

Computational Systems Biology

UniShare

Davide Cozzi
@dlcgold

Indice

1	Introduzione	4
2	Introduzione alla Systems Biology	5
2.1	PCNA ubiquitylation	13
2.2	I Sistemi Complessi	15
2.3	Rappresentazione Grafica	21
2.4	Tipologie di Modelli	21
2.4.1	Modelli Basati su Interazioni	27
2.4.2	Modelli Logici	28
2.4.3	Modelli Meccanicistici	30
2.4.4	Modelli Basati su Vincoli	30
2.4.5	Confronto tra i Vari Approcci	31
3	Interaction-Based Modelling	34
3.1	Introduzione alle Reti PPI	37
3.2	La Teoria dei Grafi	39
3.3	Tipologie di Reti	51
3.3.1	Random Network	52
3.3.2	Scale-Free Networks	54
3.3.3	Hierarchical Networks	58
3.4	Software	62
4	Logic-Based Modelling	63
4.1	Introduzione alla Logica Booleana	70
4.2	Simulazioni su Modelli Logic-Based	71
4.2.1	Vari Esempi	74
4.3	Logica Fuzzy	81
4.4	Seminario Logica Fuzzy	86
4.4.1	Metodo Mandani	88
4.4.2	Metodo TSK	89
4.4.3	Modelli Fuzzy Dinamici	90

4.4.4	Modello della Morte Cellulare Programmata	90
4.4.5	Simpful	93
4.5	Note Conclusive	94
4.6	Software	96
5	Mechanism-Based Modelling	97
5.1	Reaction-Based Models	100
5.1.1	Sistema di Lotka-Volterra	104
5.1.2	Dinamica dei Modelli Reaction-Based	105
5.1.3	Dalle Reazioni alle Equazioni Differenziali	107
5.1.4	Esempio del Pathway RAS/CAMP/PKA	114
5.1.5	Parameter Sweep Analysis	122
5.2	Simulazioni Deterministiche	126
5.2.1	Metodi di Integrazione Numerica	128
5.2.2	Errori dei Risolutori Numerici	132
5.2.3	Problema della Stiffness	133
5.3	Simulazioni Stocastiche	134
5.3.1	Il modello di Schlögl	140
5.3.2	Chemiotassi Batterica	142
5.3.3	Altri Esempi	143
5.3.4	Verso l'Algoritmo di Gillespie	144
5.3.5	Chemical Master Equation	149
5.4	Deterministico VS Stocastico	155
5.5	Altri Algoritmi	155
5.5.1	τ -leaping	157
5.5.2	Equazioni Differenziali Stocastiche	158
5.5.3	Metodi Ibridi	158
5.5.4	Modelli spaziali	159
5.6	Parameter Estimation	161
5.6.1	Fitness Function	163
5.6.2	Gradient Descent	165
5.6.3	Simulated Annealing	166
5.6.4	Meta-Euristiche Population-Based	167
5.6.5	Particle Swarm Optimization	169
5.6.6	Dilation Function	174
5.6.7	surF	175
6	Rule-Based Modelling	176
6.1	Regole in BioNetGen	178
6.2	Simulazioni	180
6.3	Software	183

7 Constraint-Based Modelling	186
7.1 Flux Balance Analysis	189
7.1.1 Ricostruzione della Rete Metabolica	190
7.1.2 Simulazione della Rete	193
7.1.3 Validazione della Rete	202
7.1.4 Tuning della Rete	202
7.1.5 Esperimenti in Silico	203
7.2 Data-Driven Constraint	204
8 Robustezza dei Sistemi Biologici	206
8.1 Garantire la Robustezza	209
8.1.1 Feedback Regulation	209
8.1.2 Alternative/Fail-Safe Mechanism	210
8.1.3 Modularità	211
8.1.4 Tradeoff della Robustezza	212
8.2 Robustezza e Parametrizzazione	213
8.2.1 Teoria della Biforcazione	214
8.3 Sensitive Analysis	215
8.4 Bow-Tie Architecture	218
8.5 Cancro	219

Capitolo 1

Introduzione

Questi appunti sono presi a lezione. Per quanto sia stata fatta una revisione è altamente probabile (praticamente certo) che possano contenere errori, sia di stampa che di vero e proprio contenuto. Per eventuali proposte di correzione effettuare una pull request. Link: <https://github.com/dlcgold/Appunti>.

Capitolo 2

Introduzione alla Systems Biology

Per descrivere sistemi biologici complessi si hanno vari tipi di modelli. Kitano (il “padre” di quest’ambito), nel 2002, disse che per capire i sistemi biologici complessi bisogna integrare risultati sperimentali e metodi computazionali, ottenendo quindi la vera e propria **Systems Biology**. Tramite l’interazione di vari componenti si ottengono tali sistemi. Disse infatti:

To understand complex biological systems requires the integration of experimental and computational research — in other words a systems biology approach.

Weston, nel 2004, ha aggiunto l’importanza dello studio delle interazioni e delle regolazioni tra i vari componenti del sistema, studiando le risposte alla genetica o alle perturbazioni ambientali, al fine di capire nuove proprietà del sistema. Infatti disse:

Systems biology is the analysis of the relationships among the elements in a system in response to genetic or environmental perturbations, with the goal of understanding the system or the emergent properties of the system

Ideker (altro “padre” di quest’ambito), già nel 2001, aveva definito la System Biology come l’integrazione dei dati sperimentali con i modelli matematici che descrivono componenti e interazioni, al fine di simulare il comportamento complessivo “in silico”. Nel dettaglio, citandolo:

Systems biology studies biological systems by systematically perturbing them (biologically, genetically, or chemically); monitoring the gene, protein, and informational pathway responses; integrating these data; and ultimately, formulating mathematical models that describe the structure of the system and its response to individual perturbations

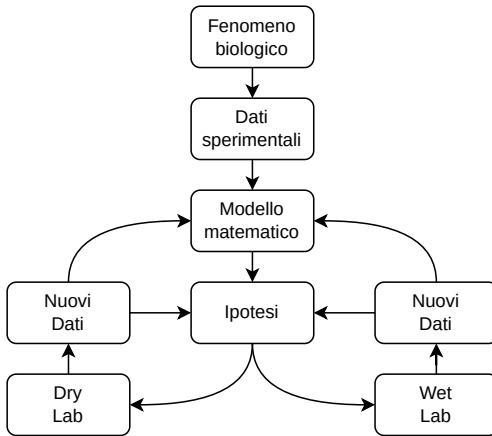


Figura 2.1: Grafico rappresentante il processo ciclico della Systems Biology.

Ai metodi standard della biologia quindi si aggiungono le teorie informatiche, quelle matematiche, quelle fisiche, quelle chimiche, quelle ingegneristiche. A partire dal fenomeno biologico quindi si effettuando esperimenti, ottenendo dei dati sperimentali relativi alle funzioni, alle strutture e alle interazioni delle varie componenti biologiche. A partire da questi dati si costruisce un **modello matematico** che porterà alla produzione di *ipotesi* a partire da esso. Inoltre l'insieme di ipotesi produrrà nuovi dati che potranno essere anche usati per rifinire il modello stesso. Inoltre tali ipotesi possono portare a sperimentazioni in **dry lab**, quindi “in silico” tramite simulazioni, ma anche in **wet lab**, quindi in laboratorio qualora possibile. Tali sperimentazioni contribuiranno a migliorare i dati stessi, producendone anche di nuovi. Si ha quindi un sistema ciclico di costante miglioramento della ricerca stessa, come visualizzabile in figura 2.1.

Un altro aspetto fondamentale del discorso è capire cosa **non** sia la *systems biology*. Citando Wolkenhauer¹:

Opening then the book, which I discovered in the London bookstore, I read the contents list: “Shotgun Fragment Assembly”, “Gene Finding”, “Local Sequence Similarities”, ... What?? ... “Protein Structure Prediction”, “Some Computational Problems Associated with Horizontal Gene Transfer” ... what on earth has this to do with systems biology, I asked myself?

...

Most important to me is however that cells and proteins are interacting in space and time, that is, we are dealing here with (nonlinear) dynamic

¹O. Wolkenhauer, Why Systems Biology is (not) called Systems Biology, BIOforum Europe 4/2007

systems. If you ask me then, systems biology is a merger of systems theory with cell biology.

...

Systems biology and bioinformatics are different but complementary.

Infatti tematiche come l'assemblaggio, l'allineamento etc... non sono tematiche della *systems biology* ma della *bioinformatica*, nonostante spesso vengano confuse e sovrapposte. L'analisi diretta dei dati biologici non è campo della *systems biology* in quanto si perde uno degli aspetti fondamentali, ovvero quello del **tempo**, che comporta lo studio di **sistemi dinamici**, che appunto di evolvono nel tempo. In bioinformatica d'altro canto si ha spesso a che fare con dati provenienti da pochi timestamp (se non direttamente da uno solo). Inoltre, sempre in bioinformatica, si studiano solitamente poche componenti biologiche, senza studiarne l'interazione tra esse.

La domanda più importante della *systems biology*, della quale possiamo vedere uno schema generale delle fasi in figura 2.2, è quindi:

dato un sistema biologico d'interesse, di cui si vogliono studiare le funzioni etc..., quale approccio modellistico è più adatto per descrivere quel sistema?

Una volta risposto a questo quesito bisogna ovviamente capire quale sia lo strumento computazionale di cui si ha bisogno per simulare e analizzare tale sistema. Bisogna infine capire quali predizioni si possono ottenere da questo modello, che comunque deve prima essere validato. Tra le cose principali che si vogliono capire abbiamo, ad esempio, se si può controllare il sistema e se si può riprodurre il tutto in laboratorio riducendo il numero di tentativi e di conseguenza anche il costo dell'esperimento in *wet lab*.

Possiamo quindi facilmente intuire che uno degli aspetti fondamentali di questo ambito è quello di fare le corrette *assunzioni*. Citando ancora Wolkenhauer²:

The modelling process itself is more important than the model. The discussion between the experimentalists and the theoretician, to decide which variables to measure and why, how to formally represent interaction in a mathematical form is the basis for successful interdisciplinary research in Systems Biology. In light of the complexity of molecular systems and the available experimental data, Systems Biology is the art of making the right assumptions in modelling.

si nota come il raggiungimento delle assunzioni stesse per ottenere il modello sia una fase di importanza maggiore rispetto al modello stesso. Il modello

²O. Wolkenhauer, Why Systems Biology is (not) called Systems Biology, BIOforum Europe 4/2007

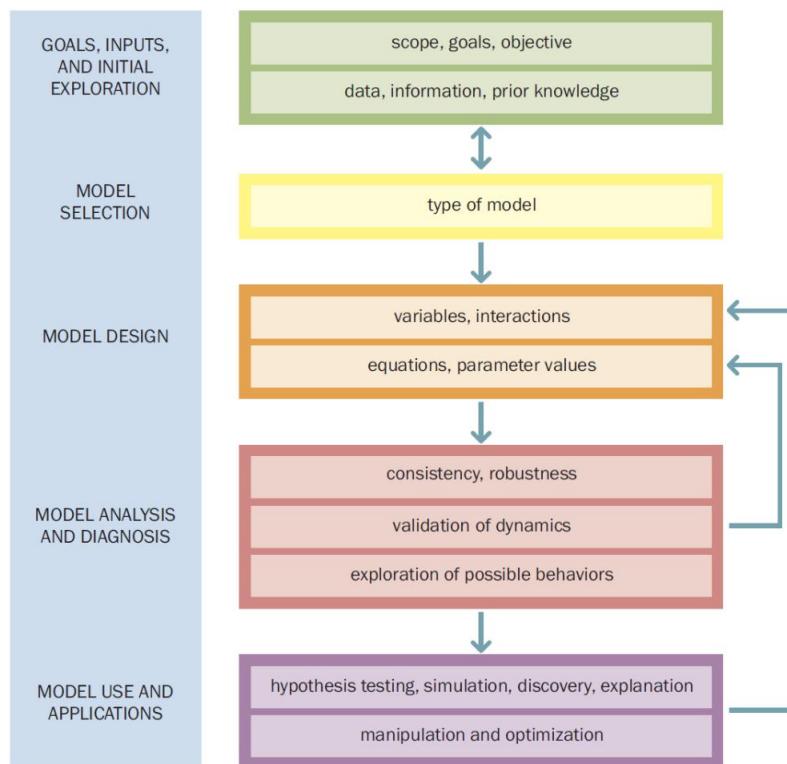


Figura 2.2: Schema generale delle fasi tipiche che compongono la systems biology.

infatti rappresenta la realtà ma non è la realtà stessa e partire da assunzioni false ed errate porterà ad un modello magari funzionante “dal punto di vista sintattico” ma non “dal punto di vista semantico”, avendo che esso non potrà mai essere validato. Nella citazione si parla inoltre di *variabili*, come elemento base dei vari modelli. Tra tali variabili si cercano relazioni, correlazioni etc... Normalmente il punto di partenza sono i *dati omici*.

Quanto qui riportato è tratto da wikipedia (<https://it.wikipedia.org/wiki/-omica>)

Definizione 1. *In biologia molecolare, ci si riferisce comunemente al neologismo omica (in inglese omics) per indicare l'ampio numero di discipline biomolecolari che presentano il suffisso “-omica”, come avviene per la genomica o la proteomica. Il suffisso correlato -oma (in inglese -omes) indica invece l'oggetto di studio di queste discipline (genoma, proteoma).*

I più importanti “-oma” proposti recentemente all’interno della comunità scientifica sono:

- il **trascrittoma** è l’insieme degli mRNA trascritti nell’intero organismo, tessuto, cellula; è studiato dalla trascrittomica
- il **metaboloma** comprende la totalità dei metaboliti presenti in un organismo; è studiato dalla metabolomica
- il **metalloma** comprende la totalità delle specie di metalli e metalloidi; è studiato dalla metallomica
- il **lipidoma** comprende la totalità dei lipidi; è studiato dalla lipidomica
- l'**interattoma** comprende la totalità delle interazioni molecolari che hanno luogo in un organismo; un nome che comunemente indica la disciplina della interattomica è quello di biologia dei sistemi (systems biology)
- lo **spliceoma** (da non confondersi con lo spliceosoma, il complesso di proteine ed acidi nucleici coinvolti nello splicing) comprende la totalità delle isoforme proteiche dovute a splicing alternativo; è studiato dalla spliceomica

- l'**ORFeoma** comprende la totalità delle sequenze di DNA che iniziano con un codone ATG e terminano con un codone di stop (sequenze note come ORF, open reading frames). Queste sequenze sono ritenute in grado di codificare per una proteina o per una parte
- **textoma**: l'insieme della letteratura scientifica disponibile alla consultazione (studiato dalla textomica)
- **kinoma**: l'insieme delle protein chinasi (dall'inglese kinase) di una cellula. Esistono pubblicazioni scientifiche che citano il termine kinomica
- **glicosiloma**: correlato alle reazioni di glicosilazione (studiato dalla glicosilomica)
- **fisioma**: correlato alla fisiologia (studiato dalla fisiomica)
- **neuroma**: l'insieme delle componenti nervose di un organismo (studiato dalla neuromica)
- **predittoma**: l'insieme delle predizioni di struttura proteica
- **reattoma**: l'insieme dei processi biologici
- **ionoma**: insieme dei nutrienti minerali e degli elementi in tracce che si trovano in un organismo
- **connettoma**: l'insieme di tutti i neuroni e le sinapsi di un cervello

Si hanno quindi vari “livelli” di studio, al variare dei dati omici, per i quali variano gli strumenti. Ad esempio:

- si ha il **genoma**, studiato tramite il *sequenziamento*, la *genotipizzazione* etc...
- il **trascrittoma**, ottenuto dopo la *trascrizione*, studiato tramite *microarrays*, *oligonucleotide chips* etc...
- il **proteoma**, ottenuto dopo la *traduzione*, studiato tramite *proteomica MS-based*, *elettroforesi* etc...
- il **metaboloma**, ottenuto tramite le *reazioni*, studiato tramite *spettroscopia di massa*, *risonanze magnetiche* etc...

- l'**interattoma**, ottenuto tramite appunto le varie *interazioni*, studiato tramite *screens yeast-to-hybrid* etc... .
- il **fenomeno**, ottenuto dopo l'*integrazione* delle varie interazioni, studiato tramite *gene inactivations* etc... .

Ognuno di questi “livelli” ha una panoramica diversa su quello che sta accadendo, è accaduto, potrebbe accadere o accadrà ad una certa cellula. Partendo dalle informazioni dinamiche/cinetiche, ovvero dai dati, e dalle informazioni strutturali dei vari *pathway* si riesce ad ottenere la rappresentazione matematica. Ovviamente è impensabile pensare di studiare tutti i “livelli” contemporaneamente ma si può studiare solo una parte del sistema, studiandone un paio di “livelli” o poco più. Inoltre ogni “livello” ha associato un suo formalismo matematico, legato alla singola modellazione matematica. Non sempre tali formalismi sono facilmente integrabili (magari in un caso ho delle EDO e in un altro dei grafi). Si ha quindi non solo un discorso di *data integration* ma anche di integrazione dei modelli matematici stessi e questo non sempre è possibile.

Nella realtà, inoltre, prima di scegliere un modello bisogna scegliere l'*approccio* con cui ottenerlo. Generalmente se ne hanno due in *systems biology*:

1. l'approccio **top-down**. In questo caso si parte dalle analisi omiche, solitamente con pochissimi timestamp, i cui risultati vengono trattati con tecniche bioinformatiche, che riducono anche l'influenza degli errori, per ottenere una **mappa globale di interazioni**, con le interazioni tra migliaia di componenti cellulari, dalla quale si ottiene il **modello predittivo del sistema**. Questo approccio è quindi supportato da una grande quantità di dati basati su *high-throughput* e *global profiling*
2. l'approccio **bottom-up**. In questo caso si parte dalle informazioni, prevalentemente di letteratura, le interazioni tra le componenti individuali del sistema, cercando magari le concentrazioni o il *kinetic-rate*, ovvero la variazione della concentrazione di un reagente o di un prodotto nel tempo misurata in moli per secondo $\left[\frac{M}{s}\right]$. Tali informazioni potrebbero non essere precise. Da queste si formalizza un modello matematico per avere poi comparazioni tra esperimenti e modelli di simulazione, ottenendo alla fine il **modello predittivo del sistema**. Questo approccio soffre quindi la mancanza di dati, specialmente di dati quantitativi. Questo approccio è più vicino a quello tipico della biologia, avvicinandosi per alcuni aspetti al *pensiero riduzionista* (che mira a studiare

piccole componenti del sistema).

Tale approccio è sicuramente più complesso, per quanto si possa limitare a studiare pathway e non l'intero metaboloma, ma per questo anche più informativo.

Ovviamente tali approcci, per quanto sarebbe fantastico, non possono essere usati in contemporanea. Detto questo solitamente l'approccio top-down studia i sistemi su larga scala per poi, a volte, procedere con uno studio bottom-up. In generale comunque la scelta dipende dalla singola situazione. Non esiste un meglio o un peggio, anche se i modelli generati dall'approccio top-down hanno generalmente una minor capacità predittiva anche se studiano sistemi più ampi rispetto all'approccio bottom-up.

