'uno dall'altro, se i punti sono sparsi.

Il metodo da utilizzare è il seguente:

*sampleScatterMatrix(rxnNames, model, sample, nPoints, dispRFlag, rxnNames2);*

il cui input necessario *rxnNames* (oltre al modello (*model*) e ai campioni (*sample*), ottenuti con il metodo precedentemente descritto) definisce una lista di reazioni da plottare.

Con gli input opzionali si possono settare diversi parametri:

*nPoints*: numero di punti di campionamento da plottare (default: 100);

*fontSize*: dimensione delle label (di default è calcolato in base al numero di reazioni da plottare);

*dispRFlag*: se posto uguale a *'true'* mostra i coefficienti di correlazione (default: *'false'*);

*rxnNames2*: secondo insieme (opzionale) di reazioni.

### B.9.1 Riconoscere moduli di reazioni nelle reti metaboliche

Come abbiamo accennato precedentemente, il campionamento può essere utilizzato per determinare la dipendenza tra due reazioni: questa informazione può essere utile per definire dei moduli di reazioni all'interno dell'intera rete (set di reazioni correlate tra loro). A partire da un insieme di campioni (cioè dall'output del metodo *sampleCbModel*), è possibile identificare tali insiemi di reazioni correlate, tramite la funzione del Toolbox:

*[sets, setNumber, setSize] = identifyCorrelSets(model, samples, corrThr, R);*

i cui argomenti opzionali permettono di settare la soglia minima di correlazione ( $corrThr = R^2$ , che di default vale  $1 - 1e - 8$ ) e il coefficiente di correlazione ( $R$ ).

Il metodo restituisce quindi:

*sets*: array ordinato di insiemi trovati (in ordine decrescente per grandezza);

*setNumber*: lista contenente il numero di set a cui appartiene ciascuna reazione);

*setSize*: lista delle dimensioni degli insiemi trovati.

## B.10 Simulazione di delezioni di geni

Nei modelli in silico si può facilmente simulare l'effetto nella rete in seguito alla delezione di particolari geni. Per poter utilizzare tale funzionalità offerta da COBRA Toolbox, è necessario che nel modello sia definita anche la GPR map, ovvero le relazioni esistenti tra i geni e le reazioni a cui partecipano. (Dalla mappa GPR viene definita la matrice gene-reaction e il set di regole booleane che identificano le relazioni tra ciascuna reazione e i geni della rete).

Questo Toolbox permette di eseguire due tipi di analisi riguardanti la simulazione di delezione dei geni:

1. Single essential gene deletions (delezione di un singolo gene);
2. Pairs of syntetic lethal genes (delezione di coppie di geni).

I risultati che si possono osservare in relazione a tali KnockOut sono tre.

- flusso obiettivo invariato;
- flusso obiettivo ridotto;
- flusso obiettivo nullo (il KnockOut è letale).

### B.10.1 Essential gene study

Per simulare la delezione di un singolo gene è sufficiente richiamare il metodo:

$$[grRatio, grRateKO, grRateWT, hasEffectdelRxns, fluxSolution] = \\ = singleGeneDeletion(model, method, geneList, verbFlag);$$

che richiede come unico input obbligatorio il modello in analisi. I vincoli opzionali permettono di restringere lo studio a degli insiemi di geni di interesse e/o di settare il metodo da utilizzare:

*method*: può valere 'FBA' (default), 'MOMA' o 'lMOMA';

*geneList*: lista di geni da eliminare (uno alla volta) (default: tutti i geni);

*verbFlag*: se posto uguale a 'true' visualizza l'output (default: 'false').

In uscita tale funzione ci restituisce diverse informazioni interessanti:

*grRatio*: rapporto tra il rate di crescita (o dell'obiettivo cellulare impostato) del modello mutato e del modello originale (wild type);

*grRateKO*: rate di crescita nel modello privato del gene (1/h);

*grRateWT*: rate di crescita nel modello wild type (1/h);

*hasEffect*: indica se la delezione del gene ha comportato la rimozione di qualche reazione della rete;

*delRxns*: lista delle reazioni rimosse a causa di ogni gene KO (KnockOut);

*fluxSolution*: flussi ottenuti da FBA/MOMA/lMOMA nel modello mutato per ogni gene rimosso.

Con questa analisi è possibile capire facilmente per quali geni il KnockOut diventa letale per l'ottimizzazione in corso: i geni che, una volta bloccati, portano a zero il flusso obiettivo, sono quindi da considerarsi essenziali.

### B.10.2 Synthetic lethal study

Per analizzare come si comporta la rete in seguito al KnockOut di coppie di geni, è sufficiente eseguire il comando:

$$[grRatioDble, grRateKO, grRateWT] = \\ = doubleGeneDeletion(model, method, geneList1, geneList2, verbFlag);$$

dove:

*method*: può valere 'FBA' (default), 'MOMA' o 'lMOMA';

*geneList1*: lista di geni da eliminare, che corrisponde a *model.genes* (default: tutti i geni);

*geneList2*: lista di geni che corrisponde ad un secondo set di geni da eliminare (default: tutti i geni);

*verbFlag*: se posto uguale a 'true' visualizza l'output (default: 'false').

La chiamata di tale funzione restituisce:

*grRatioDble*: rapporto tra il rate di crescita (o dell'obiettivo cellulare impostato) del modello mutato e del modello originale (wild type);

*grRateKO*: rate di crescita nel modello privato dei due geni (1/h);

*grRateWT*: rate di crescita nel modello wild type (1/h).

## B.11 Metabolic engineering

Una delle principali innovazioni della seconda versione del Toolbox è quella di poter effettuare diversi tipi di Metabolic engineering. In particolare vengono implementati tre metodi:

1. OptKnock;
2. OptGene;
3. GDLS.

### B.11.1 OptKnock

Per eseguire l'algoritmo di OptKnock, al fine di determinare un insieme di reazioni da bloccare per realizzare la doppia ottimizzazione dell'obiettivo cellulare e della produzione di particolari metaboliti all'interno della rete, occorre utilizzare il metodo:

$$[optKnockSol, bilevelMILPproblem] = OptKnock(model, selectedRxnList, options, constrOpt, prevSol, verbFlag, solFileNameTmp);$$

i cui ingressi obbligatori sono solamente il modello e il primo campo della struttura *options*: *options.targetRxn* definisce il flusso target da ottimizzare.

Per effettuare un'analisi più adeguata alle particolari esigenze è bene però settare anche gli input opzionali:

*options.numDel*: numero di KO permessi (default: 5);

*options.numDelSense*: direzione dei vincoli di KO (*G*: greater; *E*: equal; *L*: less). Di default vale '*L*';

*options.vMax*: flusso massimo (default: 1000);

*options.solveOptKnock*: se '*true*' (default) risolve il problema all'interno dell'ambiente Matlab;

*options.gamsFile*: nome del file GAMS in input;

*constrOpt.rxnList*: lista di reazioni che si vuole vincolare;

*constrOpt.values*: valori cu vincolare le reazioni desiderate;

*constrOpt.sense*: verso dei vincoli appena assegnati (*G/E/L*);

*prevSol*: soluzione precedente;

*solFileNameTmp*: nome del file su cui salvare le soluzioni temporanee.

Il metodo restituisce la struttura *optKnockSol*, contenente i campi *rxnList*, contenente la lista delle reazioni bloccate e *fluxes*, contenente la distribuzione dei flussi ottenuta. Inoltre nella struttura restituita *bilevelMILPproblem*, possiamo visualizzare la struttura del problema OptKnock.

Al fine di valutare correttamente i risultati, è necessario tenere a mente che *OptKnock* utilizza i valori  $-vMax$  e  $vMax$  (se reversibile) oppure 0 e  $vMax$  (se irreversibile) come lower ed upper bound dei flussi: è quindi necessario usare *constrOpt* per vincolare in modo differente le reazioni desiderate.

### B.11.2 OptGene

OptGene viene utilizzato per determinare delle strategie di KnockOut al fine di ottenere una sovrapproduzione di un prodotto specifico. Uno dei suoi maggiori vantaggi consiste nel permettere anche la massimizzazione di una funzione obiettivo NON lineare.