Bisogna distinguere quindi quali siano le tecniche tipiche della bioinformatica (ma anche della statistica) e quali quelle della *systems biology*. L'uso di tecniche per la ricerca di similarità, correlazioni, causalità probabilistica, clustering (dove si noti che non ha un ruolo significativo il **tempo**) etc... non sono di interesse della *systems biology*, che invece è interessata allo studio delle causalità in cui il *tempo* è intrinseco e necessario. Questa necessità di avere il *tempo* comporta una maggior difficoltà nel recuperare i dati e dell'eseguire la sperimentazione ma comporta, del resto, un forte “potere di spiegazione e predizione” da parte del modello stesso.

Vediamo ora qualche definizione di base.

Definizione 2. Definiamo **modello** come una descrizione rigorosa e assolutamente non ambigua di un sistema. Nel dettaglio tale descrizione è ottenuta tramite un adeguato formalismo matematico (l'unico per definizione non ambiguo) e un adeguato livello di astrazione (importante per non avere informazioni ridondanti o inutili nel modello).

Definizione 3. Definiamo **proprietà/comportamento emergente** ogni caratteristica strutturale (quindi di topologia) o dinamica (quindi in evoluzione nel tempo) di un sistema che non può essere capita e/o spiegata banalmente tramite l'enumerazione delle componenti ma che deve essere derivata unicamente come conseguenza delle interazioni tra le componenti stesse del sistema.

Definizione 4. Definiamo **simulazione** come una tecnica “computer-based” per determinare una qualsiasi caratteristica emergente e/o predire l’evoluzione temporale del sistema.

Definizione 5. Definiamo **metodo computazionale** come una soluzione automatica, basata su uno specifico algoritmo, usata per risolvere problemi difficili (da intendersi “difficili” anche a livello computazionale) e per analizzare sistemi in diverse condizioni.

Si noti che, come evidenziato da Fawcett e Higginson³, l'uso eccessivo dei formalismi matematici rendono difficile la comunicazione con i biologi, quindi bisogna muoversi di conseguenza. I modellatori dovrebbero essere preparati a sviluppare nuovi strumenti matematici e computazionali, invece di "forzare" la descrizione e l'analisi del sistema con un framework preferito e facilmente applicabile (tipo usare le EDO per tutto a priori). I biologi sperimentali dovrebbero essere aperti a progettare nuovi protocolli di laboratorio per identificare tutte le caratteristiche qualitative e, soprattutto, quantitative che ancora mancano (per aiutare anche i modellisti). **La parte più interessante del gioco del modellismo non è ciò che il modello permette di capire, ma esattamente ciò che non è in grado di spiegare**, infatti, secondo, Box:

essentially, all models are wrong, but some are useful.

e, secondo Bower e Bolouri:

In fact, all modelers should be prepared to answer the question: “what do you know that you did not know before?” If the answer is “that i was correct”, it is best to look elsewhere.

Infatti un modello non solo deve rispondere a quello che già si sa ma deve predire qualcosa che ancora non si sa (magari anche non funzionando).

2.1 PCNA ubiquitylation

Vediamo brevemente uno studio in cui ha partecipato anche la professoressa Besozzi dove il non funzionamento del modello ha portato ad una nuova scoperta scientifica⁴.

In questo studio si cercava di studiare la **Post Replication Repair (PRR)**, ovvero il principale pathway di tolleranza al danno del DNA che bypassa le lesioni del DNA durante la *fase S*, che è in citologia (la branca della biologia che studia la cellula dal punto di vista morfologico e funzionale) una fase del ciclo cellulare, durante la quale il processo principale è la sintesi e duplicazione del materiale genetico contenuto nel DNA. Bombardando il lievito con raggi UV si è quindi studiata la proteina **PCNA**, ovvero l'*l'antigene*

³Tim W. Fawcett and Andrew D. Higginson, Heavy use of equations impedes communication among biologists, PNAS 2012

⁴Flavio Amara, Riccardo Colombo, Paolo Cazzaniga, Dario Pescini, Attila Csikász-Nagy, Marco Muzi Falconi, Daniela Besozzi, Paolo Plevani , In vivo and in silico analysis of PCNA ubiquitylation in the activation of the Post Replication Repair pathway in *S. cerevisiae*, BMC 2013

nucleare di proliferazione cellulare. La struttura di tale proteina (di forma a ciambella) è in grado di assumere una peculiare conformazione la quale le consente di contattare il DNA (DNA clamp) e di promuovere l'azione della polimerasi durante la replicazione del DNA⁵. I raggi UV provocano lesioni che vengono “trattate” dalla PCNA. Se ne è quindi studiata l'**ubiquitazione**, modificazione post-traduzionale di una proteina dovuta al legame covalente di uno o più monomeri di ubiquitina. Tale legame porta, solitamente, alla degradazione della proteina stessa⁶. La *mono-ubiquitazione* avviene tramite gli enzimi *Rad6* e *Rad8* mentre la *poli-ubiquitazione* tramite gli enzimi *Rad5* e *Ubc13-Mms2*. La prima comporta errori di trascrizione, in quanto si aveva sintesi di DNA tra le lesioni, formando *mutageni*, mentre la seconda è “error free”.

Si conoscevano quindi i principali attori del fenomeno, ovvero la proteina e gli enzimi. C'erano varie cose che però non si conoscevano:

- l'ordine spazio temporale della cascata delle interazioni delle varie proteine, non sapendo anche i tempi di attivazione dei vari enzimi
- se il numero di lesioni influenzasse il bilanciamento tra le *mono-ubiquitazioni* e le *poli-ubiquitazioni*
- se esistesse una soglia relativa al danno che regolasse l'interazione tra i due sub-pathway

Si è quindi proceduto, in *wet lab*, irradiando il lievito in modo controllato, misurando *mono-ubiquitazioni* e le *poli-ubiquitazioni* al passare del tempo (da 0 a 300 minuti) a varie dosi di UV, e contemporaneamente studiando un modello matematico (tramite le varie reazioni, rappresentate tramite *prodotti* e *reagenti*) per effettuare le simulazioni. Si è visto, in laboratorio, che le varie forme ubiquitilate di PCNA sono assenti a basse dosi di UV ($5 \frac{J}{m^2}$ e $10 \frac{J}{m^2}$), mentre ad alte dosi di UV ($50 \frac{J}{m^2}$ e $75 \frac{J}{m^2}$) entrambi i segnali sono ancora presenti dopo 5 ore nei *western blot*. La simulazione matematica confermava quanto stesse succedendo a bassi dosaggi ma non riusciva ad ottenere i risultati ad alti dosaggi. Dopo vari tentativi, rifacendo gli esperimenti (variando enzimi e geni) e sistemandone il modello (tramite *parameter sweeping/estimation, analisi di sensitività*) si è sospettato che il modello fosse in realtà “corretto” ma non completo, mancava qualche ipotesi. Da qui la scoperta: si ha anche un altro pathway, il **Nucleotide Excision Repair (NER)** che “assiste” la *PCNA* quando le cellule sono gravemente lesionate.

⁵<https://it.wikipedia.org/wiki/PCNA>

⁶<https://it.wikipedia.org/wiki/Ubiquitina>

NER è infatti attivo nella *fase S* e serve alla *PRR* per funzionare correttamente *in vivo*. Risistemando il modello con *NER* ed enzimi annessi le simulazioni hanno funzionato.

Questa è la prova che quando un modello non funziona si può ottenere anche una scoperta scientifica, ed è una delle situazioni (coi giusti limiti) più interessanti di questa branca di ricerca.

2.2 I Sistemi Complessi

In *systems biology* si ha quindi a che fare con sistemi che vengono definiti **sistemi complessi**, ovviamente presi nella loro “sottoclasse” relativa ai sistemi biologici.

Definizione 6. *Si definisce un **sistema complesso** come un sistema consistente di un certo numero di componenti più o meno semplici che, prese nel loro insieme, danno vita ad un comportamento emergente, grazie alle loro mutue interazioni.*

In questo contesto assumono importanza tre concetti chiave:

1. **comportamento non lineare**, quindi non facilmente prevedibile
2. **sistema aperto**, ovvero dove l’interazione con l’ambiente da parte del sistema è una delle caratteristiche da studiare e modellare
3. **sistema dinamico**, ovvero si ha che il sistema evolve nel tempo

Uno dei punti cruciali è inoltre capire che quando si procede alla modellazione di un certo sistema non si deve modellare anche cosa ci si aspetta da quel modello. Tale informazione infatti deve scaturire dalle simulazioni del modello stesso in modo completamente autonomo.

Come visto si studiano quindi insiemi di componenti. L’insieme complessivo delle funzionalità del sistema non è determinato però da una specifica funzione di ogni componente ma dalle loro interazioni. Si hanno quindi altri due concetti chiave:

1. **topologia/architettura interna**
2. **moduli funzionali**

Anche componenti molto semplici possono dare vita a un sistema complesso. Vediamo quindi qualche esempio di sistema complesso:

- un esempio “semplice” è quello di una **reazione enzimatica con feedback negativo**. In questo caso si hanno una serie di *reazioni lineari* che dal legame di un *substrato* portano ad un *prodotto*. La complessità viene data dal *feedback negativo* in quanto la produzione del prodotto stesso porta il substrato a non legare. Si ha quindi la cosiddetta **autoregolazione** che rende questo un vero e proprio *sistema complesso*, avendo che il comportamento emergente del sistema è in realtà difficile da prevedere.
Ovviamente si potrebbe anche assumere il caso meno semplice dove si ha una serie di *reazioni non lineare*.
Si può quindi arrivare anche a parlare di casi più “estesi”, come quello ad esempio di un **pathway metabolico**. Ci sarebbe inoltre un caso, seguendo questo filo pensiero, ancora più estremo, ovvero quello del **metabolismo di un’intera cellula**, come visualizzabile nella figura 2.3, dove si hanno moltissime parti che nel dettaglio sono lineari ma che si autoregolano a vicenda, ottenendo quindi un *sistema complesso* davvero impossibile da studiare.
- un altro esempio può essere quello di una **rete di interazioni proteina-proteina (protein-protein interaction network)** dove si hanno:
 - **nodi** che rappresentano le proteine
 - **archi** che rappresentano **interazioni fisiche** e **interazioni funzionali** tra proteine (???)

In questo caso la “complessità” è data soprattutto dal numero incredibilmente alto di proteine (quindi di nodi) e di interazioni tra esse (quindi di archi) nel nostro sistema. Un esempio è visualizzabile in figura 2.4.

- un altro esempio è quello delle **reti di regolazione genica (gene regulatory network)** dove appunto si studiano le relazioni che si hanno tra le espressioni e le regolazioni tra geni. In questo caso si hanno:
 - **nodi** che rappresentano i geni
 - **archi** che rappresentano le regolazioni tra geni

Anche in questo caso si possono avere feedback e tali reti sono utili per studiare l’*over-espressione di geni*. Inoltre deve essere chiaro che la modifica in un certo gene si ripercuote, chi più chi meno,

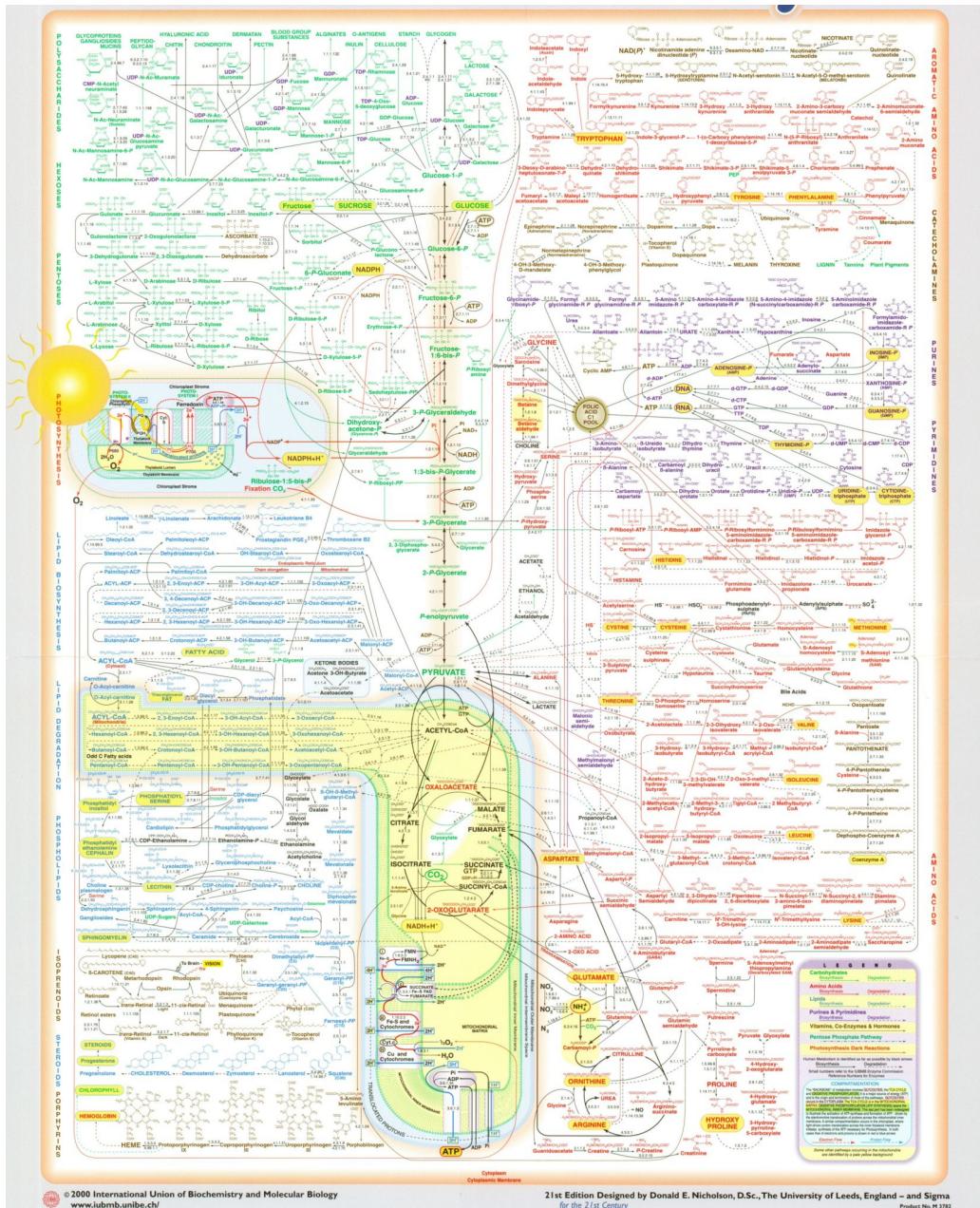


Figura 2.3: Insieme dei pathway che “compongono” il metabolismo di un’intera cellula. Tale rappresentazione è stata fatta da Donald E. Nelson, dell’università di Leeds e dall’azienda Sigma-Aldrich per la International Union of Biochemistry and Molecular Biology del 2000.

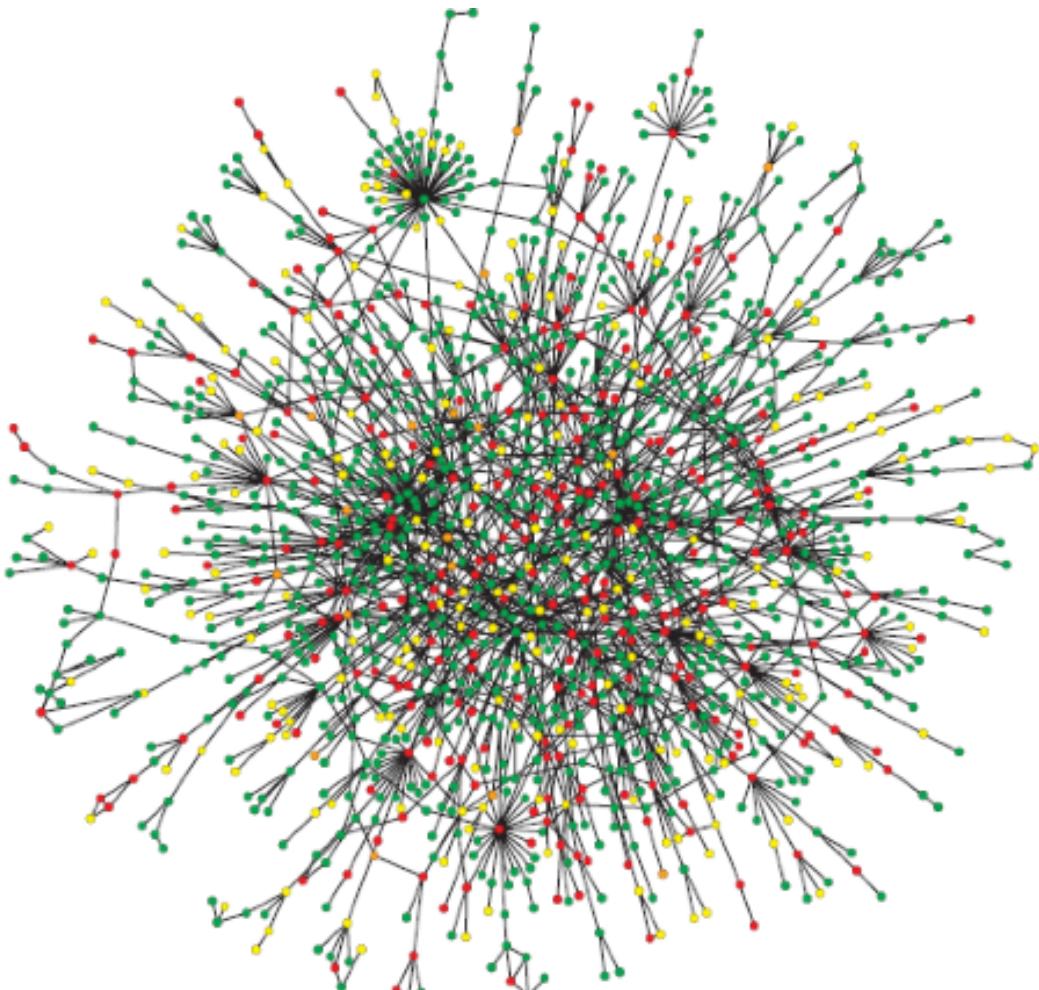


Figura 2.4: Esempio di rete di interazioni proteina-proteina, <https://www.ebi.ac.uk/training/online/courses/network-analysis-of-protein-interaction-data-an-introduction/protein-protein-interaction-networks/>. In tale rete si ha lo studio sul lievito e i vari colori dei nodi rappresentano vari effetti fenotipici legati alla rimozione della proteina rappresentata dal nodo stesso. Si ha rosso per l'effetto letale, verde per l'effetto non letale, arancione per la crescita lenta e giallo per effetto sconosciuto.

sull’intera rete anche se non si ha un singolo gene che “controlla” l’intera rete ma tutti contribuiscono alla funzionalità dell’intero sistema complesso