Per eseguire tale funzionalità, COBRA Toolbox mette a disposizione il seguente metodo:

$$[x, population, scores, optGeneSol] = optGene(model, targetRxn, substrateRxn, generxnList, MaxKOs, population);$$

i cui input necessari sono:

*model*: il modello in analisi;

*targetRxn*: stringa contenente il nome della reazione da massimizzare;

*substrateRxn*: specifica la reazione di scambio per la crescita;

*generxnList*: lista che specifica quali geni o reazioni possono essere bloccate;

*MaxKOs*: indica il numero massimo di KnockOut ammessi;

L'input opzionale *population* è una matrice binaria da utilizzare per interrompere la simulazione e riprenderla in seguito.

Il metodo restituisce diversi risultati:

*x*: il miglior punteggio (score) ottenuto tramite le funzioni *optGeneFitness* o *optGeneFitnessTilt*;

*population*: una matrice binaria che rappresenta i set di KnockOut ottenuti. Se si vuole reiterare, è necessario passare questa matrice come input opzionale al metodo *OptGene*, come sopra descritto;

*scores*: array dei punteggi;

*optGeneSol*: struttura che raccoglie i risultati ottenuti dall'invocazione del metodo.

### B.11.3 GDLS

Similmente ad OptKnock e OptGene, l'algoritmo GDLS può essere utilizzato per identificare quali KnockOut effettuare al fine di ottimizzare la produzione di particolari prodotti (obiettivo "sintetico") oltre l'obiettivo cellulare del modello (obiettivo "naturale"). Il metodo offerto dal Toolbox per eseguire tale funzionalità richiede le seguenti specifiche:

```
[gdlSolution, bilevelMILPProblem, gdlSolutionStructs] = GDLS(model, targetRxns, varargin);
```

dove gli unici parametri obbligatori in ingresso sono il modello (*model*) e le reazioni da massimizzare (*targetRxn*) (sottoforma di lista di stringhe). *varargin* rappresenta i parametri opzionali che si possono aggiungere, sia sottoforma di struttura, sia come lista di nomi dei parametri, ciascuno seguito dal suo valore. I parametri opzionali sono i seguenti:

'nbhdz': dimensione di vicinato (default: 1), che corrisponde al parametro *K*, utilizzato nella descrizione del metodo nella parte introduttiva;

'M': numero di path di ricerca (default: 1);

'maxKO': numero massimo di KO (default: 50), corrispondente al parametro *C* usato nella descrizione dell'algoritmo;

'koCost': costo (numero di geni da bloccare) per mettere in KO una reazione, un set di geni o un gene. Può essere definito un costo per ciascun KnockOut (di default sono posti tutti a costo pari a 1);

'selectedRxns': lista di reazioni/geneSet che possono essere bloccate;

'koType': definisce cosa bloccare: reazioni ('rxns'), geneSet ('geneSets') o geni ('genes');

'iterationLimit': massimo numero di iterazioni (default: 70);

'timeLimit': tempo massimo in secondi (default: 252000);

'minGrowth': minimo rate di crescita accettabile.

L'esecuzione del metodo restituisce la soluzione dell'algoritmo GDLS sottoforma di struttura (*gdlSolution*), una struttura contenente il problema bilevel MILP utilizzato per la risoluzione (*bilevelMILPProblem*) e le soluzioni intermedie (*gdlSolutionStructs*), calcolate durante l'esecuzione delle varie ricerche locali effettuate dall'algoritmo.

Rispetto all'esecuzione di OptKnock, che effettua una ricerca globale, tipicamente GDLS impiega meno tempo per restituire i risultati, tuttavia non garantisce che la sua tecnica di ricerca locale porti allo stesso risultato di una soluzione ottima globale.

## B.12 Descrizione della struttura di un file SBML compilabile in COBRA

Un file SBML compilabile in COBRA deve contenere i seguenti dati:

- Unit definitions
- Compartments
- Metabolites (format: '*M\_ <metabolite abbreviation>\_ <compartment id>*')
- Metabolite name, compartment, charge, formula
- Reactions (format: '*R\_ <reaction abbreviation>*')
- Reaction name, reversibility, reaction stoichiometry, gene-protein-reaction (GPR) association, subsystem, E. C. number, lower bound, upper bound, flux value, objective coefficient

Le definizioni delle unità di misura vengono elencate in una sezione del file, chiamata *<listOfUnitDefinitions>*, nel seguente formato (viene riportato un esempio):

```
<listOfUnitDefinitions>
<unitDefinition id=mmol per gDW per hr>
<listOfUnits>
<unit kind=mole scale=-3/>
<unit kind=gram exponent=1/>
<unit kind=second exponent=-1 multiplier= 0.00027777777777777778"/>
</listOfUnits>
</unitDefinition >
</listOfUnitDefinitions>
```

I compartimenti devono essere elencati in una sezione del file chiamata *<listOfCompartments>*, nel seguente formato (viene riportato un esempio):

```
<listOfCompartments>
<compartment id=c name=Cytoplasm/>
<compartment id=e name=ExtraOrganism/>
</listOfCompartments>
```

Il compartimento *id* deve corrispondere all'abbreviazione del compartimento aggiunta ai nomi dei metaboliti. Il nome completo del compartimento è definito usando il nome del parametro.

I metaboliti devono essere elencati nella sezione del file *<listOfSpecies>*, nel formato seguente (viene riportato un esempio):

```
<listOfSpecies>
...
<species id=M_ACALD_c name=Acetaldehyde compartment=c/>
<notes>
<body xmins=http://www.w3.org/1999/xhtml>
<p>FORMULA: C2H4O</p>
<p>CHARGE: 0</p>
</body>
</notes>
...
</listOfSpecies>
```

L'abbreviazione del compartimento deve essere aggiunto alla fine dell'identificativo del metabolita. La formula del metabolita è definita nella sezione *<notes>*

Le reazioni vengono elencate nella sezione *<listOfReactions>*, nel seguente formato (viene riportato un esempio):

```

<listOfReactions>
...
<reaction id=R_LDH name=D-lactate dehydrogenase>
<notes>
<body xmlns=http://www.w3.org/1999/xhtml>
<p>GENE ASSOCIATION: (b1380) or (b2133)</p>
<p>SUBSYSTEM: Pyruvate Metabolism</p>
<p>EC Number: 1.1.1.28</p>
<p>Confidence level:0 </p>
</body>
</notes>
<listOfReactants>
<speciesReference species=M_lac_DASH_D_c/>
<speciesReference species=M_nad_c/>
</listOfReactants>
<listOfProducts>
<speciesReference species=M_h_c/>
<speciesReference species=M_nadh_c/>
<speciesReference species=M_pyr_c/>
</listOfProducts>
<kineticLaw>
<math xmlns=http://www.w3.org/1998/Math/MathML>
<ci> FLUX_VALUE </ci>
</math>
<listOfParameters>
<parameter id="LOWER_BOUND" value="-1000" units="mmol_per_gDW_per_hr"/>
<parameter id="UPPER_BOUND" value="1000" units="mmol_per_gDW_per_hr"/>
<parameter id="FLUX_VALUE" value="0" units="mmol_per_gDW_per_hr"/>
<parameter id="OBJECTIVE_COEFFICIENT" value="0" units="mmol_per_gDW_per_hr"/>
</listOfParameters>
</kineticLaw>
</reaction>
...
</listOfReactions>

```

La mappa di associazioni geni-proteine-reazioni (GPR) e il sottosistema di ciascuna reazione devono essere inserite nella sezione `<notes>`, mentre i reagenti e i prodotti vanno elencati rispettivamente nelle sezioni `<listOfReactants>` e `<listOfProducts>`. I limiti upper bound e lower bound e i coefficienti della funzione obiettivo vanno scritti nella lista dei parametri `<listOfParameters>` all'interno della sezione `<kineticLaw>`.