- aumentando ancora il livello di complessità potremmo pensare allo studio di un certo pathway, come ad esempio il *segnale di trasduzione*, in una cellula tenendo però conto anche della *componente spaziale*, tridimensionale, della stessa, nonché le interazioni con l’ambiente. La componente spaziale, che ovviamente aggiunge complessità, è una parte rilevante del modello, come il “movimento” al suo interno (prestando sempre attenzione a non aggiungere componenti inutili al modello stesso). L’interazione con l’ambiente può portare, ad esempio, a *cascate di reazioni intra-cellulari* e a vari “input”, come *ormoni, fattori di sopravvivenza, fattori di crescita/anti-crescita, fattori di morte etc...* da considerare nel modello
- un altro esempio ancor più “complesso” può essere quello della **crescita tumorale**, magari ponendo al centro dello studio anche il rapporto tra essa e la **vascolarizzazione**, ovvero la distribuzione di vasi sanguigni in un tessuto, in quanto magari si vuole studiare la vicinanza tra il tumore e i vasi sanguigni. In questo contesto non solo lo spazio tridimensionale è di fondamentale importanza ma bisogna anche modellare cellule di vario tipo (normali, cancerogene, legate al sistema immunitario, in apoptosis, necrotiche etc...), che interagiscono in modo diverso tra loro, magari avendo anche “mutazioni” da normali a cancerogene etc... Si hanno quindi componenti eterogenee, dovendo per lo più anche modellare i vasi sanguigni e le interazioni con le cellule.
- un altro esempio, “complesso” almeno quanto il precedente, è lo studio della **formazione di biofilm**, ovvero una aggregazione complessa di microrganismi contraddistinta dalla secrezione di una matrice adesiva e protettiva. Tale barriera è comunque una struttura permeabile permettendo il passaggio dei nutrienti. In un biofilm i microrganismi, tendenzialmente batteri, non solo crescono ma, soprattutto quelli più interni e “protetti”, diventano anche più resistenti. Questa è una seria complicazione per la loro eliminazione quando fuoriescono dal biofilm. Anche qui quindi bisogna modellare lo spazio tridimensionale, l’interazione con l’ambiente, l’interazione tra i vari microrganismi (anche se si ha solitamente poca eterogeneità)

- cambiando prospettiva un altro esempio di *sistema complesso* è quello dello **sviluppo embrionale e della differenziazione cellulare** dove, a partire dall'embrione e da cellule staminali si vanno a formare tutti i tipi di cellule che formeranno, ad esempio, i tessuti, gli organi etc... dell'uomo. In questo caso solitamente si ha un tipo di modellazione diverso, basato su componenti semplici
- un altro esempio è quello dello studio dell'**ecosistema**. Nel dettaglio uno degli aspetti studiati è quello della cosiddetta **dinamica preda-predatore**. Tale dinamica descrive il rapporto tra il numero di prede e di predatori e osserva un comportamento oscillatorio (se aumentano i predatori diminuiscono le prede fino a che non sono abbastanza per i predatori, che calano di numero, portando il numero di prede a crescere etc...)
- infine un ultimo esempio, molto attuale, di *sistema complesso* è quello dello **studio epidemiologico della diffusione di epidemie/pandemie** dove la “complessità” è incrementata anche dagli aspetti sociali e psicologici delle persone, nonché dalla loro eterogeneità anche nel dominio epidemiologico (infetti, gravemente infetti, guariti, esposti, suscettibili all'infezione, morti etc...)

In questi esempi si è spesso parlato più o meno esplicitamente di **livelli di complessità**. Per poter avere un'idea di quanti possano essere bisogna considerare vari punti di vista:

- un primo punto di vista è dato dalla **scala spaziale** dei fenomeni che si studiano. Possiamo studiare infatti eventi che accadono nel range dei nanometri, o meno, fino a pensare ad eventi in scala umana, in metri. Inoltre anche un evento che avviene in nanometri può avere conseguenze visibili in metri. Questo tipo di complessità è per lo più un problema matematico dal punto di vista della gestione della stessa
- un secondo punto di vista è dato dalla **scala temporale** dei fenomeni che si studiano. Anche in questo caso si passa dai nano-secondi, o meno, ai miliardi di anni. Un evento quasi istantaneo può avere conseguenze evolutive tra milioni di anni. La gestione di questo tipo di complessità è un grande problema dal punto di vista computazionale. La complessità aumenta all'aumentare della scala temporale

- altri livelli di complessità sono dati dai *livelli di funzione di un organismo*, avendo, ad esempio, che da *trascrittoma*, *proteoma* e *metaboloma*, in ottica pathway, si passa al *fisioma*, in ottica cellule, tessuti, organi e direttamente l'uomo

Pensando anche solo alla scala spaziale e quella temporale si ha inoltre che esse sono in sinergia ma è comunque pressoché impossibile pensare ad un modello che tenga traccia in modo completo o quasi di entrambe queste scale.

2.3 Rappresentazione Grafica

Ai biologi/biotecnologi etc... piace fare diagrammi e mappe concettuali per rappresentare graficamente le conoscenze biologiche che hanno su un sistema, ad esempio componenti molecolari e le loro mutue relazioni, formazione di complessi molecolari, presenza di feedback di regolazione positivi/negativi etc.... Come si intuisce facilmente diagrammi di questo tipo sono soggetti ad interpretazioni ambigue e limitano anche l'esplicita rappresentazione della conoscenza biologica. La matematica è l'unico linguaggio non ambiguo e fortunatamente esistono anche formalismi, come i *grafi*, le *reti di Petri* etc... che non solo sono formalmente rigorosi ma hanno anche un'interpretazione grafica (tanto amata dalle persone). Ovviamente non sempre si hanno queste soluzioni intermedie. La modellazione matematica risolve ogni errata interpretazione e descrive in modo non ambiguo quello che accade nel sistema e può potenzialmente includere ogni tipo di ipotesi che può poi essere studiata e testata in *wet lab*. In ogni caso i diagrammi possono avere utilità nella fase preliminare di discussione tra il biologo e il modellista: può essere un buon punto di partenza ma non sarà mai sufficiente per modellare il sistema, che si può ottenere solo con la formalizzazione matematica di componenti e interazioni. Un esempio è visualizzabile in figura 2.5⁷.

2.4 Tipologie di Modelli

Sistemi biologici differenti necessitano di approcci modellistici differenti, ovvero di framework matematici, quindi ad un preciso formalismo, e conseguentemente computazionali diversi. Inoltre bisogna sempre tenere in con-

⁷Besozzi D. (2016) Reaction-Based Models of Biochemical Networks. In: Beckmann A., Bienvenu L., Jonoska N. (eds) Pursuit of the Universal. CiE 2016. Lecture Notes in Computer Science, vol 9709. Springer, Cham. https://doi.org/10.1007/978-3-319-40189-8_3

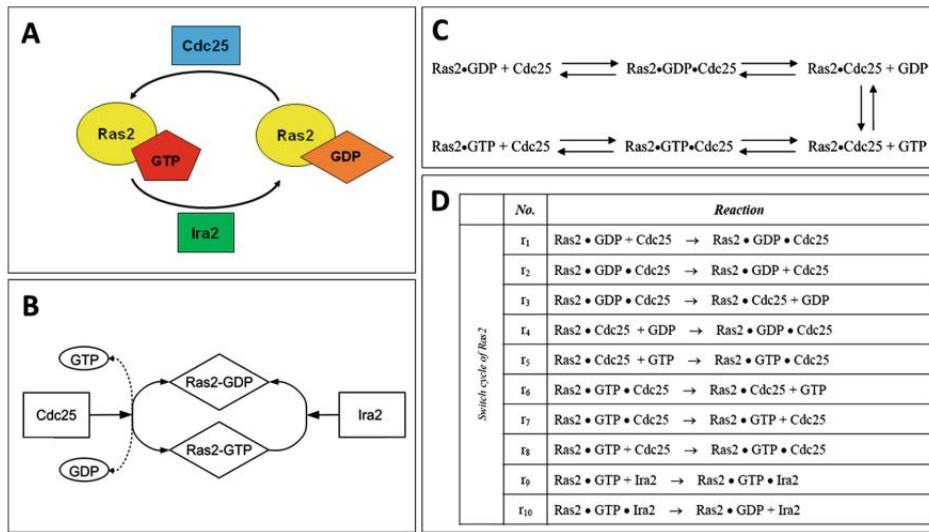


Figura 2.5: Esempio (senza entrare nei dettagli biologici che sarebbero ora superflui) di un diagramma ambiguo, in figura A, tipico dell’approccio biologico. Si hanno poi successive migliorie formali fino ad arrivare al modello matematico preciso, in figura D, e non ambiguo formato da 10 reazioni biochimiche.

siderazione che ogni metodo computazionale legato ad un preciso modello può rispondere solo a certe tipologie di domande. Non si ha però una corrispondenza biunivoca tra ogni approccio modellistico e ogni sistema biologico, infatti diversi modelli potrebbero prestarsi bene ad un certo sistema biologico (anche se alcuni modelli sono inapplicabili per certi sistemi biologici o per certe domande su tali sistemi). La scelta del modello è quindi fortemente legata alle entità che si vogliono rappresentare e alle risposte che si vogliono ottenere dal modello. **Non si ha una strategia universalmente valida per scegliere il miglior approccio modellistico in base al sistema biologico d’interesse.**

Il primo passo è quindi l’interazione tra il biologo/biotecnologo e il modellista. Il primo deve porsi varie domande tra cui:

- cosa si sa e cosa non si sa del sistema biologico in questione?
- che tipologie di dato di laboratorio sono disponibili?
- che tipologie di dato posso misurare effettivamente in laboratorio?

Anche il modellista quindi si deve porre delle domande fondamentali, tra cui:

- quale formalismo matematico si presta meglio per questo problema?

- che strumenti computazionali sono necessari?
- che tipo di predizioni mi aspetto dal modello?

Queste questioni sono “in ciclo” tra di esse e sono la base degli studi in *systems biology*, dove farsi domande è una parte fondamentale. In merito all’ultima domanda del biologo è interessante notare che un modello **dove** essere validato in laboratorio. Qualora non sia possibile, ad esempio un “caso limite” emerso dallo studio del modello, non si può fare nulla (anche se, qualora si avessero più modelli completamente distinti che portano allo stesso risultato si può presupporre che ci sia un fondo di verità).

La domanda fondamentale resta però:

qual è la questione scientifica? Perché mi serve un modello?

La risposta a questa domanda deve essere “sicura” prima di intraprendere uno studio di modellazione.

Nel dettaglio, durante il corso, si vedranno i quattro approcci modellistici tradizionali più usati anche se si tratta di una selezione tra la moltitudine degli approcci presenti:

1. **modelli basati su interazioni (*interaction-based models*)**
2. **modelli basati su vincoli (*constraint-based models*)**
3. **modelli logici (*logici-based models*)**
4. **modelli meccanicistici (*mechanism-based models*)**

In generale dato un certo sistema biologico d’interesse, dopo aver risposto alla domanda fondamentale e avendo quindi ben chiaro il fine di tale modello, la scelta del modello stesso viene presa considerando secondo quattro aspetti fondamentali:

1. la **dimensione del sistema**, data in primis dal numero di componenti e dal numero delle interazioni tra esse. Si distinguono, secondo questo aspetto, due grandi macro-categorie di sistemi:
 - (a) **sistemi small-scale**, se siamo nel range delle unità o delle decine di componenti/interazioni
 - (b) **sistemi large-scale**, se siamo nel range delle centinaia o migliaia (se non oltre) di componenti/interazioni

Questo è già un ottimo fattore discriminatorio per la scelta del sistema

2. il **livello di dettaglio** necessario a descrivere in modo completo le componenti del sistema e le loro interazioni. Si ricorda sempre però che formalizzare informazioni inutili e/o ridondanti comporta solo un'inutile spreco dal punto di vista computazionale
3. il **tipo** e la **qualità** dei **dati sperimentali** che sono già disponibili o che si è in grado di produrre all'evenienza con precisi protocolli al fine di supportare la fase di modellazione. Tali dati possono essere ad esempio *dati omici*, *western blots* etc...
4. il **carico computazionale** che l'approccio scelto comporta in fase di simulazione e analisi dei dati. Un esempio è quello della *dinamica molecolare*, che studia come interagiscono tra loro più molecole (o anche il comportamento interno di una sola). Tali studi normalmente impiegano settimane per simulare anche range temporali molto ridotti e necessitano di molte informazioni che non possono essere trascurate per ottenere un modello ed una simulazione realistici. Questa scelta è spesso un trade-off nella scelta di **approcci modellistici quantitativi** e **approcci modellistici qualitativi**.
L'uso di **super computer**, di **tecniche di calcolo parallelo su GPU** etc... sono molto comuni in *systems biology*

Una misura fondamentale è poi la **capacità predittiva del modello** che, se bassa, ci porta a preferire *modelli qualitativi*, se alta invece a *modelli quantitativi*.

Si ha quindi un comodo grafico che classifica le quattro tipologie di modelli in base a questi quattro aspetti⁸ in figura 2.6. Nel grafico notiamo come i *modelli meccanicistici*, tra quelli analizzati, siano quelli con il più alto potere predittivo, che è un aspetto fondamentale ma anche con i più alti livelli di dettaglio, costi computazionali e sfide nella misurazione dei dati richiesti. L'ovvia conseguenza è che la dimensione del sistema da studiare deve essere ridotta, avendo quindi *sistemi small-scale*.

Si noti che collocare i *modelli logici* non sia così banale in quanto il livello di dettaglio e la facilità di misurazione possono essere migliorati usando ad esempio le **logiche fuzzy**.

Un altro aspetto fondamentale da tenere in considerazione è che il processo di modellazione richiede molto tempo e si basa su continui rifinimenti del

⁸Besozzi D. (2016) Reaction-Based Models of Biochemical Networks. In: Beckmann A., Bienvenu L., Jonoska N. (eds) Pursuit of the Universal. CiE 2016. Lecture Notes in Computer Science, vol 9709. Springer, Cham. https://doi.org/10.1007/978-3-319-40189-8_3

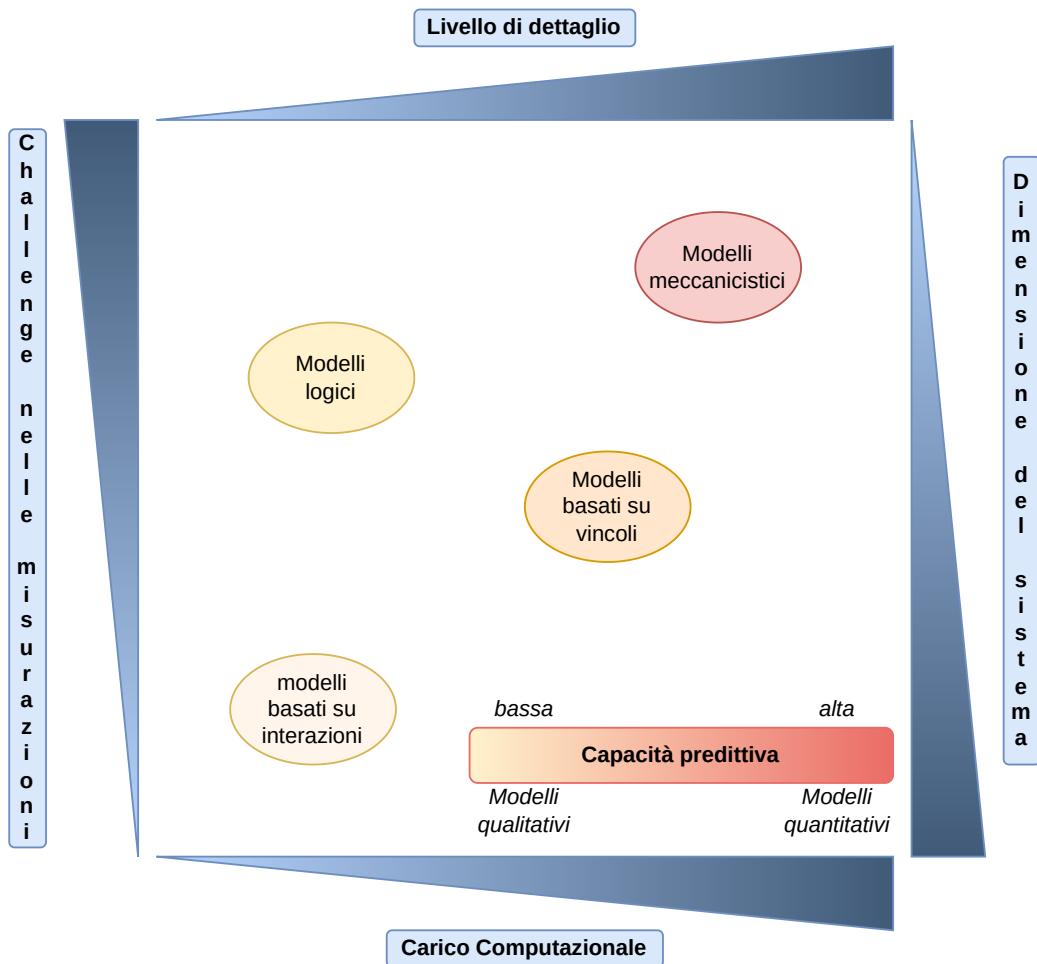


Figura 2.6: Schema riassuntivo dei quattro approcci

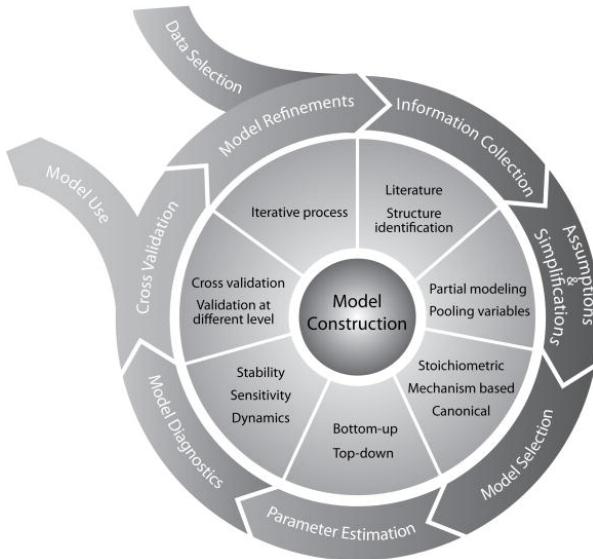


Figura 2.7: Raffigurazione che mostra i dettagli del processo circolare di modellazione in *systems biology*. Molte, ma non tutte, delle keyword presenti verranno approfondite nel corso

modello stesso, in un processo circolare, come visualizzabile in figura 2.7⁹. Quindi partendo dai dati biologici si abbozza un primo modello che viene poi continuamente sistemato tramite nuove ipotesi, altri dati, analisi *in silico* etc... fino all'ottenimento di un modello validato. Vediamo ora una breve introduzione ai quattro approcci elencati al fine di poter fare un confronto tra essi prima di studiarli e formalizzarli nel dettaglio.

Prima di fare ciò diamo una più precisa idea dei criteri con cui si classifica un modello.

Definizione 7. Si definisce **modello qualitativo** un modello che specifica le interazioni tra le componenti del modello stesso.

Definizione 8. Si definisce **modello quantitativo** un modello che assegna un valore ad ogni elemento che descrive e anche alle interazioni tra essi. In questo caso si possono avere o non avere cambiamenti nel modello.

Definizione 9. Si definisce **modello deterministico** un modello per il quale l'evoluzione attraverso i vari stati può essere predetta a partire dallo stato corrente, nel dettaglio anche dallo stato iniziale. Il comportamento evoluzionario del modello quindi non varierà tra una simulazione e l'altra.