## Appendix C

### Condizioni iniziali del modello cinetico

Condizioni iniziali del modello cinetico del metabolismo del triptofano utilizzate durante le simulazioni effettuate con il software COPASI.

	concentrazioni iniziali (mmol/l)		concentrazioni iniziali (mmol/l)
3HAA	0	Kyna	1
3HKyn	0	L_ALA	1
5HAc	1	L-Glu	0
5HFKyn	0	L-Kyn	0
5HKyn	0	L-Kyn_ex	0
5HTrp	0	MMTrypta	0
AA	1	MTrypta	0
AcetylCoA	0	NAAD	0
Acms	1	NAcSet	0
AHCys	1	Nactrypta	0
Am6sa	1	NADP	0
AMet	1	NADPH	0.03
AMP	1	NaMN	0
ATP	1	NH4	0
Cin	1	NMSer	0
CO2	1	O2	1
CoaA	1	O2s	0
Dhbpt	1	PPi	0
F5HKyn	0	PRPP	1
FAA	0	Quin	0
FKyn	0	Serotonin	0
Formate	0	Thbpt	1
H	1	tRNA	1e-05
H2O	1	Trp	0
H2O2	0	Trp_tRNA	0
IndolAc	0	Trypta	0
IndolP	0	Xanth	0
Ketoglutarate	1	Trp_ex	0.005

## Appendix D

# File relativi alla costruzione dei modelli FBA utilizzati

In questa Appendice sono riportati i modelli FBA costruiti in Excel. Per questioni di dimensioni, sono stati omessi i contenuti delle colonne 'Notes' e 'References' nei file inerenti alle Reazioni e i contenuti delle colonne 'InChI string' e 'Smiles' nei file dei metaboliti. I contenuti di tali colonne non sono comunque essenziali per la costruzione dei modelli.

I file sono riportati nel seguente ordine:

1. Modello FBA completo:
2. reazioni interne;
3. reazioni di scambio con l'esterno (parte A);
4. reazioni di scambio con l'esterno (parte B);
5. metaboliti.
6. Modello FBA ridotto:
7. reazioni interne;
8. reazioni di scambio con l'esterno;
9. metaboliti.





Modello FBA completo - 'reactions' (scambio con l'esterno - Parte B)

Abbreviation	Name	Reaction	GPR	Genes	Protein	Subsystem	Reversible	Lower bound	Upper bound	Objective	Confidence Score	EC Number	Notes	References
EX_for(c)	EX_for(c)	[c] : for <=>				Tryptophan metabolism	1	0	100	0				
EX_glu-L(c)	EX_glu-L(c)	[c] : glu-L <=>				Tryptophan metabolism	1	0	100	0				
EX_glucoa(m)	EX_glucoa(m)	[m] : glucoa <=>				Tryptophan metabolism	1	0	0	0				
EX_h(c)	EX_h(c)	[c] : h <=>				Tryptophan metabolism	1	-100	100	0				
EX_h(m)	EX_h(m)	[m] : h <=>				Tryptophan metabolism	1	-100	100	0				
EX_h(x)	EX_h(x)	[x] : h <=>				Tryptophan metabolism	1	-100	100	0				
EX_h2o(c)	EX_h2o(c)	[c] : h2o <=>				Tryptophan metabolism	1	-100	100	0				
EX_h2o(m)	EX_h2o(m)	[m] : h2o <=>				Tryptophan metabolism	1	-100	100	0				
EX_h2o(x)	EX_h2o(x)	[x] : h2o <=>				Tryptophan metabolism	1	-100	100	0				
EX_h2o2(c)	EX_h2o2(c)	[c] : h2o2 <=>				Tryptophan metabolism	1	0	100	0				
EX_hLkynr(c)	EX_hLkynr(c)	[c] : hLkynr <=>				Tryptophan metabolism	1	0	0	0				
EX_id3acald(c)	EX_id3acald(c)	[c] : id3acald <=>				Tryptophan metabolism	1	0	0	0				
EX_id3acald(m)	EX_id3acald(m)	[m] : id3acald <=>				Tryptophan metabolism	1	-100	0	0				
EX_ind3ac(c)	EX_ind3ac(c)	[c] : ind3ac <=>				Tryptophan metabolism	1	0	100	0				
EX_ind3ac(m)	EX_ind3ac(m)	[m] : ind3ac <=>				Tryptophan metabolism	1	0	100	0				
EX_kynate(c)	EX_kynate(c)	[c] : kynate <=>				Tryptophan metabolism	1	0	100	0				
EX_melan(c)	EX_melan(c)	[c] : melan <=>				Tryptophan metabolism	1	0	0	0				
EX_nad(c)	EX_nad(c)	[c] : nad <=>				Tryptophan metabolism	1	-100	0	0				
EX_nad(m)	EX_nad(m)	[m] : nad <=>				Tryptophan metabolism	1	-100	100	0				
EX_nad(x)	EX_nad(x)	[x] : nad <=>				Tryptophan metabolism	1	-100	100	0				
EX_nadh(c)	EX_nadh(c)	[c] : nadh <=>				Tryptophan metabolism	1	0	100	0				
EX_nadh(m)	EX_nadh(m)	[m] : nadh <=>				Tryptophan metabolism	1	-100	100	0				
EX_nadh(x)	EX_nadh(x)	[x] : nadh <=>				Tryptophan metabolism	1	-100	100	0				
EX_nadp(c)	EX_nadp(c)	[c] : nadp <=>				Tryptophan metabolism	1	0	100	0				
EX_nadph(c)	EX_nadph(c)	[c] : nadph <=>				Tryptophan metabolism	1	-100	0	0				
EX_niformanth(c)	EX_niformanth(c)	[c] : niformanth <=>				Tryptophan metabolism	1	0	100	0				
EX_nh4(c)	EX_nh4(c)	[c] : nh4 <=>				Tryptophan metabolism	1	0	100	0				
EX_ninthstrn(c)	EX_ninthstrn(c)	[c] : ninthstrn <=>				Tryptophan metabolism	1	0	100	0				
EX_o2(c)	EX_o2(c)	[c] : o2 <=>				Tryptophan metabolism	1	-100	0	0				
EX_quin(c)	EX_quin(c)	[c] : quin <=>				Tryptophan metabolism	1	0	100	0				
EX_srn(c)	EX_srn(c)	[c] : srn <=>				Tryptophan metabolism	1	0	0	0				
EX_thbpt(c)	EX_thbpt(c)	[c] : thbpt <=>				Tryptophan metabolism	1	-100	0	0				
EX_thbpt4acam(c)	EX_thbpt4acam(c)	[c] : thbpt4acam <=>				Tryptophan metabolism	1	0	100	0				
EX_trp-L(c)	EX_trp-L(c)	[c] : trp-L <=>				Tryptophan metabolism	1	-100	0	0				
EX_trypta(c)	EX_trypta(c)	[c] : trypta <=>				Tryptophan metabolism	1	0	0	0				

Modello FBA completo - 'metabolites'