⁹Chou IC, Voit EO. Recent developments in parameter estimation and structure identification of biochemical and genomic systems. *Math Biosci.* 2009;219(2):57-83. doi:10.1016/j.mbs.2009.03.002

Definizione 10. Si definisce **modello stocastico** un modello che descrive, a partire da uno stato corrente, uno stato futuro attraverso una distribuzione di probabilità.

Definizione 11. Un processo è detto **reversibile/irreversibile** se si può o meno procedere in avanti o indietro tra i vari stati.

Definizione 12. Con il termine **periodicità** si intende che il sistema assume una serie di stati nell'intervallo di tempo $[t, t + \Delta t]$ ma anche in:

$$[t + i\Delta t, t + (i + 1)\Delta t], \quad i = 1, 2, 3, \dots$$

2.4.1 Modelli Basati su Interazioni

Questo tipo di modelli vengono usati per *sistemi large-scale* con centinaia o migliaia di componenti che interagiscono tra loro in modo fisico o funzionale. Abbiamo vari esempi di questi modelli, tra cui:

- **reti di interazioni proteina-proteina**
- **reti di regolazione genica**
- **reti metaboliche**
- **reti di malattie**, modelli più complessi, modellati tramite un particolare tipo di grafo, che sfruttano l'integrazione tra *reti di regolazione genica* e grafi/reti rappresentanti le malattie e le relazioni tra esse. Si ottiene quindi un grafo che mette in relazione componenti genomiche e malattie

In questo caso la scelta del formalismo matematico ricade principalmente sulla **teoria dei grafi** e si ha quindi un *modello qualitativo e statico*. Infatti il *tempo* non viene considerato in tali modelli, che di conseguenza non permettono di ottenere informazioni su eventuali **comportamenti emergenti**. Non si possono nemmeno ottenere informazioni quantitative.

Parlando di *modelli basati su interazioni* non si può propriamente parlare di “simulazioni” vere e proprie in quanto in primis manca la modellazione del *tempo* ma anche di altri fattori come il *kinetic rate*. Inoltre tali modelli difettano anche di una qualsivoglia modellazione dello *spazio*. Il fulcro dello studio di tali modelli quindi solitamente si concerta sulle proprietà “architettoniche” della struttura della rete, studiando, ad esempio:

- la presenza di **hub**, ovvero nodi in cui sono entranti/uscenti un gran numero di archi rispetto agli altri nodi della rete

- misure di centralità
- presenza di *motivi (motifs)* nella rete
- la robustezza topologica

Tutte queste misure permettono anche di caratterizzare, caratterizzando la topologia stessa, una rete rispetto ad un'altra. Infatti si vedranno vari tipologie di rete, tra cui:

- **random network**
- **scale-free network**, caratterizzate da una forte *robustezza*
- **hierarchical network**

2.4.2 Modelli Logici

Questi modelli possono essere usati sia per sistemi *small-scale* che per sistemi *large-scale* e alcuni degli esempi sono:

- **reti di regolazione gene-gene**
- **pathway di trasduzione del segnale** (che si ricorda essere la capacità di una cellula di convertire uno stimolo esterno in una particolare risposta cellulare)
- **differenziazione cellulare**
- **pathway per la morte cellulare programmata**

Il primo caso è un esempio di *sistema large-scale* mentre gli altri di *sistemi small-scale*.

Dal punto di vista del formalismo matematico si ha anche qui la **teoria dei grafi**, a cui viene aggiunta la **logica booleana**, con i classici operatori logici \neg, \wedge, \vee , o anche, preferibilmente, la **logica fuzzy**, che verrà approfondita più avanti. L'idea di base è quella che lo stato delle componenti è regolato da altre componenti del sistema stesso. I nodi possono assumere o valore booleano 0/1 o, in logica fuzzy, qualsiasi valore tra 0 e 1 (con varie conseguenze nel loro studio).

Tali modelli sono in grado di simulare il tempo, rientrando quindi nella categoria dei *sistemi dinamici* ma sono anch'essi della tipologia dei *modelli qualitativi*. Tali sistemi si prestano ad essere sia *deterministici* che *non deterministici*.

Lo studio di tali modelli solitamente consiste, tramite le simulazioni e le analisi, nel determinare:

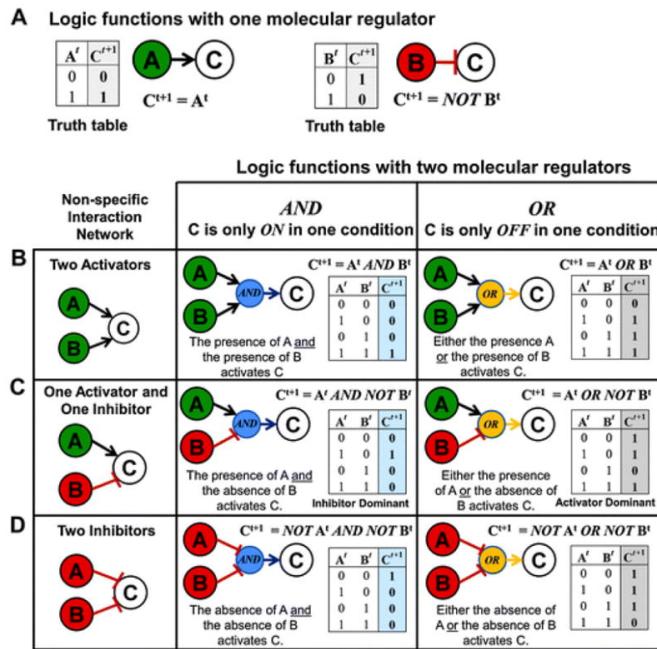


Figura 2.8: Esempio di modellazione di interazioni tra componenti tramite funzioni logiche.

- **cicli**, ovvero sequenze finite di stati complessivi del sistema che si ripetono
- **attrattori**, ovvero degli *stati finali* che sono raggiungibili da qualsiasi stato iniziale e una volta raggiunti si resta in tali stati
- **bacini di attrattori**, ovvero percorsi che partono da stati intermedi e che conducono a degli *attrattori*

Tendenzialmente si arriva sempre ad un *ciclo* o ad un insieme di *attrattori*. La “potenza” della *logica fuzzy* permette, come detto, anche di modellare il *tempo*, derivando quindi un comportamento dinamico del sistema, descrivendo, ad esempio, la variazione nel tempo tra i valori degli stati di ogni componente.

Un esempio semplice di quello che si può ottenere con tali modelli è visualizzabile in figura 2.8¹⁰.

¹⁰Wynn ML, Consul N, Merajver SD, Schnell S. Logic-based models in systems biology: a predictive and parameter-free network analysis method. Integr Biol (Camb). 2012;4(11):1323-1337. doi:10.1039/c2ib20193c

2.4.3 Modelli Meccanicistici

Come già anticipato tali modelli si limitano a descrivere *sistemi small-scale*. Questa è la classe di approcci modellistici più complessa ed eterogenea infatti, in primis, richiede una parametrizzazione completa delle componenti, con un ampio range di formalismi matematici, tra cui spiccano tra gli altri i **metodi numerici** e gli **algoritmi di simulazione stocastica**. Un problema, già solo a questo punto, è che non si hanno spesso i dati per effettuare la parametrizzazione in quanto i biologi/biotecnologi spesso non sono interessati a misurarli.

Le simulazioni con questi modelli sono usate per studiare l'**evoluzione nel tempo**, quindi la dinamica, del sistema. Si usano metodi *deterministici*, *stocastici* e *ibridi*, insieme ad una serie infinita di altre tecniche computazionali, tra cui l'*analisi di sensitività* o il *parameters sweeping*.

Si arriva quindi a modelli *quantitativi* e *dinamici*.

La scelta tra metodi stocastici, solitamente più dispendiosi, e deterministici dipende anche dal fatto che **la vita non è deterministica**. Ad esempio modellare le interazioni tra, ad esempio tra la proteina *Mdm3* e la proteina *p53*, avendo che la prima inibisce la seconda, comporta una funzione molto pulita se studiata in modo deterministico quando in realtà, a causa di molti fattori, non si ha tale precisione se si va a studiare cosa accade realmente in natura. Da qui l'uso anche di *modelli stocastici*.

La stima dei parametri resta comunque un grandissimo problema e spesso si usano altre tecniche computazionali/modellistiche in pipeline per inferire gli stessi.

2.4.4 Modelli Basati su Vincoli

Tali modelli sono usati esplicitamente e solo per *sistemi large-scale per reti metaboliche*. Il formalismo matematico qui usato si compone di **matrici stoichiometriche**, **algebra lineare** e tecniche di **ricerca operativa** mentre le simulazioni e le analisi consistono nello studiare le variazioni nelle **distribuzione di flusso**, calcolando i valori di flusso di tutte le reazioni metaboliche, a seconda di perturbazioni/input prefissati.

Non è semplice se si può dire di ottenere dei *sistemi quantitativi* in quanto si studia il comportamento ad uno *steady state*.

Tra gli esempi di uso si hanno:

- l'**ingegnerizzazione metabolica**, ovvero l'ottimizzazione, il design e la regolarizzazione di certe strutture/funzioni metaboliche al fine di ottenere un certo fenotipo metabolico

- studiare **bersagli di farmaci**, attraverso ad esempio lo studio del *rewiring metabolico del cancro*

L'idea è quindi quella di:

- stabilire dei **vincoli**
- stabilire una **funzione obiettivo** da *massimizzare/minimizzare*
- determinare automaticamente la distribuzione dei flussi

Si parla di **Flux Balance Analysis (FBA)**.

2.4.5 Confronto tra i Vari Approcci

Viste queste prime piccole premesse sui quattro approcci possiamo fare qualche piccolo confronto.

In primis abbiamo capito come lo studio del *tempo* sia assente del tutto nei *modelli basati su interazioni* e che sia di dubbio uso nel caso dei *modelli basati su vincoli* a causa dello *steady state*. Quindi se si dovesse, ad esempio, studiare il cambio di concertazione di una certa molecola al variare del *tempo* tali approcci sarebbero da scartare a priori.

Si è anche visto come, in realtà, praticamente solo i *modelli meccanicistici* ci offrono uno studio quantitativo, al costo di una complessità sia formale, che di dati, che computazionale molto alta e riducendosi a studiare solo sistemi piccoli. Ne segue che:

I sistemi meccanicistici sono l'approccio modellistico migliore per comprendere e acquisire nuove intuizioni il funzionamento del sistema.

Nella realtà però “avere tutto” è un’utopia quindi non si ha un vero e proprio vincitore in questa “gara tra approcci modellistici”, ben riassunti nella figura 2.9¹¹, in quanto al variare del problema, dei dati, e di mille altri fattori potrei aver motivi validi per preferire un approccio ad un altro.

Inoltre, in questa breve introduzione, si è scoperto come ci siano moltissime **dicotomie** in *systems biology*:

- *top-down* e *bottom-up*
- *qualitativo* e *quantitativo*

¹¹Bordbar, A., Monk, J., King, Z. et al. Constraint-based models predict metabolic and associated cellular functions. Nat Rev Genet 15, 107–120 (2014). <https://doi.org/10.1038/nrg3643>

Method	Model systems	Parameterization	Typical prediction type	Advantages	Disadvantages
Stochastic kinetic modelling	Small-scale biological processes	Detailed kinetic parameters	Reaction fluxes, component concentrations and regulatory states	<ul style="list-style-type: none"> Mechanistic Dynamic Captures biological stochasticity and biophysics 	<ul style="list-style-type: none"> Computationally intensive Difficult to parameterize Challenging to model multiple timescales
Deterministic kinetic modelling	Small-scale biological processes	Detailed kinetic parameters	Reaction fluxes, component concentrations and regulatory states	<ul style="list-style-type: none"> Mechanistic Dynamic 	<ul style="list-style-type: none"> Computationally intensive Difficult to parameterize
Constraint-based modelling	Genome-scale metabolism	Network topology, and uptake and secretion rates	Metabolic flux states and gene essentiality	<ul style="list-style-type: none"> Mechanistic Large scale No kinetic information is required 	<ul style="list-style-type: none"> No inherent dynamic or regulatory predictions No explicit representation of metabolic concentrations
Logical, Boolean or rule-based formalisms	Signalling networks and transcriptional regulatory networks	Rule-based interaction network	Global activity states and on-off states of genes	Can model dynamics and regulation	Biological systems are rarely discrete
Bayesian approaches	Gene regulatory networks and signalling networks	High-throughput data sets	Probability distribution score	<ul style="list-style-type: none"> Non-biased Can include disparate and even non-biological data Takes previous associations into account 	<ul style="list-style-type: none"> Statistical Issues of over-fitting Requires comprehensive training data
Graph and interaction networks	Protein–protein and genetic interaction networks	Interaction network that is based on biological data	Enriched clusters of genes and proteins	<ul style="list-style-type: none"> Incorporates prior biological data Encompasses most cellular processes 	Dynamics are not explicitly represented
Pathway enrichment analysis	Metabolic and signalling networks	Pathway databases (for example, KEGG, Gene Ontology and BioCyc)	Enriched pathways	<ul style="list-style-type: none"> Simple and quick Takes prior knowledge into account 	<ul style="list-style-type: none"> Biased to human-defined pathways Non-modelling approach

Figura 2.9: Schema riassuntivo delle caratteristiche dei vari approcci.

- *statico e dinamico*
- *deterministico e stocastico*
- *discreto e continuo* (sia in ottica di rappresentazione del *tempo* che della numerazione delle componenti)
- *omogeneo e eterogeneo*
- *a singolo volume e multicompartmentale*

Tutte queste dicotomie rappresentano la complessità degli studi in *systems biology*.

Integrare i vari modelli è per lo più utopia. Fare *data integration* è già di per se uno scoglio complesso ma si aggiunge anche la difficoltà di integrare vari formalismi matematici. Non si ha il modello “perfetto” ma si può scegliere bene in base al sistema biologico da studiare, magari integrando anche qualche (molto pochi) approccio modellistico diverso, come visibile in figura 2.10¹². Nel diagramma si segnala il paper centrale di Karr et al.: *A whole-*

¹²Gonçalves E, Bucher J, Ryll A, et al. Bridging the layers: towards integration of signal transduction, regulation and metabolism into mathematical models. Molecular Biosystems. 2013 Jul;9(7):1576-1583. DOI: 10.1039/c3mb25489e. PMID: 23525368.

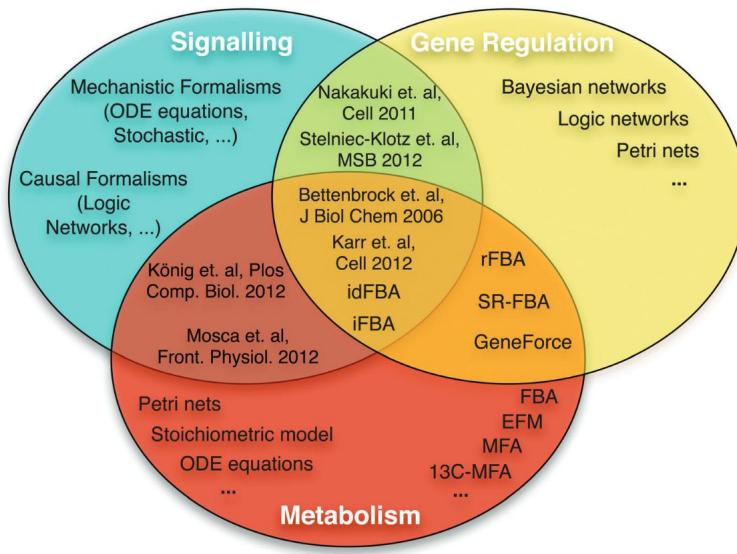


Figura 2.10: Diagramma che mostra varie soluzioni modellistiche al variare del problema biologico

cell computational model predicts phenotype from genotype¹³ cruciale nello studio di un approccio misto per modellare il sistema di un'intera cellula di un piccolo batterio.

¹³Karr JR, Sanghvi JC, Macklin DN, et al. A whole-cell computational model predicts phenotype from genotype. *Cell.* 2012;150(2):389-401. doi:10.1016/j.cell.2012.05.044

Capitolo 3

Interaction-Based Modelling

Si parte con la descrizione della prima classe di modelli, parlando quindi della **interaction-based modelling**.

Ricordiamo che tali modelli, come visibile in figura 2.6:

- hanno un sistema di grandi dimensioni, con centinaia o migliaia di componenti (se non di più), essendo quindi *modelli large-scale*
- presentano tendenzialmente un basso costo computazionale per le analisi
- hanno un basso livello di dettaglio
- non presentano particolari difficoltà nella misurazione dei dati

Inoltre, sempre ricordando l'introduzione alla classe, questo approccio modelistico:

- non permette propriamente di parlare di simulazioni non modelando *tempo, localizzazione spaziale e kinetic-rates*
- si indirizzano le analisi verso lo studio topologico della rete
- si studiano proprietà emergenti prettamente strutturali quali la presenza di *hubs*, misure di centralità, presenza di *motifs* e *robustezza topologica* contro certe *perturbazioni*

Abbiamo quindi a che fare con **modelli qualitativi e statici**.

Come anticipato possiamo avere vari tipi di relazione tra i nodi della rete in questo modello, ovvero vari tipi di **interazioni**.

Vediamo qualche esempio¹ (*nel corso si approfondiranno solo i primi tre esempi, a livello genico e proteico*):

- **interazioni fisiche**, ovvero interazioni che si verificano tra biomolecole a diretto contatto. Ad esempio, le reti proteina-proteina con tali interazioni sono importanti in processi come la formazione di complessi proteici, la trasduzione del segnale e il trasporto. Sono tendenzialmente il caso di studio più semplice
- **interazioni di regolazione**, ovvero interazioni che sono eventi di attivazione o inibizione diretta. Ad esempio, nella regolazione dell'espressione genica, un fattore di trascrizione è collegato ai suoi bersagli da archi diretti nella rete. Quindi posso quindi avere sia *regolazioni positive* che *regolazioni negative* e non si hanno più *interazioni fisiche*
- **interazioni genetiche**, ovvero interazioni che connettono geni la cui simultanea perturbazione genetica porta a un risultato fenotipico diverso da quello previsto dalla combinazione di singoli effetti. Ad esempio, le interazioni letali sintetiche collegano i geni che influenzano debolmente la vitalità dell'organismo quando eliminati individualmente, ma sono letali quando eliminati in combinazione. Le *interazioni genetiche* sono utili per studiare la funzione genica e per identificare complessi e pathway che lavorano insieme per controllare le funzioni essenziali
- **relazioni di similarità**, avendo collegamenti tra oggetti biologici che sono *simili* secondo un attributo comune. È possibile utilizzare molte diverse misure di somiglianza, come la somiglianza della sequenza proteica o la coespressione genica basata su profili trascrizionali correlati (avendo sia correlazioni positive che negative). Le relazioni di somiglianza sono utili per identificare gruppi di geni o proteine funzionalmente correlati. Un altro esempio è quello di studiare se, in una certa condizione data, si possono identificare geni indipendenti con un profilo di espressione (ovvero descrizione qualitativa e quantitativa dell'insieme dei geni trascritti in un dato momento da una cellula o da un tessuto, studiabile tramite, ad esempio, l'uso dei *microarrays*)

¹Merico D, Gfeller D, Bader GD. How to visually interpret biological data using networks. Nat Biotechnol. 2009 Oct;27(10):921-4. doi: 10.1038/nbt.1567. PMID: 19816451; PMCID: PMC4154490.