Abbrivation	Name	Formula (neutral)	Formula (charged)	charge	Compartment	ChEBI ID	PubChem ID	ChEMBL ID	InChI string	Smiles
Zoxadip	2-Oxoadipate	C6H6O5	-2	Cytosol, Mitochondria	C00032		57499			
3nithim	3-Hydroxyanthranilate	C7H7NO3	0	Cytosol	C00632		15793			
3hcca	6S-3-Hydroxybutanoil-CoA	C25H38N7O18P3S	-4	Peroxisome, Mitochondria	C01144		57316			
3hkyam	3-Hydroxykynurenamine	C9H13NO2	1	Cytosol	C05636	797				
4243HP24DB	4-(2-Amino-3-hydroxyphenyl)-2,4-dioxobutanoate	C10H8NO5	-1	Cytosol	C05645	7956				
46thoxuin	4,6-Dihydroxyquinoline	C8H7NO2	0	Cytosol	C05639		28799			
48dthoxuin	4,8-Dihydroxyquinoline	C9H7NO2	0	Cytosol	C05637		28883			
4aphob	4-(2-Aminophenyl)-2,4-dioxobutanoate	C10H8NO4	-1	Cytosol	C01252		58147			
5xovindact	5-Hydroxyindoleacetalddehyde	C10H8NO2	0	Cytosol, Mitochondria	C05634		50157			
5toxindia	5-Hydroxyindoleacetate	C10H8NO3	-1	Cytosol, Mitochondria	C05635		27623			
5toxirfyn	5-Hydroxy-L-formylkynurenine	C11H12NO5	0	Cytosol	C05648		2065			
5ttrp	5-Hydroxy-L-tryptophan	C11H12NO3	0	Extra-organism, Cytosol	C00643		17780			
5hkyin	5-Hydroxykynurenine	C10H12NO4	0	Cytosol	C05651		2076			
5hkyam	5-Hydroxykynurenamine	C9H13NO2	1	Cytosol	C05638					
5moxact	5-Methoxyindoleacetate	C11H10NO3	-1	Cytosol	C05660	7971				
6xomelth	6-Hydroxymelatonin	C13H16NO3	0	Cytosol	C05643	7954				
lfrkynr	L-Formylkynurenine	C11H12NO4	0	Cytosol	C02700		30249			
lkyrr	L-Kynurenine	C10H12NO3	0	Cytosol	C00328		16946			
lkysertrh	N-Acetylserylalohin	C12H14NO2	0	Cytosol	C00978		17697			
laxcoa	Acetoacetyl-CoA	C25H36N7O18P3S	-4	Peroxisome, Cytosol, Mitochondria	C00032	C00332	57286			
laxcoa	Acetyl-CoA	C23H34N7O17P3S	-4	Peroxisome, Golgi Apparatus, Cytosol, Endoplasmic Reticulum, Nucleus, Mitochondria	C00024	C00024	57288			
laxys	5-Adenosyl-L-homocysteine	C14H20NO5S	0	Cytosol, Endoplasmic Reticulum, Nucleus, Mitochondria	C00021	C00021	16680			
lalg	2-Oxoglutarate	C5H4O5	-2	Peroxisome, Extra-organism, Cytosol, Endoplasmic Reticulum, Mitochondria	C00026	C00026	16810			
lale-L	L-Alanine	C3H7NO2	0	Peroxisome, Extra-organism, Cytosol, Mitochondria, Lysosome	C00041		16977			
lansia	2-Aminomucronate 6-semialdehyde	C6H7NO3	0	Cytosol	C03824		57495			
lannuco	2-Aminomucronate	C5H7N2O6S	1	Cytosol, Endoplasmic Reticulum, Nucleus, Mitochondria	C00019	C00019	15414			
lanth	Anthranilate	C6H6NO4	-1	Cytosol	C02220		57937			
laxcoa	Crotonoyl-CoA	C25H36N7O17P3S	-4	Peroxisome, Mitochondria	C00877		57332			
laxcoa	2-Amino-3-carboxymucronate semialdehyde	C7H8NO5	-1	Cytosol	C04409		994			
laxcoa	Coenzyme A	C27H42N7O16P5S	0	Peroxisome, Extra-organism, Golgi Apparatus, Cytosol, Endoplasmic Reticulum, Nucleus, Mitochondria	C00011	C00011	16526			
laxcoa	Formyl-5-hydroxykynurenamine	C10H13NO3	1	Cytosol	C00010	C00010	57287			
laxcoa	Flavin adenine dinucleotide oxidized	C27H31N9O15P2	-2	Peroxisome, Cytosol, Endoplasmic Reticulum, Mitochondria	C00547		7958			
laxcoa	Flavin adenine dinucleotide reduced	C27H33N9O15P2	-2	Peroxisome, Cytosol, Endoplasmic Reticulum, Mitochondria	C00016	C00016	16238			
laxcoa	Formyl-N-acetyl-5-methoxykynurenamine	C13H16NO4	0	Cytosol	C01352	C01352	58307			
laxcoa	Formate	C1H1O2	-1	Extra-organism, Cytosol, Endoplasmic Reticulum, Mitochondria	C00542					
laxcoa	L-Glutamate	C5H8NO4	-1	Extra-organism, Cytosol, Endoplasmic Reticulum, Nucleus, Mitochondria	C00058	C00058	15740			
laxcoa	Glutaryl-CoA	C26H37N7O19P3S	-5	Mitochondria	C00025		29985			
laxcoa	H+	H	1	Peroxisome, Extra-organism, Cytosol, Endoplasmic Reticulum, Lysosome	C00527		57378			
laxcoa	H2O	H2O	0	Peroxisome, Extra-organism, Golgi Apparatus, Cytosol, Endoplasmic Reticulum, Nucleus, Mitochondria, Lysosome	C00080		3380	24656		
laxcoa	Hydrogen peroxide	H2O2	0	Peroxisome, Extra-organism, Golgi Apparatus, Cytosol, Endoplasmic Reticulum, Nucleus, Mitochondria, Lysosome	C00001	C00001	3303			
laxcoa	3-Hydroxy-L-kynurenine	C10H12NO4	0	Cytosol	C03227		17380			
laxcoa	Indole-3-acetalddehyde	C10H8NO	0	Cytosol	C00637		18086			
laxcoa	Indole-3-acetate	C10H8NO2	-1	Cytosol, Mitochondria	C00954		30854			
laxcoa	Indole-3-carboxylic acid	C9H6NO3	-1	Cytosol	C01717		58454			
laxcoa	Melatonin	C13H16NO2	0	Cytosol	C01598		16796			
laxcoa	Nicotinamide adenine dinucleotide	C21H26N7O14P2	-1	Peroxisome, Extra-organism, Cytosol, Endoplasmic Reticulum, Nucleus, Mitochondria	C00003		3305	57540		
laxcoa	Nicotinamide adenine dinucleotide - reduced	C21H27N7O14P2	-2	Peroxisome, Cytosol, Endoplasmic Reticulum, Mitochondria	C00004		3306	57945		
laxcoa	Nicotinamide adenine dinucleotide phosphate	C21H25N7O17P3	-3	Peroxisome, Cytosol, Endoplasmic Reticulum, Nucleus, Mitochondria	C00006		58449			
laxcoa	Nicotinamide adenine dinucleotide phosphate - reduced	C21H26N7O17P3	-4	Peroxisome, Cytosol, Endoplasmic Reticulum, Nucleus, Mitochondria	C00005		57783			
laxcoa	N-formylanthranilate	C8H8NO3	-1	Cytosol	C05653		18410			
laxcoa	Ammonium	NH4	1	Peroxisome, Extra-organism, Cytosol, Endoplasmic Reticulum, Nucleus, Mitochondria	C01342		28938			
laxcoa	N-Methylserobin	C11H15NO2	1	Cytosol	C06212		48294			
laxcoa	O2	O2	0	Peroxisome, Extra-organism, Cytosol, Endoplasmic Reticulum, Nucleus, Mitochondria	C00007	C00007	15379			
laxcoa	Quinolinate	C7H5NO4	-2	Cytosol	C03722		29959			
laxcoa	Serobin	C10H13NO2	1	Extra-organism, Cytosol	C00780		30546			
laxcoa	Tetrahydrobiopterin	C9H15NO3	0	Cytosol, Nucleus	C00072		15372			
laxcoa	Tetrahydropterin-4a-carbinolamine	C9H15NO4	0	Cytosol	C00072		15374			
laxcoa	L-Tryptophan	C11H12NO2	0	Extra-organism, Cytosol	C00078		16828			
laxcoa	Tryptamine	C10H13N2	1	Cytosol	C00098		57887			

Modello FBA ridotto - 'reactions' (interne)

Abbreviation	Name	Reaction	Genes	Protein	Subsystem	Reversible	Lower bound	Upper bound	Objective	Confidence Score	EC Number	Notes	References
RO677	IL4I1	[c] : M, Trp + M, H2O + M, O2 -> M, IndolP + M, NH4 + M, H2O2	IL4I1	IL4I1	Tryptophan metabolism	0	0	100	0	1	1.4.3.2		
RO678_Tdo	TDO	[c] : M, Trp + M, O2 -> M, Fkyn	Tdo2.1	Tdo2.1	Tryptophan metabolism	0	0	100	0	4	1.13.11.11		
RO685	DDC_Trypta	[c] : M, Trp -> M, Trypta + M, CO2	Ddc.1	Ddc.1	Tryptophan metabolism	0	0	100	0	4	4.1.1.28		
RO987	KYNU_L_Kyn	[c] : M, L_Kyn + M, H2O -> M, AA + M, L_ALA	Kynu.1	Kynu.1	Tryptophan metabolism	0	0	100	0	4	3.7.1.3		
RO1814_Tph1	TPH1	[c] : M, Thbpt + M, Trp + M, O2 -> M, 5HTp + M, Dhhpt + M, H2O	Tph1.1	Tph1.1	Tryptophan metabolism	0	0	100	0	4	1.14.16.4		
RO1956_kat1	KAT1_L_Kyn	[c] : M, L_Kyn + M, Ketoglutarate -> M, Kyna + M, LGlu	Kat.1	Kat.1	Tryptophan metabolism	0	0	100	0	4	2.6.1.7		
RO1959	AFMID_Fkyn	[c] : M, Fkyn + M, H2O -> M, Formate + M, L_Kyn	Afmid	Afmid	Tryptophan metabolism	0	0	100	0	1	3.5.1.9		
RO1960	KMO	[c] : M, L_Kyn + M, O2 + M, NADPH + M, H -> M, 3HKyn + M, NADP + M, H2O	Kmo.1	Kmo.1	Tryptophan metabolism	0	0	100	0	4	1.14.13.9		
RO2174	INMT_Trypta	[c] : M, AMet + M, Trypta -> M, AHcys + M, MTTrypta	Inmt.1	Inmt.1	Tryptophan metabolism	0	0	100	0	1	2.1.1.49		
RO2665	HAOA	[c] : M, 3HAA + M, O2 -> M, Acms	Haoa.1	Haoa.1	Tryptophan metabolism	0	0	100	0	4	1.13.11.6		
RO2668	KYNU_3HKyn	[c] : M, 3HKyn + M, H2O -> M, 3HAA + M, L_ALA	Kynu.1	Kynu.1	Tryptophan metabolism	0	0	100	0	4	3.7.1.3		
RO2701	DDC_5HTp	[c] : M, 5HTp -> M, Serotonin + M, CO2	Ddc.1	Ddc.1	Tryptophan metabolism	0	0	100	0	4	4.1.1.28		
RO2702	IDO_5HTp	[c] : M, 5HTp + M, O2 -> M, 5HFkyn	Ido.1	Ido.1	Tryptophan metabolism	0	0	100	0	1	1.13.11.52		
RO3664	WARS	[c] : M, ATP + M, Trp + M, tRNA -> M, AMP + M, PPI + M, Trp, tRNA	Wars	Wars	Tryptophan metabolism	0	0	100	0	1	6.1.1.2		
RO4911	AFMID_5HFkyn	[c] : M, 5HFkyn + M, H2O -> M, SHKyn + M, Formate	Afmid	Afmid	Tryptophan metabolism	0	0	100	0	1	3.5.1.9		
quin_form	Quinolic Acid formation	[c] : M, Acms -> M, Quin			Tryptophan metabolism	0	0	100	0	1			
RO2173_Maoa	MAOA_Trypta	[c] : M, Trypta + M, H2O + M, O2 -> M, IndolAc + M, NH4 + M, H2O2	Maoa.1	Maoa.1	Tryptophan metabolism	0	0	100	0	4	1.4.3.4		
RO2670	Cinnavalinate	[c] : 2 M, 3HAA + 4 M, O2 -> M, Cin + 2 M, O2s + 2 M, H2O2 + 2 M, H	Cat	Cat	Tryptophan metabolism	0	0	100	0	1	1.11.1.6		
TRPtrans_Sic7a5	Tryptophan_Transport_Sic7a5	M, Trp, ex[e] <=> M, Trp[c]	Sic7a5	Sic7a5	Tryptophan metabolism	1	-100	100	0	1			
RO678_Indo	IDO	[c] : M, Trp + M, O2 -> M, Fkyn	Ido.1	Ido.1	Tryptophan metabolism	0	0	100	0	1	1.13.11.52		
RO3348	QPRT	[c] : M, H + M, PRPP + M, Quin -> M, CO2 + M, NaMin + M, PPI	Qprt	Qprt	Tryptophan metabolism	0	0	100	0	1	2.4.2.19		
RO3936	KYNU_Fkyn	[c] : M, Fkyn + M, H2O -> M, FAA + M, L_ALA	Kynu.1	Kynu.1	Tryptophan metabolism	0	0	100	0	1	3.7.1.3		
RO3988	AFMID_FAA	[c] : M, FAA + M, H2O -> M, Formate + M, AA	Afmid	Afmid	Tryptophan metabolism	0	0	100	0	1	3.5.1.9		
RO3005	NMNAT1	[c] : M, ATP + M, H + M, NaMin -> M, NAAD + M, PPI	Nmnat.1	Nmnat.1	Tryptophan metabolism	0	0	100	0	1	2.7.7.1		
RO4323	ACMSD	[c] : M, Acms -> M, Am6sa + M, CO2	Acmsd.1	Acmsd.1	Tryptophan metabolism	0	0	100	0	4	4.1.1.45		
RO2908_Maoa	MAOA_Serotonin	[c] : M, Serotonin + M, H2O + M, O2 -> M, 5HAc + M, H2O2 + M, NH4	Maoa.1	Maoa.1	Tryptophan metabolism	0	0	100	0	4	1.4.3.4		
RO2911	AANAT_Serotonin	[c] : M, AcetylCoA + M, Serotonin -> M, NAcSet + M, CoA + M, H	Aanat.1	Aanat.1	Tryptophan metabolism	0	0	100	1	4	2.3.1.87		
RO4171_kat1	KAT1_3HKyn	[c] : M, 3HKyn + M, Ketoglutarate -> M, Xanth + M, LGlu	Kat.1	Kat.1	Tryptophan metabolism	0	0	100	0	1	2.6.1.7		
RO2909	IDO_Serotonin	[c] : M, Serotonin + M, O2 -> M, 5HFkyn	Ido.1	Ido.1	Tryptophan metabolism	0	0	100	0	1	1.13.11.52		
RO2174_metTrypta	INMT_MTTrypta	[c] : M, AMet + M, MTTrypta -> M, AHcys + M, MMTrypta	Inmt.1	Inmt.1	Tryptophan metabolism	0	0	100	0	1	2.1.1.49		
RO2910	INMT_Serotonin	[c] : M, AMet + M, Serotonin -> M, AHcys + M, NM5er	Inmt.1	Inmt.1	Tryptophan metabolism	0	0	100	0	1	2.1.1.49		
RO1956_kat2	KAT2_3HKyn	[c] : M, 3HKyn + M, Ketoglutarate -> M, Xanth + M, LGlu	Kat.2	Kat.2	Tryptophan metabolism	0	0	100	0	1			
RO1956_kat3	KAT3_3HKyn	[c] : M, 3HKyn + M, Ketoglutarate -> M, Xanth + M, LGlu	Kat.3	Kat.3	Tryptophan metabolism	0	0	100	0	1			
RO4171_kat2	KAT2_L_Kyn	[c] : M, L_Kyn + M, Ketoglutarate -> M, Kyna + M, LGlu	Kat.2	Kat.2	Tryptophan metabolism	0	0	100	0	1	2.6.1.7		
RO4171_kat3	KAT3_L_Kyn	[c] : M, L_Kyn + M, Ketoglutarate -> M, Kyna + M, LGlu	Kat.3	Kat.3	Tryptophan metabolism	0	0	100	0	1	2.6.1.7		
RO2908_MaoB	MAOB_Serotonin	[c] : M, Serotonin + M, H2O + M, O2 -> M, 5HAc + M, H2O2 + M, NH4	MaoB.1	MaoB.1	Tryptophan metabolism	0	0	100	0	4	1.4.3.4		
RO2173_MaoB	MAOB_Trypta	[c] : M, Trypta + M, H2O + M, O2 -> M, IndolAc + M, NH4 + M, H2O2	MaoB.1	MaoB.1	Tryptophan metabolism	0	0	100	0	1	4.3.4		
AANAT_Trypta	AANAT_Trypta	[c] : M, AcetylCoA + M, Trypta -> M, Nacrypta + M, CoA + M, H	Aanat.1	Aanat.1	Tryptophan metabolism	0	0	100	0	1	2.3.1.87		
RO1814_Tph2	TPH2	[c] : M, Thbpt + M, Trp + M, O2 -> M, 5HTp + M, Dhhpt + M, H2O	Tph2	Tph2	Tryptophan metabolism	0	0	100	0	1	1.14.16.4		
TRPtrans_Sic7a8	Tryptophan_Transport_Sic7a8	M, Trp, ex[e] <=> M, Trp[c]	Sic7a8	Sic7a8	Tryptophan metabolism	1	-100	100	0				
Lkynr_trans_Sic7a5	Lkynr_Transporter_Sic7a5	M, LKyn, ex[e] <=> M, L_Kyn[c]	Sic7a5	Sic7a5	Tryptophan metabolism	1	-100	100	0				
Lkynr_trans_Sic7a8	Lkynr_Transporter_Sic7a8	M, LKyn, ex[e] <=> M, L_Kyn[c]	Sic7a8	Sic7a8	Tryptophan metabolism	1	-100	100	0				