Quindi in generale si hanno tipi modelli di reti di interazioni associati ai vari tipi di interazione (fisica, funzionale, genica, etc...), ad esempio²:

- **Association networks**, che modellano qualsiasi tipo di relazione tra molecole, ad esempio legami, coespressione e somiglianze strutturali. Esempi di tali reti sono le **gene co-expression networks** e le **protein similarity networks**
- **Functional networks**, che modellano le relazioni funzionali tra coppie di molecole (solitamente geni o proteine). Un collegamento implica che entrambe le componenti sono coinvolte nella stessa funzione, processo o fenotipo. Un esempio sono le **genetic interaction networks** rappresentano interazioni in cui una doppia mutazione porta a un effetto epistatico (che si ha quando una coppia di alleli copre l'espressione fenotipica di un'altra coppia di alleli), ad esempio peggiore o migliore del previsto rispetto alla singola mutazione
- **Protein-Protein Interaction (PPI) networks**, che sono reti non dirette che modellano il legame proteico tra le componenti, che sono appunto proteine. Tali reti sono derivate da esperimenti ad alto rendimento che utilizzano tecniche come lo *screening di due ibridi di lievito*, la *spettrometria di massa* e la *purificazione per affinità tandem*, che sono metodi *high-throughput*. Le **signaling networks** sono correlate alle reti PPI ma in questo caso i collegamenti sono diretti in base al flusso di segnali molecolari
- **Transcription-Regulatory (TR) networks**, tali reti sono reti bipartite con un insieme di nodi che rappresentano i geni e l'altro che rappresenta i *fattori di trascrizione (TF, da "transcription factors")*. I TF sono prodotti di geni (modellati da collegamenti *gene-TF*) mentre i geni sono regolati dai TF (modellati da collegamenti *TF-gene*). I dati per tali reti sono derivati attraverso il processo di immunoprecipitazione della cromatina (detto *ChIP*). Le **gene regulatory networks** sono correlate alle reti TR ma contengono solo geni e i collegamenti rappresentano relazioni regolatorie indirette
- **Metabolic networks** che sono reti bipartite che modellano le relazioni tra le reazioni chimiche che si verificano nelle cellule e i

²Winterbach W, Van Mieghem P, Reinders M, Wang H, de Ridder D. Topology of molecular interaction networks. BMC Syst Biol. 2013;7:90. Published 2013 Sep 16. doi:10.1186/1752-0509-7-90

substrati coinvolti nelle reazioni. Spesso vengono studiate anche reti metaboliche ridotte e non bipartite contenenti solo metaboliti o solo reazioni. Parlando di metabolismo ovviamente i risultati che ottengo tramite queste reti sono diversi da quelli che otterrei usando, ad esempio, un *modello basato su vincoli*

L'analisi della topologia delle reti è formata quindi da:

- la **teoria dei grafi**, mediante la quale si misura il sistema
- le **misure di centralità**, con le quali si analizza (e non si simula) la rete
- la **classificazione della rete**, ovvero la caratterizzazione della sua topologia, e la **robustezza topologica**, che sono i risultati delle analisi. Tali risultati possono anche corrispondere a nuovi “insight” biologici

Se aggiungiamo funzioni logiche per descrivere come cambia lo stato di ogni nodo nel tempo, possiamo effettuare un'analisi dinamica di una rete. Si hanno quindi i modelli basati sulla logica, che aggiungono complessità, sia formale che computazionale, al fine di raggiungere ulteriori risultati.

3.1 Introduzione alle Reti PPI

Prima di approfondire la classe di modelli si vede un piccolo approfondimento sulle **Protein-Protein Interaction (PPI) networks**, come l'esempio in figura 2.4 (anche se si ricorda che la rappresentazione di un sistema è sempre limitante), anche al fine di capire il motivo biologico e i limiti tecnici. Gli algoritmi tipici della teoria dei grafi ci permetteranno di studiarne la topologia.

Questo sarà comunque il modello più studiato in questo corso per questa classe.

Ovviamente le reti sono formalizzate come dei **grafi** e, questo caso, si hanno:

- i *nodi* che rappresentano le proteine
- gli *archi*, che non sono orientati, che corrispondono alle interazioni di legame tra coppie di proteine

Come già anticipato le *reti PPI*, più precisamente, parlando di sistemi *large-scale*, **large-scale PPI** vengono costruite a partire da dati proteomici ottenuti tramite metodi *high-throughput*, ben definiti e con protocolli chiari (per

questo non si hanno particolari challenge dal punto di vista dell'ottenimento dati). Le *reti large-scale PPI* sono però anche caratterizzate da:

- **conflitti**, in quanto tali reti sono ottenute rappresentando tutti i casi possibili, ignorando appunto la componente temporale. Ne segue quindi che, in realtà, le interfacce proteiche possono legare molte altre diverse in modo mutuamente esclusivo, avendo magari che alcuni legami possono avvenire in modo mutuamente esclusivo in un dato tempo. Rappresentare quindi “tutti i tempi” può creare ambiguità. L’idea di togliere componenti al modello per evitare ambiguità non è una buona idea in quanto si toglierebbe senso al modello (che già ha poca capacità predittiva)
- **complessità combinatoria** in quanto si ha un’esplosione del numero di complessi distinti che possono essere formati da una rete di, appunto, “possibili” legami

Il sistema è quindi *statico*, non cambiando nel tempo, che non viene nemmeno rappresentato. Il tempo non è comunque un fattore rilevante per gli studi che si fanno su tali modelli.

Si hanno anche alte criticità, ad esempio:

- gli archi in una *rete PPI* non rappresentano necessariamente connessioni fisiche persistenti, ma piuttosto riassumono le possibilità di interazione, come già detto, e quindi si lascia spazio a:
 - *gaps* sia per i nodi che per gli archi, ovvero nodi/archi non rappresentati in quanto non si conosce una certa interazione (non ancora studiata nella letteratura scientifica)
 - interazioni che sono *false positive* o *false negative*
- le interazioni proteina-proteina nell’intera rete non si verificano, come detto, necessariamente contemporaneamente e/o utilizzando domini di legame diversi e/o nello stesso compartimento cellulare. In altre parole il non rappresentare dove accade una certa interazione può essere un problema. Si “pone tutto allo stesso livello”
- come non si hanno informazioni temporali non si hanno nemmeno informazioni quantitative riguardo al funzionamento della cellula, al suo stato, al ciclo cellulare etc. . .

- si rischia un bias dovuto al fatto che alcune proteine hanno più connessioni semplicemente perché sono meglio studiate (si parla, quando si usa molto la letteratura come base del modello, di **literature-curated networks**). Un esempio banale è pensare che la proteina *p53* compare, a data Marzo 2019, in 94552 paper secondo PubMed (34205 direttamente nel titolo) mentre *Snf1* 1037 volte (solo 318 nel titolo)
- si hanno forti limiti di modellazione. Ad esempio non si può modellare che una proteina interagisca con altre sse queste due hanno prima avuto un’interazione tra loro (avendo che il massimo che posso avere è un “triangolo” nella rete)

Possiamo quindi concludere che **i risultati dello studio di una rete PPI (ma anche delle altre reti tipiche dell’interaction-based modelling) non sono sempre così affidabili.**

3.2 La Teoria dei Grafi

Lo studio dei **grafi** è centrale in vari problemi/tecniche comuni, dal web ai social, dalle reti di collaborazione (pensando ad esempio al *erdős number*) a ipotesi come la famosa *ipotesi dei sei gradi di separazione* che dimostra (come visto sia nel collegare attori a Kevin Bacon che con lo studio del 1967 più “accademico” dello psicologo Stanley Milgram) come si abbiano di media sei passaggi per collegare due persone a caso nel mondo. Si parla infatti spesso della **teoria del mondo piccolo**, che è appunto una teoria matematica e sociologica che sostiene che tutte le reti complesse presenti in natura sono tali che due qualunque nodi possono essere collegati da un percorso costituito da un numero relativamente piccolo di collegamenti.

Definizione 13. *Si definisce formalmente un **grafo** G come una coppia di insiemi:*

$$G = \langle V, E \rangle$$

dove:

- $V = \{v_1, \dots, v_n\}$ è l’insieme dei **nodi**, di cardinalità n
- $E \subseteq V \times V$ è l’insieme degli **archi**, di cardinalità m . Si ha che $E = \{e_1, \dots, e_m\}$, dove un generico arco $e_k = (v_i, v_j)$ con $k = 1, \dots, m$, è definito come una coppia di vertici $v_i, v_j \in V$

Definizione 14. *In un grafo si definisce **nodo isolato** un nodo che non è connesso a nessun altro nodo.*

Definizione 15. In un grafo si definisce **cappio** un arco tra un nodo e se stesso.

Definizione 16. In un grafo $G = \langle V, E \rangle$ si definisce **percorso/cammino** come una sequenza finita o infinita di archi che unisce una sequenza di vertici. Dati quindi due nodi $v_1, v_n \in V$ un cammino da v_1 a v_n un cammino è una sequenza di nodi:

$$(v_1, v_2, \dots, v_n) \text{ tali che } \forall i = 1, 2, \dots, n-1, \exists e = (v_i, v_{i+1}) \in E$$

Se il vertice di partenza coincide con quello di fine si parla di **ciclo**, quindi $v_1 = v_n$.

Ovviamente tra due nodi si possono avere distinti cammini.

Il concetto di *cammino* e il concetto di *ciclo* assumono particolare rilevanza in biologia. I *cicli* servono, ad esempio, per rappresentare i *feedback* mentre i *cammini*, ad esempio, per le *vie di trasduzione del segnale* dove si parte da un recettore transmembrana per poi attivare una catena di reazioni.

Definizione 17. Si definisce **grafo连通的** un grafo dove esiste un percorso tra ogni coppia di vertici. In caso contrario si parla di **grafo non连通的**.

Definizione 18. Si definisce **grafo completo o grafo completamente连通的** un grafo, di n nodi, dove ogni nodo è collegato ai rimanenti $n - 1$ nodi.

Definizione 19. Si definisce **grafo totalmente sconnesso** un grafo che non presenta archi.

Definizione 20. Due nodi v_i e v_j si dicono **adiacenti** se esiste un arco, diretto o indiretto, $e = (v_i, v_j)$.

Definizione 21. Dato un nodo v in un grafo indiretto si definisce **grado/-connettività/degree** come il numero di nodi adiacenti ad esso. Tale valore solitamente si indica con k_v , che è ovviamente un numero intero non negativo. Qualora ci sia un cappio esso conta doppio.

Un **nodo isolato** presenta un grado nullo.

Definizione 22. Dato un grafo diretto si definiscono:

- **indegree** di un nodo v come il numero di archi entranti in v
- **outdegree** di un nodo v come il numero di archi uscenti da v

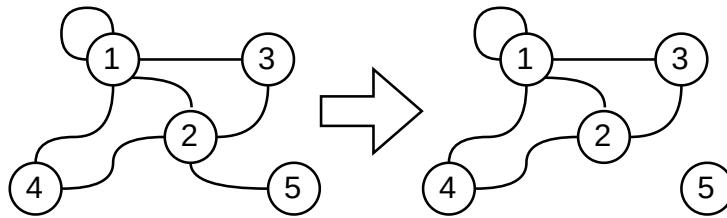


Figura 3.1: Esempio di passaggio da *grafo connesso* a *grafo non connesso* dopo un’ipotetica perturbazione, che produce il *nodo isolato* etichettato con 5.

In letteratura ha volte si definisce **grado/connettività/degree** in un grafo diretto come la somma di indegree e outdegree.

Nella modellazione di sistemi biologici mediante questa classe di modelli è interessante notare come si possa passare da un grafo connesso ad uno non connesso dopo l’influenza di una *perturbazione* sul sistema, che influisce sulle interazioni (e quindi sugli archi) tra le componenti. Tali cambiamenti hanno forte valenza dal punto di vista biologico in quanto se un sistema è propenso a produrre un grafo disconnesso dopo una perturbazione, come ad esempio in figura 3.1, allora è un sistema che è in generale propenso a “fallire”.

Definizione 23. Si definisce **grafo orientato** un grafo dove ogni arco consiste in una **coppia orientata** di vertici, altrimenti si parla di **grafo non orientato**. Formalmente si ha quindi, per un grafo orientato, con $v_1, v_2 \in V$:

$$e = (v_1, v_2) \neq e' = (v_2, v_1)$$

In tal caso, a livello grafico, gli archi sono rappresentati mediante frecce.

In questa classe modellistica si hanno:

- **grafi orientati** per *gene regulatory networks* e *signal transduction networks*
- **grafi non orientati** per *reti PPI*

Definizione 24. Dato un grafo $G = \langle V, E \rangle$ si definisce S come **sottografo** di G come una coppia di insiemi:

$$S = \langle V', E' \rangle$$

dove:

- $V' \subseteq V$

- $E' \subseteq E$

Possiamo quindi dire che, riprendendo la figura 3.1, mediante una *perturbazione*, si ottiene un sottografo del grafo di partenza.

I sottografi sono molto utili per studiare “caratteristiche” di forte interesse biologico, rappresentate appunto da sottografi³:

- **modules** che sono sottografi indotti la cui densità di archi è elevata rispetto al resto del grafo. Questa non è una vera e propria definizione in quanto al natura dei *modules* varia dal contesto e dall’algoritmo utilizzato per scoprirli
- **motifs** che sono piccoli sottografi, solitamente di tre o quattro nodi, la cui sovrarappresentazione o sottorappresentazione può indicare che le loro strutture sono “importanti” o “dannose” per il sistema. Di solito, vengono contati tutti i *motifs* distinti in una rete, ottenendo una *motif signature* per la rete che può quindi essere confrontata con le firme, ottenute campionando da un modello nullo di rete casuale appropriato, per determinare la sovrarappresentazione o sottorappresentazione. Le *motif signature* possono essere usate per caratterizzare le reti stesse
- **graphlets** che sono simili ai *motifs* ma sono *completamente connessi*. Anch’essi vengono utilizzati per costruire firme che catturano le caratteristiche locali di una rete

Questi argomenti verranno approfonditi più avanti, queste sono solo le “definizioni” recuperate nel paper indicato.

Torniamo a parlare della nozione di grado.

Definizione 25. Si definisce *distribuzione dei gradi/degree distribution* come la distribuzione di probabilità dei gradi dei nodi sull’intera rete. Tale distribuzione, denotata con $P(k)$, quindi è la probabilità che un certo nodo abbia grado esattamente pari a k . Tale probabilità si ottiene contando il numero di nodi del grafo, denotati $N(k)$, che presentano grado k e dividendo tale valore per il numero totale di nodi del grafo, che indichiamo con $N = |V|$. Si ha quindi:

$$P(k) = \frac{N(k)}{N}, \quad k = 1, 2, \dots$$

³Winterbach W, Van Mieghem P, Reinders M, Wang H, de Ridder D. Topology of molecular interaction networks. BMC Syst Biol. 2013;7:90. Published 2013 Sep 16. doi:10.1186/1752-0509-7-90

Avendo una distribuzione di probabilità ne segue che:

$$\sum_{i=1}^{k_{max}} P(i) = 1$$

La **degree distribution** ci permette di classificare un grafo, anche solo piazzando con un istogramma i valori di $P(k)$ al variare di k stesso. Ad esempio qualora si avesse un picco nel plot di tali valori allora si avrebbe che la rete ha un “grado caratteristico” (estremizzando l’esempio magari si ha una rete dove tutti i nodi hanno grado $k = 2$) e quindi non si hanno nodi fortemente connessi, con alto *degree*. Questo tipo di analisi non può essere fatta “visivamente” su reti reali ma ci si deve per forza affidare a conti precisi o al più plot della distribuzione stessa. Da tali studi posso estrarre alcune informazioni, ad esempio:

- i nodi con $k = 0$, ovvero i nodi isolati possono rilevare che ci sono probabilmente informazioni mancanti, falsi negativi etc... mentre il valore di $P(0)$ mi dice la probabilità stessa che un qualsiasi nodo della rete sia un nodo isolato
- i nodi con un k elevato sono tendenzialmente molto pochi e sono i cosiddetti **hubs**, che hanno un ruolo chiave nello studio di *reti large-scale*

Al fine di rappresentare i vari valori si usa un piano cartesiano (spesso per necessità rappresentato in *scala logaritmica*) dove:

- l’asse delle x è formato dai valori di k
- l’asse delle y è formato dai valori di $P(k)$

Rappresentando tutte le varie coppie $\langle k, P(k) \rangle$ si ottiene una “forma” che è la cosiddetta **power-law degree distribution** (che rappresenta la relazione funzionale tra k e $P(k)$ dove una variazione relativa in una delle due quantità si traduce in una variazione relativa proporzionale nell’altra quantità, indipendentemente dalla dimensione iniziale delle due quantità, avendo quindi che una quantità varia come potenza di un’altra). Lo studio della *power-law degree distribution* è spesso essenziale nello studio di sistemi biologici.

Sempre restando sullo stesso discorso si è notato che in molte reti se si ha un nodo v_i connesso con un nodo v_j , che a sua volta è connesso al nodo v_h , allora è altamente probabile che v_i sia anch’esso collegato al nodo v_h . Questo **fenomeno di clustering** può essere quantificato usando il cosiddetto **coefficiente di clustering**.

Definizione 26. Dato un nodo v si definisce il **coefficiente di clustering** del nodo v , denotato con C_v , come il numero di archi che connettono nodi adiacenti a v diviso il numero totale delle possibili connessioni che si avrebbero tra i nodi adiacenti a v . Formalmente:

$$C_v = \frac{2N_v}{k_v(k_v - 1)}$$

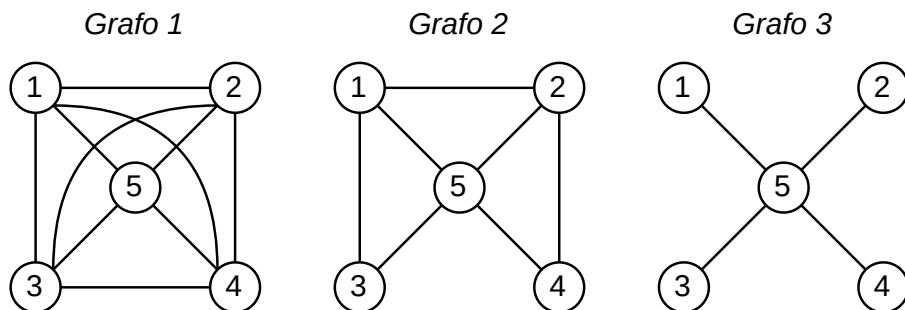
infatti si hanno:

- N_v come numero di archi che connettono coppie di nodi adiacenti a v . Questo valore è facilmente contabile avendo il grafo
- $\frac{k_v(k_v-1)}{2}$, parlando di grafo indiretto, come numero di tutti i possibili archi tra coppie di nodi adiacenti a v , che ha grado k_v , un valore conosciuto

Si osserva inoltre che, ricordando che alla fine si ha a che fare con la misura di probabilità di quanto sia probabile che si abbia o meno un cluster che include il nodo v :

$$0 \leq C_v \leq 1$$

Esempio 1. Vediamo un semplice esempio che mostra come varia il coefficiente di clustering. Si studia, nel dettaglio, il valore C_5 nei seguenti grafi:



In tutti i casi si ha $k_5 = 4$ ma:

- nel Grafo 1 si ha $N_5 = 6$ e $C_5 = \frac{12}{12} = 1$ e infatti il nodo 5 è sicuramente in cluster
- nel Grafo 2 si ha $N_5 = 3$ e $C_5 = \frac{6}{12} = 0.5$
- nel Grafo 3 si ha $N_5 = 0$ e $C_5 = \frac{0}{12} = 0$ e infatti il nodo 5 non è sicuramente in cluster

Questo tipo di coefficiente è utile soprattutto nel caso di *grafo indiretti*. Nel caso di *grafo diretti* bisogna invece ragionare in ottica di *indegree* e *outdegree*.