Modello FBA ridotto - 'reactions' (scambio con l'esterno)

Abbreviation	Name	Reaction	GPR	Genes	Protein	Subsystem	Reversible	Lower-bound	Upper bound	Objective	Confidence Score	EC. Number	Notes	References
EX_M_AHCys(c)	EX_M_AHCys(c)	[c] : M_AHCys <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_Ketoglutarate(c)	EX_M_Ketoglutarate(c)	[c] : M_Ketoglutarate <=>>				Tryptophan metabolism	1	-100	0	0				
EX_M_L_ALA(c)	EX_M_L_ALA(c)	[c] : M_L_ALA <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_AMet(c)	EX_M_AMet(c)	[c] : M_AMet <=>>				Tryptophan metabolism	1	-100	0	0				
EX_M_AA(c)	EX_M_AA(c)	[c] : M_AA <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_Dhbpt(c)	EX_M_Dhbpt(c)	[c] : M_Dhbpt <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_Formate(c)	EX_M_Formate(c)	[c] : M_Formate <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_LGlu(c)	EX_M_LGlu(c)	[c] : M_LGlu <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_IndolAc(c)	EX_M_IndolAc(c)	[c] : M_IndolAc <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_IndolP(c)	EX_M_IndolP(c)	[c] : M_IndolP <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_Thbpt(c)	EX_M_Thbpt(c)	[c] : M_Thbpt <=>>				Tryptophan metabolism	1	-100	0	0				
EX_M_tRNA(c)	EX_M_tRNA(c)	[c] : M_tRNA <=>>				Tryptophan metabolism	1	-100	0	0				
EX_M_Trp_tRNA(c)	EX_M_Trp_tRNA(c)	[c] : M_Trp_tRNA <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_AMP(c)	EX_M_AMP(c)	[c] : M_AMP <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_ATP(c)	EX_M_ATP(c)	[c] : M_ATP <=>>				Tryptophan metabolism	1	-100	0	0				
EX_M_CO2(c)	EX_M_CO2(c)	[c] : M_CO2 <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_H2O2(c)	EX_M_H2O2(c)	[c] : M_H2O2 <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_H2O(c)	EX_M_H2O(c)	[c] : M_H2O <=>>				Tryptophan metabolism	1	-100	100	0				
EX_M_H(c)	EX_M_H(c)	[c] : M_H <=>>				Tryptophan metabolism	1	-100	100	0				
EX_M_NADP(c)	EX_M_NADP(c)	[c] : M_NADP <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_NADPH(c)	EX_M_NADPH(c)	[c] : M_NADPH <=>>				Tryptophan metabolism	1	-100	0	0				
EX_M_NH4(c)	EX_M_NH4(c)	[c] : M_NH4 <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_O2(c)	EX_M_O2(c)	[c] : M_O2 <=>>				Tryptophan metabolism	1	-100	0	0				
EX_M_O2s(c)	EX_M_O2s(c)	[c] : M_O2s <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_PPI(c)	EX_M_PPI(c)	[c] : M_PPI <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_Cin(c)	EX_M_Cin(c)	[c] : M_Cin <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_Kyna(c)	EX_M_Kyna(c)	[c] : M_Kyna <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_PPPP(c)	EX_M_PPPP(c)	[c] : M_PPPP <=>>				Tryptophan metabolism	1	-100	0	0				
EX_M_NAAD(c)	EX_M_NAAD(c)	[c] : M_NAAD <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_Am6sa(c)	EX_M_Am6sa(c)	[c] : M_Am6sa <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_5HAc(c)	EX_M_5HAc(c)	[c] : M_5HAc <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_NAcSet(c)	EX_M_NAcSet(c)	[c] : M_NAcSet <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_AcetylCoA(c)	EX_M_AcetylCoA(c)	[c] : M_AcetylCoA <=>>				Tryptophan metabolism	1	-100	0	0				
EX_M_CoA(c)	EX_M_CoA(c)	[c] : M_CoA <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_Xanth(c)	EX_M_Xanth(c)	[c] : M_Xanth <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_F5HKyn(c)	EX_M_F5HKyn(c)	[c] : M_F5HKyn <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_MMTryp(c)	EX_M_MMTryp(c)	[c] : M_MMTryp <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_NMSer(c)	EX_M_NMSer(c)	[c] : M_NMSer <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_Nactrypt(c)	EX_M_Nactrypt(c)	[c] : M_Nactrypt <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_5HKyn(c)	EX_M_5HKyn(c)	[c] : M_5HKyn <=>>				Tryptophan metabolism	1	0	100	0				
EX_M_Trp_ex(c)	EX_M_Trp_ex(c)	[e] : M_Trp_ex <=>>				Tryptophan metabolism	1	-100	0	0				
EX_LKyn_ex(c)	EX_LKyn_ex(c)	[e] : M_LKyn_ex <=>>				Tryptophan metabolism	1	-100	0	0				

Modello FBA ridotto - 'metabolites'