Un caso limite interessante in ottica di *coefficiente di clustering* è quello della **clique**.

Definizione 27. Si definisce **clique (cricca)** di un grafo non orientato $G = \langle V, E \rangle$ come un sottoinsieme $V' \subseteq V$ di vertici tale che:

$$(v_1, v_2) \in E, \forall v_1, v_2 \in V'$$

quindi un sottoinsieme di vertici con solo vertici collegati da un arco.

Il caso della *clique* è quindi il “caso migliore” parlando del fenomeno del clustering.

Il singolo *coefficiente di clustering* di un nodo comunque non è di particolare interesse se preso in modo isolato, in quanto si vuole classificare l’intera rete.

Definizione 28. Si definisce il **coefficiente di clustering medio**, denotato con $\langle C \rangle$, il valore medio di tutti i coefficienti di clustering dei nodi del grafo.

Il *coefficiente di clustering medio* permette di caratterizzare la tendenza complessiva dei nodi di una rete a formare gruppi o cluster. Questo è quindi un valore che ci permette di caratterizzare la topologia di una rete.

Definizione 29. Si definisce la funzione $C(k)$, che potremmo chiamare “**average clustering distribution**” o **average clustering coefficient**, come la media dei coefficienti di clustering di tutti i nodi di grado pari a k nella rete.

Il valore $C(k)$ quindi fornisce un’indicazione del carattere modulare/ge-rarchico di una rete, ovvero l’esistenza di sottografi/sottoreti caratterizzati da nodi fortemente collegati internamente, che presentano scarse connessioni con altre parti della rete.

Definizione 30. Si definisce la **lunghezza del cammino** tra due nodi v_1 e v_n come il numero di archi che si hanno nel cammino.

Si definisce **cammino minimo**, notando che potrebbe non essere unico, un cammino di lunghezza minima tra due nodi.

Anche la nozione di *lunghezza del cammino* ha un ruolo centrale nella modellazione di sistemi biologici. Basti pensare, in modo comunque approssimato e semplicistico, che più è lungo il cammino e più sono le interazioni biologiche e quindi, ad esempio, più energia è richiesta alla cellula. Infatti

normalmente la natura ha fatto sì che ogni “operazione biologica” venga fatta nel modo meno dispendioso possibile, quindi, ricollegando la modellazione a grafo, mediante *cammini minimi*. Nella realtà si vedrà il concetto di **ridondanza** in quanto si hanno vari modi, nel mondo biologico, per ottenere lo stesso risultato anche se con “cammini” di lunghezza diversa (magari per casi, ad esempio, in cui alla cellula conviene “temporeggiare”). Inoltre magari ad un percorso più lungo può comunque corrispondere un dispendio energetico minore. In ogni caso l’aggiunta di **pesi** al grafo permette una modellazione leggermente più precisa, potendo rappresentare ad esempio, i “costi” delle reazioni etc...

Un’altra cosa interessante da notare nella maggior parte delle reti è che esiste un cammino relativamente breve tra qualsiasi coppia di nodi e la lunghezza media di tale cammino è proporzionale al logaritmo della dimensione della rete, quindi al numero totale di nodi. Questa è la cosiddetta **small world property** che sembra caratterizzare la maggior parte delle reti complesse, comprese *reti metaboliche* e *reti PPI*.

Definizione 31. Si definisce un **grafo pesato** $G = \langle V, E \rangle$ come un grafo a cui viene associata anche una **funzione di peso** w :

$$w : E \rightarrow \mathbb{R}$$

Esistono, oltre a quelle già citate, anche altre metriche di studio, tra cui⁴:

- ulteriori **metriche per i cammini** come il *cammino minimo* su *grafo pesati* o il *cammino minimo medio* tra ogni coppia di nodi
- la **metrica di centralità** fornisce una classifica dei nodi in base alla loro ”importanza”. La versione più semplice sfrutta appunto il grado di un nodo per misurarne la centralità, parlando quindi di **degree centrality**. Un’alternativa è la **closeness centrality** che è il reciproco della somma dei cammini più brevi verso tutti gli altri nodi (cioè un nodo la cui *closeness centrality* è alta è vicino a molti nodi). Un’ulteriore metrica è la **betweenness centrality** ovvero la frazione di cammini minimi che passano attraverso un nodo. Tra le metriche più elaborate si annoverano la **centralità per autovettori** e il **pagerank** che sono misure della frequenza con cui si arriva a un nodo quando si esegue una *random walk* su una rete.

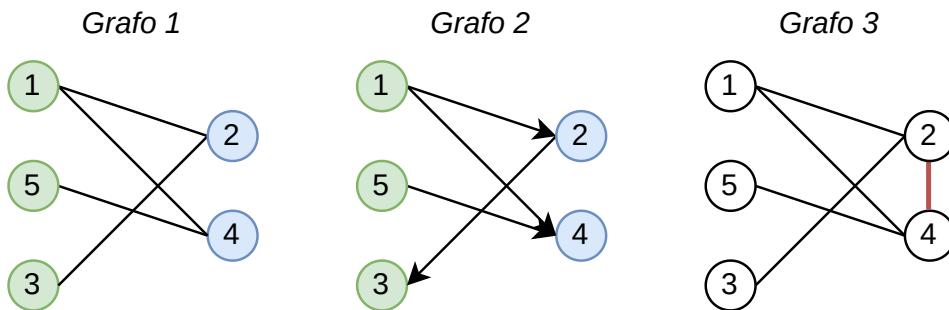
⁴Winterbach W, Van Mieghem P, Reinders M, Wang H, de Ridder D. Topology of molecular interaction networks. BMC Syst Biol. 2013;7:90. Published 2013 Sep 16. doi:10.1186/1752-0509-7-90

Queste non sono comunque metriche solitamente utili per la caratterizzazione della topologia di una rete.

Definizione 32. Si definisce un grafo $G = \langle V, E \rangle$, orientato o meno, come un **grafo bipartito** se:

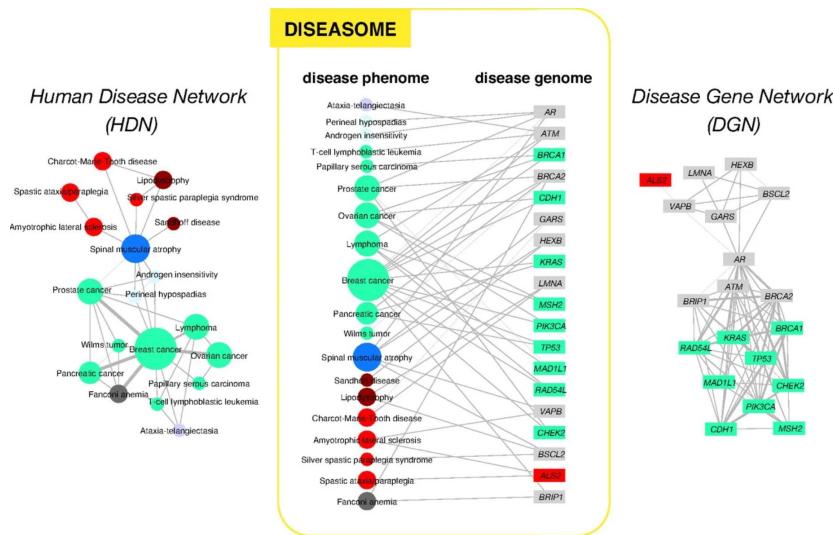
- l'insieme dei nodi V è in realtà l'unione di due sottoinsiemi di nodi V_1 e V_2 , ovvero $V = V_1 \cup V_2$, tali che la loro intersezione è nulla, ovvero $V_1 \cap V_2 = \emptyset$
- data la prima premessa si ha che ogni arco del grafo connette solo un nodo in V_1 ad un nodo in V_2

Ad esempio potremmo avere questi casi, dove i primi due grafi sono bipartiti (come evidenziato anche a livello visivo dai colori che identificano le partizioni) a differenza del terzo (a causa dell'arco in rosso):



Potenzialmente si possono anche avere **grafi tripartiti**, con 3 partizioni, o in generale **grafi multipartiti** con m partizioni.

Un esempio d'uso dei *grafo bipartito* sono le **human desaesome networks**, come ad esempio⁵:



Nella figura si hanno appunto due partizioni:

- un sottoinsieme di nodi per i geni
- un sottoinsieme di nodi per le malattie

Banalmente quindi un nodo che rappresenta una malattia è legato a un nodo che rappresenta un gene se è noto che una mutazione di quel gene induce l'insorgenza di quella malattia. A supporto posso inoltre avere due ulteriori reti solo per i geni e solo per le malattie.

Un esempio di rete basata su un *grafo multipartito* è, ad esempio, una **drug-target protein network**, come quella proposta da Nacher visualizzabile in figura 3.2⁶.

⁵Goh, Kwang-II, et al. "The human disease network." Proceedings of the National Academy of Sciences 104.21 (2007): 8685-8690.

⁶Nacher, J., Akutsu, T. Structural controllability of unidirectional bipartite networks. Sci Rep 3, 1647 (2013). <https://doi.org/10.1038/srep01647>

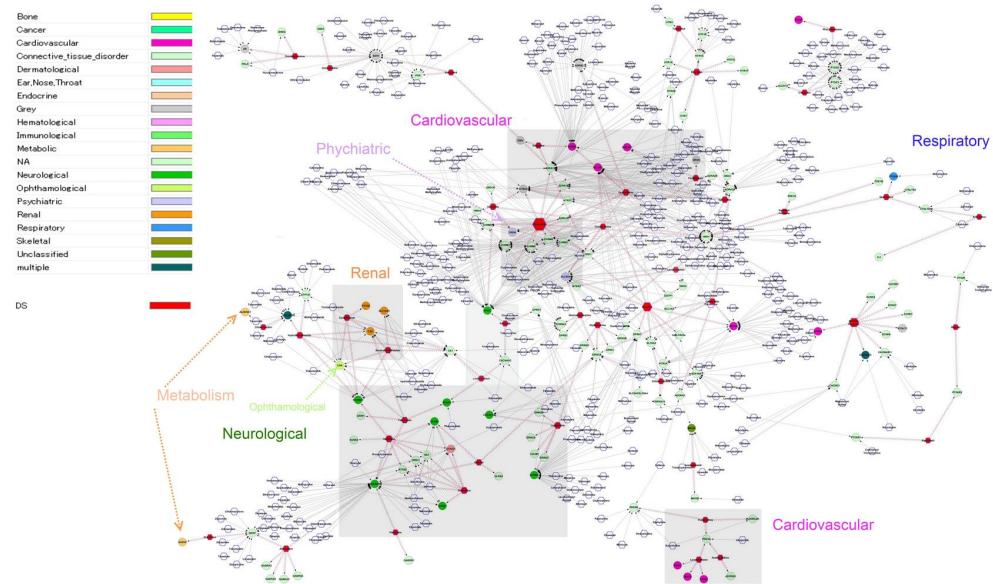
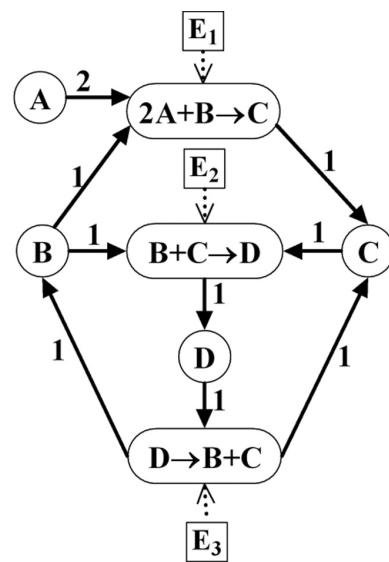


Figura 3.2: Esempio di *drug-target protein network*, rappresentata mediante grafo multipartito.

Un esempio invece di rete tripartita può essere la rappresentazione di un *pathway metabolico*⁷:



⁷Réka Albert; Scale-free networks in cell biology. J Cell Sci 1 November 2005; 118 (21): 4947–4957. doi: <https://doi.org/10.1242/jcs.02714>

dove si hanno tre tipologie di nodo:

1. nodi per i *reagenti* (nell'immagine rappresentati da cerchi)
2. nodi per le *reazioni* (nell'immagine rappresentate da ovali)
3. nodi per gli *enzimi* (nell'immagine rappresentati da quadrati)

Si hanno inoltre due tipologie di archi:

1. le linee solide per rappresentare il *mass flow*, ovvero il *tasso di turnover* delle molecole attraverso una via metabolica, tasso che serve ad indicare il dispendio energetico (???)
2. le linee tratteggiate per la *catalisi*, ovvero fenomeno chimico attraverso il quale la velocità di una reazione chimica subisce delle variazioni per l'intervento di una sostanza (o una miscela di sostanze) detta catalizzatore, che non viene consumata dal procedere della reazione stessa⁸

Inoltre i pesi degli archi indicano i *coefficienti stechiometrici*, che rappresentano infatti il rapporto tra le moli delle diverse sostanze, dei reagenti. Ovviamente l'uso delle reti in ambito biologico può essere espanso, considerando ad esempio i legami tra più reti, che rappresentano vari livelli, ad esempio:

- *reti sociali*, da usare comunque con cautela a causa della loro alta probabilità di portare *falsi positivi/negativi* anche se possono rappresentare informazioni importanti. Ormai si tende comunque a preferire il dato del singolo paziente, puntando alla *medicina personalizzata* in quanto “la media tra i pazienti” raramente è un dato utile. Esse possono rappresentare legami familiari, vicinanza tra persone, informazioni sui luoghi in cui si vive etc...
- *disease networks*
- *reti metaboliche*
- *reti PPI*
- *reti di regolazione genica*
- ...

⁸<https://it.wikipedia.org/wiki/Catalisi>

I vari livelli sono ovviamente connessi a vicenda ma non sempre è facile studiare tali connessioni a causa della mancanza di dati etc...

Altri esempi interessanti di uso sono le **reti in ecologia**, come, ad esempio, lo studio di Faust e Raes⁹ dove si studiavano le *interazioni micobiche*, tra cui il parassitismo, il mutualismo la competizione etc... tramite appunto delle reti.

Un uso recente delle reti è anche quello nelle **neuroscienze**, per capire, durante una malattia, cosa non stia funzionando bene nel cervello. Un esempio è lo studio di Chennu, Srivas et al.¹⁰ dove si è sfruttata la relazione tra reti e funzionalità del cervello per identificare quali aree del cervello/funzioni del cervello funzionassero male in pazienti in stato vegetativo e pazienti minimamente coscienti, facendo il paragone con vari soggetti controllo sani. Sono stati usati anche i concetti di grado etc... nello studio.

3.3 Tipologie di Reti

Un aspetto fondamentale nello studio delle reti è inoltre quello che sia la *degree distribution* $P(k)$ che il *average clustering coefficient* $C(k)$ sono **indipendenti** dalla grandezza delle reti e possono essere usati per identificare caratteristiche generali e classificare le reti stessi. Tra le tipologie più importanti abbiamo:

- **random network**
- **scale-free network**
- **hierarchical network**

Ad ogni tipologia ovviamente corrisponde un certo insieme di caratteristiche, come ad esempio le seguenti catalogate da Mitchell¹¹:

Network model	Degree distribution	Clustering coefficient	Average path length
Regular	constant	high	high
Random	Poisson	low	low
Watts–Strogatz small world (low, nonzero p)	depends on p	high	low
Barabási–Albert scale free	power law	high	low
Empirical results on real-world networks	power law	high	low

⁹Faust, K., Raes, J. Microbial interactions: from networks to models. Nat Rev Microbiol 10, 538–550 (2012). <https://doi.org/10.1038/nrmicro2832>

¹⁰Chennu, Srivas, et al. "Spectral signatures of reorganised brain networks in disorders of consciousness." PLoS computational biology 10.10 (2014): e1003887.

¹¹Mitchell, Melanie. (2006). Field review: Complex systems: Network thinking. Artificial Intelligence. 170. 1194-1212. 10.1016/j.artint.2006.10.002.

3.3.1 Random Network

Si comincia parlando delle **random network**, dette anche **Erdős-Rényi model**, dal nome di coloro che le formalizzarono nel 1960.

Si hanno quindi le seguenti caratteristiche principali per una *random network*:

- la rete è **statisticamente omogenea**, infatti la maggior parte dei nodi ha circa lo stesso numero di archi incidenti che quindi è vicino al grado medio della rete $\langle k \rangle$
- la *degree distribution* segue una **distribuzione di Poisson**, avendo quindi che si hanno davvero pochissimi nodi con un numero di archi incidenti maggiore o minore del valore medio. Si ha quindi, graficamente, una “campana” molto stretta sulla media del grado di tutti i nodi della rete, come visibile in figura 3.3¹²
- non si ha **modularità intrinseca**, non avendo quindi *moduli* e avendo che $C(k)$ è indipendente dal valore di k . Si hanno quindi pochi cluster
- sono caratterizzate dalla **small world property**

In merito all’ultimo punto bisogna formalizzare meglio quanto già anticipato. Quando tale proprietà è garantita si ha che la lunghezza media di un cammino tra due nodi è proporzionale a $\ln |V|$, assicurando quindi alta velocità di trasmissione delle “informazioni” nella rete. Questo è necessario nei sistemi biologici in quanto, per natura, si hanno sempre un numero basso di operazioni per ridurre sia i tempi che per ottimizzare i consumi energetica. In certi casi si ha la **ultra small world property** dove la proprietà viene estremizzata e infatti si arriva ad avere che la lunghezza media di un cammino tra due nodi è proporzionale a $\ln(\ln |V|)$, che è molto minore di $\ln(|V|)$. Un esempio di questa tipologia di rete è visibile in figura 3.4¹³

¹²Réka Albert; Scale-free networks in cell biology. J Cell Sci 1 November 2005; 118 (21): 4947–4957. doi: <https://doi.org/10.1242/jcs.02714>

¹³Barabási, AL., Oltvai, Z. Network biology: understanding the cell’s functional organization. Nat Rev Genet 5, 101–113 (2004). <https://doi.org/10.1038/nrg1272>

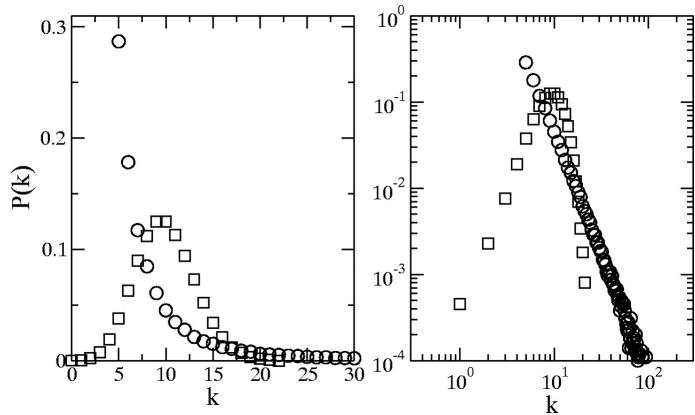


Figura 3.3: Confronto tra la *distribuzione di Poisson*, rappresentata dai quadrati, e la *power-law*, rappresentata coi cerchi, prima in scala normale e poi in scala logaritmica.