abbreviation	name	Formula (neutral)	Formula (charged)	charge	compartment	kegg_id	PubChem	CHEBI ID	InChI string	Smiles
M_3HAA	3-hydroxyanthranilate		C7H7NO3	0	Cytosol	C00632		36559		
M_5HFkyn	5-hydroxy-N-formylkynurenine		C11H12N2O5	0	Cytosol	C05648		2065		
M_5HTip	5-hydroxy-L-tryptophan		C11H12N2O3	0	Cytosol	C01017		17780		
M_5HKyn	5-hydroxykynurenine		C10H12NO4	0	Cytosol	C00643		2076		
M_Fkyn	N-formyl-L-kynurenine		C11H12N2O4	0	Cytosol	C02700		30249		
M_Lkyn	L-kynurenine		C10H12NO3	0	Cytosol	C00328		16946		
M_AHCys	S-Adenosyl-L-homocysteine		C14H20N6O5S	0	Cytosol		C00021	16680		
M_Ketoglutarate	2-Oxoglutarate		C5H4O5	-2	Cytosol		C00026	16810		
M_L_ALA	M_ala_DASH_L_c		C3H7NO2	0	Cytosol			16977		
M_AMet	S-Adenosyl-L-methionine		C15H23N6O5S	1	Cytosol		C00019	15414		
M_AA	anthranilate		C7H6NO2	-1	Cytosol	C04409		16567		
M_Dihbpt	6,7-dihydrobiopterin		C9H13N5O3	0	Cytosol			994		
M_Formate	Formate		CH1O2	-1	Cytosol	C00058	C00058	15740		
M_LGlu	L-Glutamate		C5H8NO4	-1	Cytosol	C00025		29985		
M_3HKyn	3-Hydroxy-L-kynurenine		C10H12N2O4	0	Cytosol	C05651		2076		
M_IndolAc	Indole-3-acetaldehyde		C10H9NO	0	Cytosol	C00637		18086		
M_IndolP	Indole-3-pyruvate		C11H9NO3	0	Cytosol	C00331				
M_MTTrypta	methyltryptamine		C11H14N2	0	Cytosol					
M_Quin	Quinolinate		C7H3NO4	-2	Cytosol	C03722		29959		
M_Serotonin	Serotonin		C10H13N2O	1	Cytosol	C00780		350546		
M_Thbpt	5,6,7,8-tetrahydrobiopterin		C9H15N5O3	0	Cytosol	C00272		15372		
M_TRNA	tryptophan-tRNA ligase		C39H47N13O22P3R	0	Cytosol	C01652				
M_Trp	L-Tryptophan		C11H12N2O2	0	Cytosol	C00078		16828		
M_Trp_tRNA	L-Tryptophanyl-tRNA(Trp)		C26H31N7O11PR	0	Cytosol					
M_Trypta	Tryptamine		C10H13N2	1	Cytosol	C00398		57887		
M_AMP	M_amp_c		C10H12N5O7P	-2	Cytosol					
M_ATP	M_atp_c		C10H12N5O13P3	-4	Cytosol					
M_CO2	M_co2_c		CO2	0	Cytosol	C00011	C00011	16526		
M_H2O2	Hydrogen peroxide		H2O2	0	Cytosol	C00027	C00027	16240		
M_H2O	M_h2o_c		H2O	0	Cytosol	C00001	C00001	3303		
M_H	M_h_c		H	1	Cytosol	C00080	C00080	3380	24636	
M_NADP	Nicotinamide adenine dinucleotide phosphate		C21H25N7O17P3	-3	Cytosol	C00006	C00006	58349		
M_NADPH	Nicotinamide adenine dinucleotide phosphate - reduced		C21H26N7O17P3	-4	Cytosol	C00005	C00005	57783		
M_NH4	Ammonium		H4N	1	Cytosol	C01342	C01342	28938		
M_O2	M_o2_c		O2	0	Cytosol	C00007	C00007	15379		
M_O2s	superoxide		O2	-1	Cytosol					
M_PPI	Diphosphate		HO7P2	-3	Cytosol					
M_Cin	M_Cinnavalininate_c		C14H8N2O6	-2	Cytosol					
M_Kyna	4-Hydroxy-2-quinolinecarboxylic acid		C10H6NO3	-1	Cytosol	C01717		58454		
M_NaMN	Nicotinate D-ribonucleotide		C11H12NO9P	-2	Cytosol					
M_PRP	5-Phospho-alpha-D-ribose 1-diphosphate		C5H8O14P3	-5	Cytosol					
M_FAA	N-formylanthranilate		C8H6NO3	-1	Cytosol	C05653		18410		
M_NAAD	Deamino-NAD+		C21H24N6O15P2	-2	Cytosol					
M_Am6sa	2-Aminomuconate 6-semialdehyde		C6H7NO3	0	Cytosol	C03824		57495		
M_5HAc	5-Hydroxyndoleacetaldehyde		C10H9NO2	0	Cytosol	C05634		50157		
M_NAcSet	N-acetylserotonin		C12H14N2O2	0	Cytosol	C00978		17697		
M_AcetylCoA	Acetyl-CoA		C23H34N7O17P3S	-4	Cytosol	C00024	C00024	57288		
M_CoA	Coenzyme A		C23H32N7O16P3S	-4	Cytosol	C00010	C00010	57287		
M_Xanth	M_Xanthurenate		C10H7NO4	-1	Cytosol					
M_F5HKyn	Formyl-5-hydroxykynurenamine		C10H13N2O3	1	Cytosol	C05647		7958		
M_MMTRYpta	Dimethyltryptamine		C12H16N2	1	Cytosol					
M_NMSer	N-Methylserotonin		C11H15N2O	1	Cytosol					
M_Nacrypta	N-Acetylserotonin		C12H14N2O2	0	Cytosol	C00978		17697		
M_Lkyn_ex	L-kynurenine		C10H12N2O3	0	Cytosol	C00328		16946		
M_Trp_ex	L-Tryptophan		C11H12N2O2	0	Cytosol	C00078		16828		

# Bibliography

- [1] Scott A Becker, Adam M Feist, Monica L Mo, Gregory Hannum, Bernhard Ø Palsson & Markus J Herrgard, “Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox”, Department of Bioengineering, University of California San Diego, 9500 Gilman Dr., La Jolla, California 92093-0412, USA, Published online 29 March 2007; doi:10.1038/nprot.2007.99.
- [2] Anne-Kristin Stavrum, Ines Heiland, Stefan Schuster, P. Puntervoll e Mathias Ziegler, Model of Tryptophan Metabolism, Readily Scalable using Tissue-Specific Gene Expression Data, Ottobre 2013.
- [3] Jan Schellenberger, Richard Que, Ronan M T Fleming, Ines Thiele, Jeffrey D Orth, Adam M Feist, Daniel C Zielinski, Aarash Bordbar, Nathan E Lewis, Sorena Rahmanian, Joseph Kang, Daniel R Hyduke & Bernhard Ø Palsson, “Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0”, *Nature Protocols* 6, 1290–1307 (2011) | doi:10.1038/nprot.2011.308, Published online 04 August 2011.
- [4] D. M. Richard, M. A. Dawes, C. W. Mathias, A. Acheston, N. Hill-Kapturczak e D. M. Dougherty, “L-Tryptophan: Basic Metabolic Functions, Behavioral Research and Therapeutic Indication”, *International Journal Tryptophan Research* 2009:2 45-60.
- [5] Jan Schellenberger, “Monte Carlo Simulation in Systems Biology”, University of California, San Diego, 2010.
- [6] Jennifer L. Reed and Bernhard Ø. Palsson, “Genome-Scale In Silico Models of E. coli Have Multiple Equivalent Phenotypic States: Assessment of Correlated Reaction Subsets That Comprise Network States”, doi: 10.1101/gr.2546004 *Genome Res.* 2004. 14: 1797-1805
- [7] Papin JA, Price ND, Palsson BØ, “Extreme pathway lengths and reaction participation in genome-scale metabolic networks.”, *Genome Res.* 2002 Dec;12(12):1889-900.
- [8] Laszlo David, “Metabolic network analysis - Optimization-based methods”, *Mathematics in Life Sciences Freie Universität Berlin*, 29 October, 2010.
- [9] Steinn Gudmundsson, Ines Thiele, “Computationally efficient flux variability analysis”, Gudmundsson and Thiele *BMC Bioinformatics* 2010, 11:489.
- [10] Jennifer L. Reed, Trina R. Patel, Keri H. Chen, Andrew R. Joyce, Margaret K. Applebee, Christopher D. Herring, Olivia T. Bui, Eric M. Knight, Stephen S. Fong, and Bernhard O. Palsson, “Systems approach to refining genome annotation”, Edited by David S. Eisenberg, University of California, Los Angeles, CA, and approved September 28, 2006 (received for review April 25, 2006).
- [11] Vinay Satish Kumar, Madhukar S Dasika and Costas D Maranas, “Optimization based automated curation of metabolic reconstructions”, Published: 20 June 2007 *BMC Bioinformatics* 2007, 8:212 doi:10.1186/1471-2105-8-212
- [12] K. Smallbone, E. Simeonidis, “Flux balance analysis: a geometric perspective.”, *J. Theor. Biol.* 258, 311–315 (2009).
- [13] K. Patil, I. Rocha, J. Forster & J. Nielsen, “Evolutionary programming as a platform for in silico metabolic engineering.”, *BMC Bioinformatics* 6, 308 (2005).

- [14] Isabel Rocha, Paulo Maia, Miguel Rocha, Eugénio C. Ferreira, “OptGene – a framework for in silico metabolic engineering”, Book of Abstracts of the 10th International Chemical and Biological Engineering Conference – CHEMPOR 2008
- [15] A.P. Burgard, P. Pharkya, & C.D. Maranas, “Optknock: a bilevel programming framework for identifying gene KnockOut strategies for microbial strain optimization.” *Biotechnol. Bioeng.* 84, 647–657 (2003).
- [16] Priti Pharkya, Anthony P. Burgard, Costas D. Maranas, “Exploring the Overproduction of Amino Acids Using the Bilevel Optimization Framework OptKnock”, Published online: 24 November 2003 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/bit.10857
- [17] N.E. Lewis et al., “Omic data from evolved E. coli are consistent with computed optimal growth from genome-scale models.”, *Mol. Syst. Biol.* 6, 390 (2010).
- [18] T. Wilhelm, J. Behre and S. Schuster, “Analysis of structural robustness of metabolic networks”, *Syst. Biol.*, Vol. 1, No. 1, June 2004.
- [19] Jan Schellenberger and Bernhard Ø. Palsson, “Use of Randomized Sampling for Analysis of Metabolic Networks”, *THE JOURNAL OF BIOLOGICAL CHEMISTRY* VOL. 284, NO. 9, pp. 5457–5461, February 27, 2009
- [20] Daniel Machado, Zita Soons, Kiran Raosaheb Patil, Eugénio C. Ferreira and Isabel Rocha, “Random sampling of elementary flux modes in large-scale metabolic networks.” Vol. 28 *ECCB* 2012, pages i515–i521 doi:10.1093/bioinformatics/bts401.
- [21] Desmond S Lun, Graham Rockwell, Nicholas J Guido, Michael Baym, Jonathan A Kelner, Bonnie Berger, James E Galagan and George M Church, “Large-scale identification of genetic design strategies using local search”, *Molecular Systems Biology* 5; Article number 296; doi:10.1038/msb.2009.57
- [22] D. M. Richard, M. A. Dawes, C. W. Mathias, A. Acheston, N. Hill-Kapturczak e D. M. Dougherty, “L-Tryptophan: Basic Metabolic Functions, Behavioral Reserch and Therapeutic Indication”, *International Journal Tryptophan Research* 2009:2 45-60.
- [23] Karthik Raman e Nagasuma Chandra, “Flux balance analysis of biological system: application and challenges”, *Briefings in Bioinformatics*, vol 10, no 4, 435-449.
- [24] V. Pérez-De La Cruz, P. Carrillo-Mora and Abel Santamaría, “Quinolinic Acid, an endogenous Molecule combining excitotoxicity, Oxidative stress and Other Toxic Mechanisms”, *International Journal of Tryptophan Research* 2012:5 1–8
- [25] M. Rondanelli, A. Opizzi, F. Monteferrario, N. Antonello, R. Manni, C. Klersy, “Efficacia nell’utilizzo di un’associazione di nutrienti (melatonina, magnesio e zinco) veicolata da polpa di pera nel controllo dei disturbi del sonno in soggetti anziani istituzionalizzati”, [http://www.slideshare.net/MercurioEditore\\_redazione/efficacia-dellutilizzo-di-unassociazione?from\\_search=2](http://www.slideshare.net/MercurioEditore_redazione/efficacia-dellutilizzo-di-unassociazione?from_search=2)
- [26] G. Perin, “Appunti di Biochimica e Biochimica nutrizionale”, 2008/2009
- [27] D. Marazziti, S. Baroni, M. Picchetti, A. Piccini, S. Silvestri, L. Dell’Osso, “Nuovi sviluppi dell’ipotesi serotoninergica della depressione: shunt del triptofano”, *Dipartimento di Psichiatria, Neurobiologia, Farmacologia e Biotecnologie, Università di Pisa*, 2013; 48(1): 23-34
- [28] S. Delpriori, L. Puccini, “Una introduzione alle reti metaboliche e alla loro analisi strutturale”, A.A. 2008 - 2009, [http://www.google.it/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=4&ved=0CEYQFjAD&url=http%3A%2F%2Fwww.di.unipi.it%2F~pisanti%2FDIDATTICA%2FINFO09%2FDelpriori-Puccini.pdf&ei=OTCBUr\\_OOq3y7Ab6pwE&usg=AFQjCNHeU\\_Pqaym3G1tezOxUYunMZqQK4w&sig2=49IhId6YezKQpWRgrLhOQg&bvm=bv.56146854,d.bGE](http://www.google.it/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=4&ved=0CEYQFjAD&url=http%3A%2F%2Fwww.di.unipi.it%2F~pisanti%2FDIDATTICA%2FINFO09%2FDelpriori-Puccini.pdf&ei=OTCBUr_OOq3y7Ab6pwE&usg=AFQjCNHeU_Pqaym3G1tezOxUYunMZqQK4w&sig2=49IhId6YezKQpWRgrLhOQg&bvm=bv.56146854,d.bGE)
- [29] “Metabolismo”, [http://www.google.it/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=1&ved=0CDEQFjAA&url=http%3A%2F%2Fwww1.dicci.unipi.it%2FPolacco\\_Giovanni%2Fslidespb%2Fmetabolismo.pdf&ei=s3aPUpbFCuTI4ATDvoCgCw&usg=AFQjCNFyV8k-8Qj6FHeVASAg3xqKRkYRpA&sig2=u0X5KCjxupfsmQm0p0Sptw&bvm=bv.56988011,d.bGE](http://www.google.it/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=1&ved=0CDEQFjAA&url=http%3A%2F%2Fwww1.dicci.unipi.it%2FPolacco_Giovanni%2Fslidespb%2Fmetabolismo.pdf&ei=s3aPUpbFCuTI4ATDvoCgCw&usg=AFQjCNFyV8k-8Qj6FHeVASAg3xqKRkYRpA&sig2=u0X5KCjxupfsmQm0p0Sptw&bvm=bv.56988011,d.bGE)

- [30] "Stechiometria della crescita cellulare", [http://www.google.it/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=2&ved=0CDoQFjAB&url=http%3A%2F%2Fwww.skuela.net%2Fmateriale-universita%2F1961-stechiometria-della-crescita-cellulare.pdf&ei=8naPUo3MDIjn4QS8kID4Dw&usg=AFQjCNF0GN2YKDZsFhN5zZjCGu\\_MocIuw&sig2=QauEp\\_f3vR2hhNp4q44Qkw](http://www.google.it/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=2&ved=0CDoQFjAB&url=http%3A%2F%2Fwww.skuela.net%2Fmateriale-universita%2F1961-stechiometria-della-crescita-cellulare.pdf&ei=8naPUo3MDIjn4QS8kID4Dw&usg=AFQjCNF0GN2YKDZsFhN5zZjCGu_MocIuw&sig2=QauEp_f3vR2hhNp4q44Qkw)
- [31] "Metabolism", [http://www.google.it/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=1&ved=0CDEQFjAA&url=http%3A%2F%2Fwww.ecology-unit.unisalento.it%2FLinkClick.aspx%3Ffileticket%3Dt6ihQXdXo2E%253D%26tabid%3D282%26mid%3D765%26language%3Den-US&ei=WnePUteaIoqN4ASO54DwCA&usg=AFQjCNG4gP-nrDUe1rDY2A\\_DnughSGaY1g&sig2=4-DPI3LJVn-Y1WUqOJBNIw](http://www.google.it/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=1&ved=0CDEQFjAA&url=http%3A%2F%2Fwww.ecology-unit.unisalento.it%2FLinkClick.aspx%3Ffileticket%3Dt6ihQXdXo2E%253D%26tabid%3D282%26mid%3D765%26language%3Den-US&ei=WnePUteaIoqN4ASO54DwCA&usg=AFQjCNG4gP-nrDUe1rDY2A_DnughSGaY1g&sig2=4-DPI3LJVn-Y1WUqOJBNIw)
- [32] LINDO Systems, Inc., "LINDO API 6.0 User Manual".
- [33] COPASI Development Team, "COPASI Documentation", Version 4.6 (Build 32), August 3, 2010
- [34] [http://www.agdumbria.it/ricerca\\_2.php](http://www.agdumbria.it/ricerca_2.php)
- [35] <http://www.slideshare.net/monnolo/serotonina-ed-effetti-sui-propri-recettori>
- [36] <http://www.my-personaltrainer.it/>
- [37] <http://www.aliasalute.it/551/pag2/Poca-melatonina-aumenta-rischio-di-diabete.html>
- [38] <http://www.medicinalive.com/medicina-tradizionale/diabetologia/melatonina-diabete-collegati/>
- [39] <http://www.doctor33.it/con-melatonina-bassa-aumenta-rischio-di-diabete-di-tipo-/articolo-19511.html>
- [40] <http://lpsolve.sourceforge.net/5.5/>
- [41] <http://www.genome.jp/kegg/pathway/map/map00380.html>
- [42] <http://www.massimospattini.it/AMINOACIDI.htm>
- [43] <http://it.wikipedia.org>