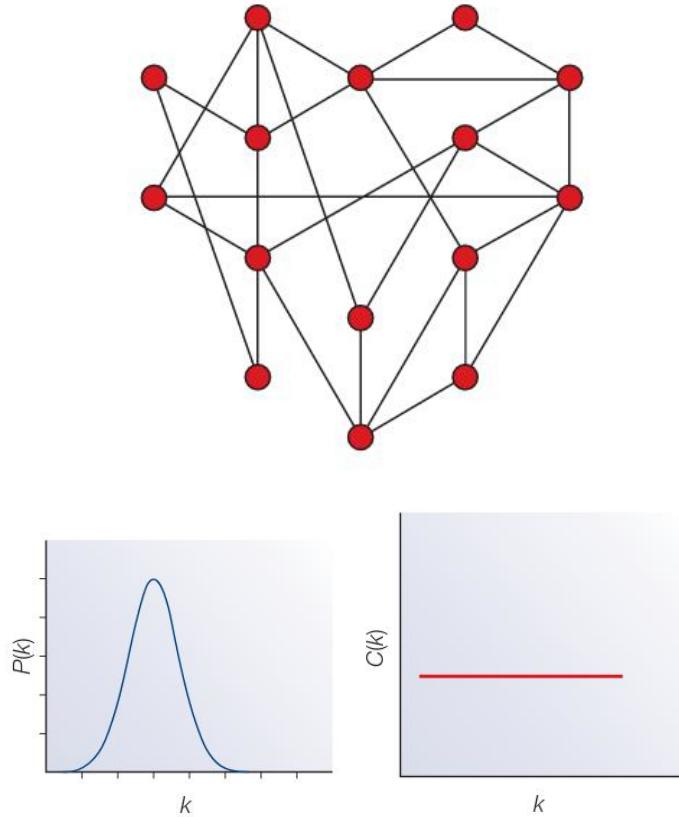


Figura 3.4: Esempio di *random network* con i grafici di $P(k)$ e $C(k)$.

3.3.2 Scale-Free Networks

Si hanno poi le **scale-free network**, dette anche **Barabási-Albert model**, dal nome di coloro che le formalizzarono nel 1999.

Si hanno quindi le seguenti caratteristiche principali per una *scale-free network*:

- la rete è **non omogenea**, da qui il nome scale-free. Si hanno quindi pochi nodi fortemente connessi, gli **hub**, e tanti nodi con pochissimi archi incidenti
- la *degree distribution* segue la **power-law**, come visibile in figura 3.3¹⁴, avendo che:

$$P(k) = k^{-\gamma}, \gamma \in \mathbb{R}$$

dove si ha la *ultra small world property* per $2 < \gamma < 3$, anche se in realtà dei valori di γ che eccedono questo range comportano altre tipologie di rete

- non si ha **modularità intrinseca**, non avendo quindi moduli e avendo che $C(k)$ è indipendente dal valore di k . Si hanno quindi pochi cluster

Nel dettaglio, al variare di γ , dato $\langle d \rangle$ come lunghezza del cammino medio e $k_{\max} \sim N^{\frac{1}{\gamma-1}}$:

- se $\gamma < 2$ si parla di **reti anomale**, avendo $\langle k \rangle$ che diverge e $\langle k^2 \rangle$ che diverge. Si ha in questo caso che $\langle d \rangle$ è circa costante. Non si hanno *large network* per tali valori di γ
- se $\gamma = 2$ non si ha una categorizzazione ma si avrebbe $k_{\max} \sim N$
- se $2 < \gamma < 3$ si parla quindi di **scale-free network**, avendo $\langle k \rangle$ finito e $\langle k^2 \rangle$ che diverge. Si ha in questo caso che $\langle d \rangle \sim \ln(\ln(N))$ parlando di *ultra small world*
- se $\gamma = 3$ si ha un *punto critico* ma si ha $\langle d \rangle \sim \frac{\ln N}{\ln(\ln(N))}$
- se $\gamma > 3$ si parla di reti indistinguibili da una **random network** (come le *reti di citazioni*, *reti di collaborazioni* etc...), avendo $\langle k \rangle$ finito e $\langle k^2 \rangle$ che finito. Si ha in questo caso che $\langle d \rangle \sim \frac{\ln N}{\ln\langle k \rangle}$ parlando di *small world*

¹⁴Réka Albert; Scale-free networks in cell biology. J Cell Sci 1 November 2005; 118 (21): 4947–4957. doi: <https://doi.org/10.1242/jcs.02714>

Questa tipologia di rete è solitamente quella più usata in ambito biologico, infatti possiamo ad esempio pensare ad una proteina come *p53*, tra le principali responsabili dell'*apoptosi*, che in una rete sarà sicuramente un *hub*. Altri esempi possono essere le molecole di *ATP*, *ADP* etc. . . . Alcuni esempi sono (*alcune immagine relative sono presenti sulle slide*):

- *rete PPI per il lievito*
- *rete di proteine per c. elegans, un eucariote*
- *reti metaboliche per a. fulgidus, un archaea, e. coli, un batterio, c. elegans etc. . . ,* dove le reti sono modellate da grafi orientati e si ha che sia la *indegree distribution* che la *outdegree distribution* seguono la *power-law*

Ma si hanno anche esempi non biologici come:

- *world wide web*
- *connessioni tra gli aeroporti mondiali*

Interessante è cercare di capire perché i sistemi biologici tendono a comportare *scale-free network*. Si suppone infatti che tale comportamento abbia due cause principali:

1. la **crescita (growth)**
2. il cosiddetto **attaccamento preferenziale (preferential attachment)**

Infatti questi due processi generano *hub* tramite il processo detto “*rich-gets-richer*” per il quale nuovi nodi tendono a collegarsi a nodi con un grado alto, facendo sì che i nodi con alto grado siano destinati ad avere sempre più nodi incidenti, aumentandone ancora il grado e aumentando la possibilità che nuovi nodi vengano attaccati a loro. Si ha inoltre che è molto probabile che i primi nodi della rete siano quelli destinati ad avere un grado alto che cresce all’aggiunta di nuovi nodi (cosa che porta ad avere reti dominate da *hub*). Questo comportamento è anche alla base dell’*algoritmo di PageRank*. Quindi in generale si hanno varie ipotesi possibili:

- l’*evoluzione*
- *ottimizzazione energetica*
- maggior *robustezza* (concetto che si introdurrà a breve) nei confronti delle perturbazioni

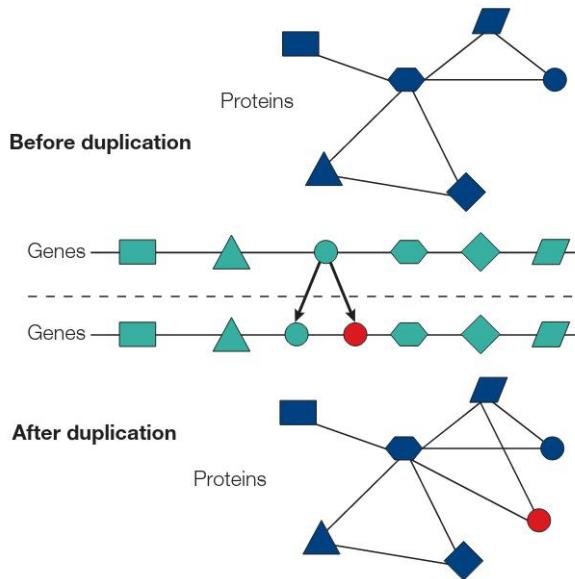


Figura 3.5: Rappresentazione grafica dell’ipotetico processo di *duplicazione genica*, che produce i due nodi rappresentati da cerchi, che porterebbe a *scale-free network*.

Un’altra teoria molto accreditata è quella, parlando di *reti PPI*, rileva le origini di tali reti nella **duplicazione genica**, spiegata visivamente in figura 3.5¹⁵. Quando le cellule si dividono, uno o più geni potrebbero essere copiati due volte nel genoma della prole e ciò induce la crescita nella *rete PPI* poiché esiste un gene in più che codifica per una nuova proteina, avendo letteralmente un nodo in più uguale ad un altro. La nuova proteina ha la stessa struttura della vecchia, quindi entrambe interagiscono con le stesse proteine e le proteine che hanno interagito con la proteina duplicata originale acquisiranno ciascuna una nuova interazione con la nuova proteina. Inoltre le proteine con un gran numero di interazioni tendono a ottenere collegamenti più spesso, poiché è più probabile che interagiscano con la proteina che è stata duplicata. Si creano così *hub* e *scale-free network*. Ovviamente poi, nella realtà, tali proteine duplicate è difficile che restino esattamente uguali e in ogni caso non è un discorso semplice pensare di rimuovere a priori tali nodi dalla rete.

Tali modelli trovano molto spazio anche fuori dal mondo biologico, basti pensare a reti per modellare materiali etc...

¹⁵Barabási, AL., Oltvai, Z. Network biology: understanding the cell’s functional organization. Nat Rev Genet 5, 101–113 (2004). <https://doi.org/10.1038/nrg1272>

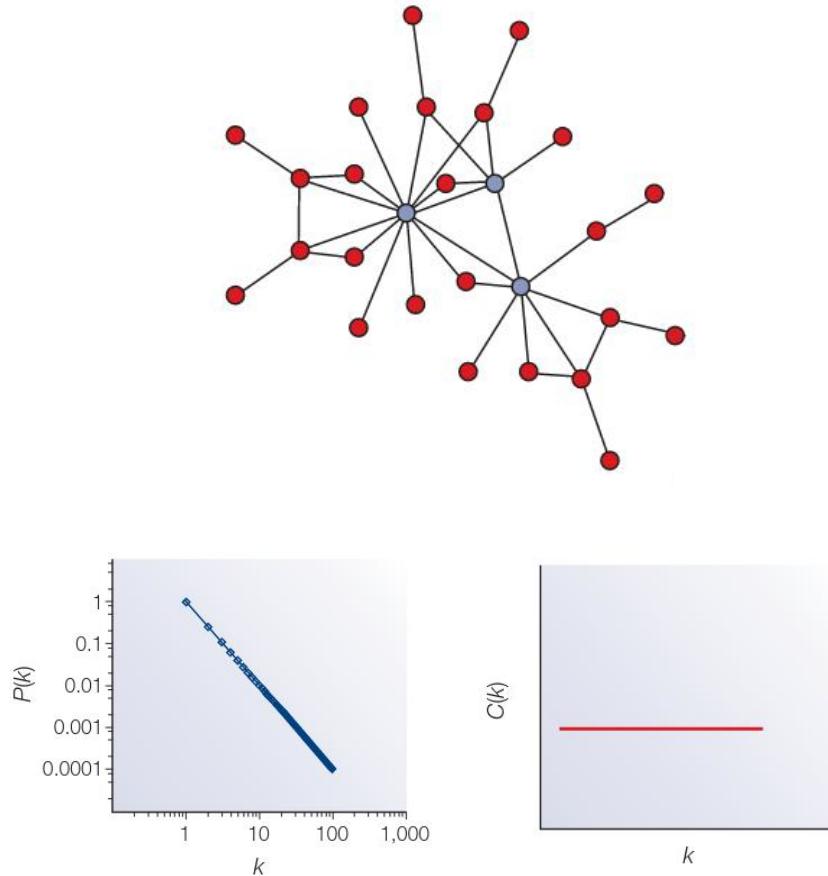


Figura 3.6: Esempio di *scale-free network* con i grafici di $P(k)$ e $C(k)$. Gli *hub* sono colorati in grigio.

Un esempio di questa tipologia di rete è visibile in figura 3.6¹⁶.

Robustezza Topologica

Un altro discorso interessante da introdurre in questo contesto è quello della “resistenza” da parte delle *scale-free* network alle *perturbazioni*. Si parla quindi di **robustezza topologica** delle reti.

Definizione 33. Si definisce **robustezza** come l’abilità del sistema di rispondere a cambiamenti nelle condizioni esterne o nell’organizzazione interna, mantenendo un comportamento “normale”.

¹⁶Barabási, AL., Oltvai, Z. Network biology: understanding the cell’s functional organization. Nat Rev Genet 5, 101–113 (2004). <https://doi.org/10.1038/nrg1272>

Una formalizzazione matematica di un “punteggio” relativo alla robustezza non è discorso banale e nemmeno unico. Si possono avere varie strategie basate sul numero di *hub*, sui pesi degli archi, sulle caratteristiche globali della rete etc...

Ci si chiede quindi:

- cosa succede se si disabilita/elimina un numero sostanziale di nodi in una rete?
- cosa succede se si verifica un errore accidentale?
- cosa succede se si rimuovono deliberatamente nodi specifici nella rete?

Parlando di rimozione di nodi potrei avere infatti due casi:

1. un **random attack**, dove letteralmente il nodo da eliminare è uno casuale e, avendo molti più nodi con basso grado che *hub* (che sono pochissimi), si ha che la rete “collassa” ad un grafo sconnesso molto lentamente in quanto gli *hub* impiegano diverse rimozioni di nodi a sparire
2. un **deliberate attack** sugli *hub*, dove appunto si mira ad eliminare specificamente i nodi *hub*. In questo caso la rete “collassa” ad un grafo sconnesso molto in fretta

Le chance di poter studiare *proprietà emergenti* da una rete è direttamente correlata alla “resistenza” ai vari tipi di attacchi anche se bisogna ricordare che non sempre avere *hub* è a priori la situazione migliore per uno studio.

3.3.3 Hierarchical Networks

Vediamo infine l’ultima tipologia di rete trattata in questo corso, le **hierarchical networks**. Tali reti hanno comunque poco spazio nella *systems biology*. Si hanno quindi le seguenti caratteristiche principali:

- si ha la *coesistenza* di **modularità**, **clustering locale** e **topologia scale-free**. Si hanno quindi cluster, poco collegati tra loro, con all’interno *modules*
- la *degree distribution* segue la **power-law**, come visibile in figura 3.3¹⁷

¹⁷Réka Albert; Scale-free networks in cell biology. J Cell Sci 1 November 2005; 118 (21): 4947–4957. doi: <https://doi.org/10.1242/jcs.02714>

- si ha **modularità intrinseca**, avendo che $C(k)$ è proporzionale a $\frac{1}{k}$

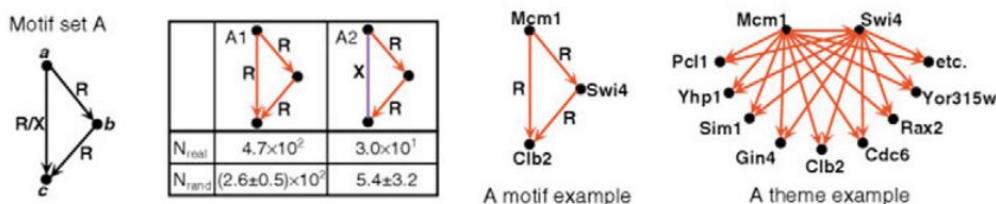
Prove crescenti suggeriscono che le reti biologiche contengono piccoli sottografi conservati dai processi evolutivi che hanno una struttura ben definita. Possiamo quindi caratterizzarli, anche se è già stato fatto, in modo descrittivo in quanto non esiste una vera e propria formalizzazione matematica di essi:

- i **module**, ovvero un gruppo di nodi collegati fisicamente o funzionalmente che lavorano insieme per ottenere una specifica funzione cellulare, come ad esempio la trasduzione del segnale di una determinata molecola
- i **motif**, ovvero sottografi che si verificano significativamente più frequentemente nella rete data di quanto si otterebbe con una *random network*

Questi due tipi di sottografo sono essenziali negli studi in *systems biology* tramite reti ma la loro identificazione è un problema computazionalmente complesso, oltre ad essere, come detto, non ben definito.

Parlando di *motif* se ne possono identificare varie tipologie, come quelle proposte da Albert¹⁸:

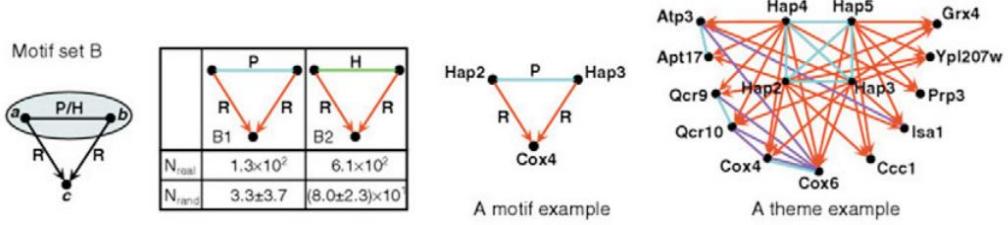
- **transcriptional feed-forward loop**, ad esempio:



dove con *R* si identificano gli archi per le regolazioni trascrizionali e con *X* le *espressioni correlate*

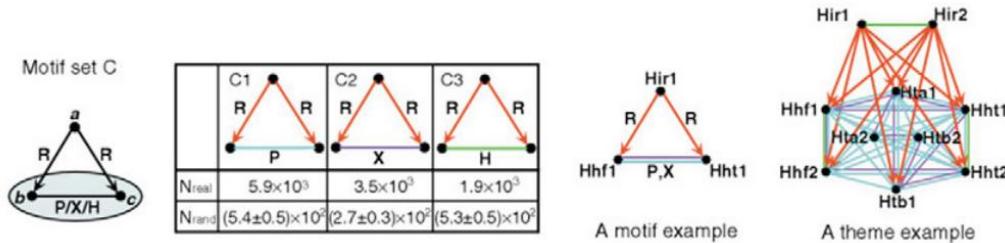
- **transcriptional co-regulation**, ad esempio:

¹⁸Réka Albert; Scale-free networks in cell biology. J Cell Sci 1 November 2005; 118 (21): 4947–4957. doi: <https://doi.org/10.1242/jcs.02714>



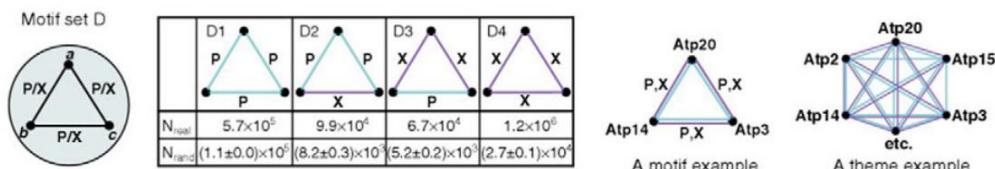
dove con R si identificano gli archi per le *regolazioni trascrizionali*, con P le *interazioni proteiche* e con H l'*omologia tra sequenze*

- co-regulation of members of a protein complex, ad esempio:



dove con R si identificano gli archi per le *regolazioni trascrizionali*, con P le *interazioni proteiche*, con H l'*omologia tra sequenze* e con X le *espressioni correlate*

- co-expressed protein cliques, ad esempio:



dove P le *interazioni proteiche* con X le *espressioni correlate*

Un esempio di questa tipologia di rete è visibile in figura 3.7¹⁹. Un testo online interessante sul tema delle reti si trova al link:

<http://networksciencebook.com>

¹⁹Barabási, AL., Oltvai, Z. Network biology: understanding the cell's functional organization. Nat Rev Genet 5, 101–113 (2004). <https://doi.org/10.1038/nrg1272>

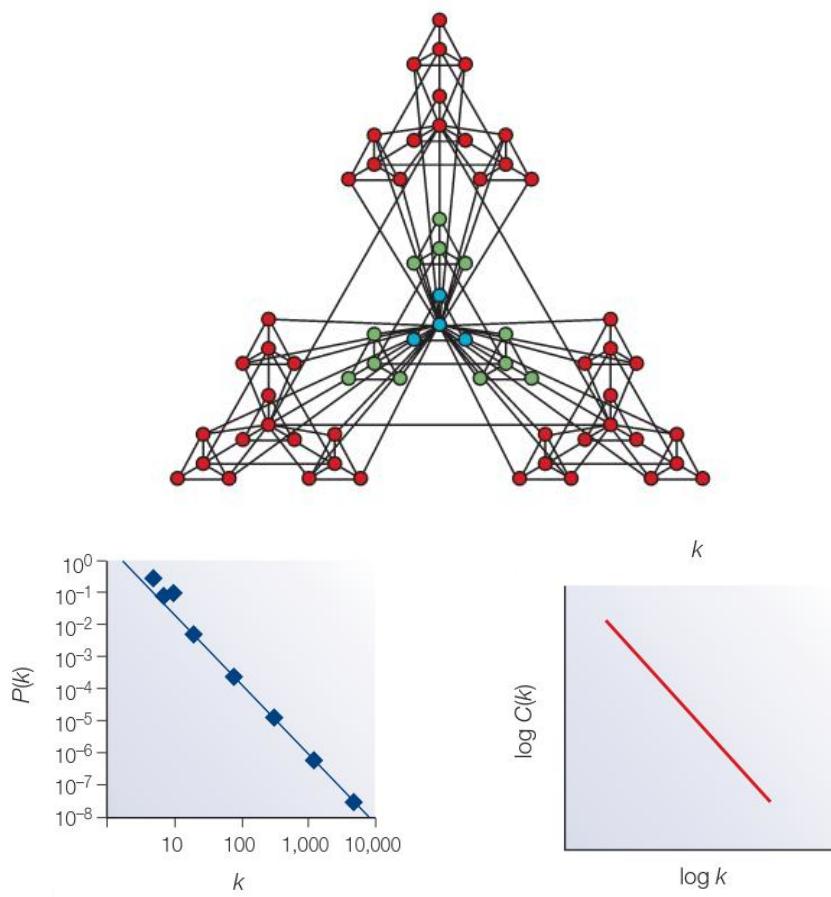


Figura 3.7: Esempio di *hierarchical network* con i grafici di $P(k)$ e $C(k)$. Gli *hub* sono colorati in grigio.

3.4 Software

Per analizzare reti, biologiche ma anche non biologiche (si passa anche a studi di scienze sociali e reti complesse generiche), uno dei tool standard in uso è **Cytoscape**²⁰, disponibile gratuitamente a <http://www.cytoscape.org/>. Citando direttamente:

“Cytoscape is an open source software platform for visualizing molecular interaction networks and biological pathways and integrating these networks with annotations, gene expression profiles and other state data.”

“Although Cytoscape was originally designed for biological research, now it is a general platform for complex network analysis and visualization.”

“Cytoscape core distribution provides a basic set of features for data integration, analysis, and visualization. Additional features are available as Apps (formerly called Plugins). Apps are available for network and molecular profiling analyses, new layouts, additional file format support, scripting, and connection with databases.”

Come scritto si hanno a disposizione una vasta serie di plugins, detti *apps* e ben introdotti dal paper di Saito et al.²¹, che aggiungono moltissime funzionalità, tra cui, ad esempio, collegare una rete ad un database esterno per fare *gene enrichment*.

²⁰Cline, Melissa S et al. “Integration of biological networks and gene expression data using Cytoscape.” Nature protocols vol. 2,10 (2007): 2366-82. doi:10.1038/nprot.2007.324

²¹Saito R, Smoot ME, Ono K, et al. A travel guide to Cytoscape plugins. Nat Methods. 2012;9(11):1069-1076. doi:10.1038/nmeth.2212

Capitolo 4

Logic-Based Modelling

Si introducono ora i **modelli logic-based**.

Ricordiamo che tali modelli, come visibile in figura 2.6:

- sono sistemi meno *large-scale* di quanto lo fossero i *modelli interaction-based*
- presentano tendenzialmente un basso costo computazionale
- presentano un livello di dettaglio, per quanto ambiguo nello schema, variabile
- presentano alcune difficoltà nella misurazione dei dati

Il tutto comporta una miglior capacità predittiva rispetto ai *modelli interaction-based* e, grazie all'uso di vari tipi di logiche, può portare anche a ottenere modelli molto più predittivi.

Un'altra differenza sostanziale rispetto ai *modelli interaction-based* è che si può iniziare a parlare di *simulazioni*, superando il “limite” delle sole analisi. In questo caso i modelli più semplici si basano sulla *logica booleana* e, come si vedrà, se simulazioni e le analisi consistono nel determinare:

- **cicli**, ovvero sequenze ripetute di stati di sistema
- **attrattori**, ovvero stati finali raggiungibili da un qualsiasi stato iniziale
- **bacini di attrazione**, percorsi di stati intermedi che iniziano da uno stato iniziale e terminano in un attrattore

Con l'aggiunta della **logica fuzzy**, più complessa di quella booleana, si può anche derivare il *comportamento dinamico* del sistema, ad esempio derivando

la variazione temporale del valore di ogni componente in uno o più stati, magari dopo una certa *perturbazione*. Si parla quindi di **modelli qualitativi** e **dinamici**. L'informazione *quantitativa* è ridotta al minimo, non avendo la rappresentazione di vere e proprie interazioni/proprietà chimiche e fisiche come, ad esempio, la rappresentazione di *kinetic-rate* etc...

Come detto si hanno sia *sistemi small-scale* che *sistemi large-scale*, anche se comunque di dimensioni ridotte rispetto ai *modelli interaction-based*, e ad esempio sono usati per:

- **reti di regolazione gene-gene**
- **pathway per il segnale di trasduzione**
- **differenziazione cellulare**, soprattutto grazie allo studio degli *attrattori*
- **pathway per la morte programmata cellulare**

Tecnologie sperimentali di natura qualitativa (ad esempio *targeting genico* e *screening fenotipici*) hanno portato allo sviluppo di metodi computazionali per modellare e analizzare reti regolatorie geniche sulla base di regole logiche. L'architettura di tali modelli, per semplificare, consiste in due “aspetti” principali:

1. la **struttura della rete**, modellata tramite *grafo diretto*, dove con la direzione si specificano principalmente, ma non solo, fenomeni di regolazione
2. le **dinamiche della rete**, modellati tramite stati logic-based mutevoli e funzioni di update degli stati stessi, che garantiscono un'evoluzione nel tempo

L'idea generale è quindi quella di partire da uno **stato iniziale** per poi assegnare un nuovo valore ad ogni variabile del modello, quindi ad ogni nodo della rete, tramite l'uso di *funzioni logiche* che combinano i valori delle variabili dello stato corrente per produrre il nuovo stato.

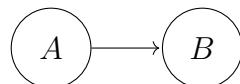
Interessante è elencare fin da subito alcuni **pro** di questo approccio modellistico:

- i *modelli logic-based* sono **versatili**, in quanto una variabile può praticamente rappresentare qualsiasi cosa, come ad esempio un gene, un'attività genica, la presenza di una proteina, un fenotipo, lo stato di una cellula etc... Inoltre si possono mischiare componenti eterogenee in modo abbastanza semplice, a differenza di

quanto accadeva coi *modelli interaction-based*, dove era molto più complesso fare ciò

- i *modelli logic-based* sono **flessibili**, in quanto lo stato di un dato componente cellulare può essere rappresentato da una o più variabili, con diversi insiemi di valori. Quindi una certa componente può avere funzioni diverse a seconda dello stato del sistema e questo risolve quanto visto nel caso dei *modelli interaction-based* in merito ai nodi ripetuti nella rete, avendo che in questo caso i due o più nodi sono ben distinti, magari avendo un nodo per una certa proteina e un altro per la stessa proteina ma fosforilata (si ricorda che la *fosforilazione* è una reazione chimica, fondamentale in biochimica, che consiste nell'addizione di un gruppo fosfato, PO_4^{3-} , ad una proteina o ad un'altra molecola e si ricorda che gli enzimi che solitamente catalizzano le fosforilazioni sono le chinasi)
- gli effetti delle *perturbazioni*, come *inibizioni* o *mutazioni*, possono essere “testati” in modo molto facile e diretto

Si hanno però anche vari **contro**, che principalmente si riconducono al fatto che non sono modelli meccanicistici. Infatti, per quanto si abbia alla base un *grafo diretto*, non è possibile inferire la proprietà meccanicistica che si ha dietro la regolazione positiva/negativa, o altro, che viene rappresentata mediante l'arco diretto. Ipotizziamo anche solo di avere due nodi *A* e *B*, con *A* che regola *B*:



Potremmo dire che “l'attività di *A* stimola l'attività di *B*” e si può modellare una regolazione positiva o negativa ma non potremo mai caratterizzare nel dettaglio il meccanismo stesso, che potrebbe essere, ad esempio:

- l'attivazione della produzione di *B*
- l'inibizione della degradazione di *B*
- la stabilizzazione dell'*high-activity state* di *B*
- ...

Posso solo sapere che c'è uno stimolo, avendo infatti perdita d'informazione che impedisce l'inferenza dei meccanismi.

Nei paragrafi precedenti si sono nominate spesso le *variabili*. Approfondendo il discorso si ha che in modello matematico esse possono essere:

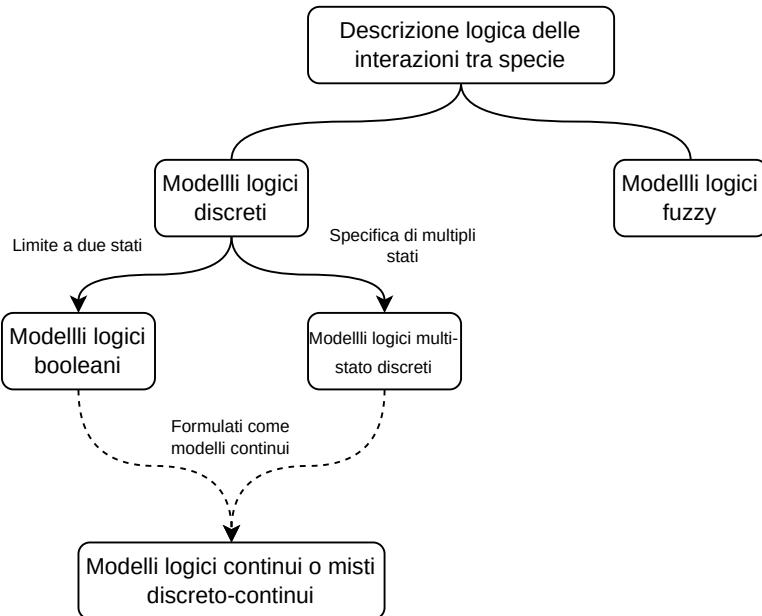


Figura 4.1: Schema riassuntivo dell'uso delle variabili in modelli *logic-based*.

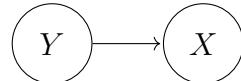
- **valori booleani**, ovvero valori binari come 0/1, presente/assente, attivo/inattivo etc...
- **valori multi-stato**, sia **linguistici** che **numerici**, come ad esempio nullo/basso/medio/alto, 0/1/2/3 etc...
- **valori numerici interi o reali**, usati ad esempio per rappresentare la concentrazione o il numero di molecole

Le prime due categorie sono quindi sono a *valori discreti* mentre la terza è a *valori continui*. Una categorizzazione più precisa, parlando di *modelli-logic based* si può osservare in figura 4.1¹.

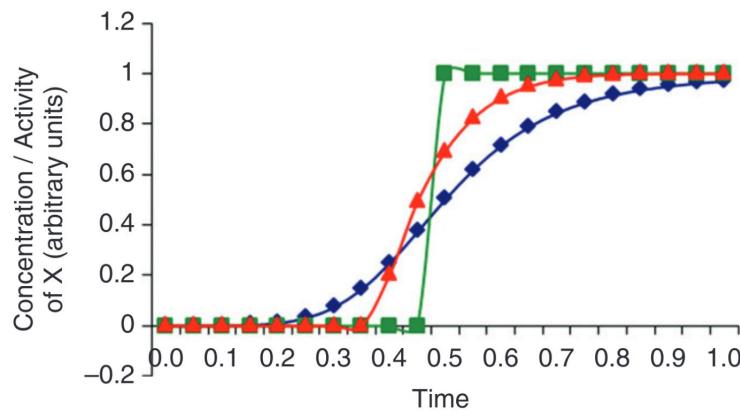
È stato anche citato il cosiddetto *stato iniziale del sistema*, ovvero la *condizione iniziale* del modello, che prevede l'assegnamento di un valore, all'inizio della simulazione, ad ogni variabile del sistema. La *condizione iniziale* influenza ovviamente i risultati stessi della simulazione, determinando come evolve il sistema nel tempo, soprattutto se si ha una simulazione deterministica.

¹Morris MK, Saez-Rodriguez J, Sorger PK, Lauffenburger DA. Logic-based models for the analysis of cell signaling networks. Biochemistry. 2010;49(15):3216-3224. doi:10.1021/bi902202q

Ovviamente la scelta tra discreto e continuo comporta delle conseguenze. Prendiamo ad esempio la seguente situazione:



e il seguente grafico², che descrive la regolazione positiva di Y su X :



Nel grafico si hanno:

- sull’asse delle ascisse il tempo
- sull’asse delle ordinate la concentrazione del nodo X (si assume che la concentrazione/attività del nodo Y cresca linearmente)
- la funzione booleana in verde
- due funzioni continue in rosso e blu che rappresentano il “caso reale”

Si nota come il “cambio” per la funzione booleana sia “secco”, prima 0 e poi 1, un certo momento temporale. Questa eccessiva semplificazione non sempre è accettabile per descrivere casi reali ma ci sono situazioni in cui è comunque accettabile. Potrei avere anche altre funzioni, magari multi-stato, come quelle in figura 4.2³ Si può quindi rilevare un ulteriore **contro** che si può

²Albert, Reka, and Juilee Thakar. "Boolean modeling: a logic-based dynamic approach for understanding signaling and regulatory networks and for making useful predictions." Wiley Interdisciplinary Reviews: Systems Biology and Medicine 6.5 (2014): 353-369.

³Morris MK, Saez-Rodriguez J, Sorger PK, Lauffenburger DA. Logic-based models for the analysis of cell signaling networks. Biochemistry. 2010;49(15):3216-3224. doi:10.1021/bi902202q

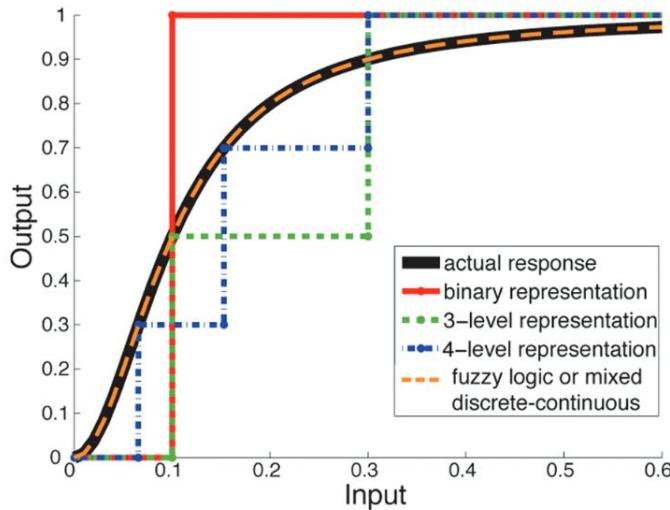


Figura 4.2: Esempio di approssimazioni di un certo andamento, partendo dall'approssimazione booleana a quella nel caso continuo, che rappresenta correttamente la relazione a sigmoide tra i livelli di input e output ad esempio nel casi dell'azione della chinasi su un substrato, passando per diverse approssimazioni multi-strato.

avere in *sistemi logic-based* in quanto quando un nodo è “off”, ad esempio, non significa esattamente che la molecola ha zero concentrazione in quel momento nel sistema. Si sottintende una *soglia implicita* che stabilisce “on” e “off”, quindi è “off” se la molecola non è presente in modo sufficiente al punto da permettere cambiamenti nelle molecole regolate da essa. Quando supera quella soglia diventa “on” (ad esempio si può pensare di avere $0 \leq x \leq 1$ e che fino a $x = 0.8$ si ha lo stato “off” e con $x > 0.8$ lo stato “on”). Si ha quindi solo un’approssimazione qualitativa della regolazione molecolare anche se bisogna notare che noti che molti dati sperimentali disponibili sulle regolazioni molecolari sono di natura qualitativa.

Si possono, come anticipato, fare delle *simulazioni* e l’aggiornamento degli stati può essere determinato in vari modi, tramite diverse concezioni del *tempo*:

- tramite **iterazioni** che non rappresentano necessariamente una durata temporale fisica specifica, infatti un “passo temporale”, un *time step* può rappresentare una durata diversa per iterazioni diverse (magari in un caso è un evento che avviene in un secondo e in un altro che avviene in un’ora ma entrambi sono un singolo *time step* di ugual durata). Si hanno inoltre due ulteriori sotto

casistiche:

1. la **modalità di aggiornamento sincrona**, dove il valore di tutte le variabili viene ricalcolato dopo ogni singola iterazione. *Questa sarà la modalità che verrà approfondita nel corso*
2. la **modalità di aggiornamento asincrona**, dove le variabili subiscono transizioni una alla volta e le variabili da aggiornare vengono scelte in modo tendenzialmente causale
 - tramite **step discreti**, che quindi hanno una specifica durata eventualmente diversa l'uno dagli altri, dopo i quali avvengono gli aggiornamenti delle variabili
 - tramite **tempo continuo**, dove si simula (eventualmente su una scala diversa) il trascorrere reale del tempo

Un'altra problematica che si presenta dopo aver capito come viene aggiornato lo stato del sistema in un *modello logic-based* è quello che il numero di stati cresce in modo esponenziale rispetto al numero di variabili, quindi al numero di nodi $|V|$. Questo aspetto rende complicata anche la validazione dei modelli, validazioni che solitamente vengono fatte in modo probabilistico (*aspetto non ben chiarito a lezione ma magari si vedrà più avanti nel corso*). Inoltre, avendo nei *modelli logic-based* solitamente l'uso di iterazioni per descrivere l'evoluzione temporale si ha difficoltà a tracciare processi lenti/veloci o anche *ritardi*, spesso frequenti nei sistemi biologici.

Come detto ad ogni iterazione si ha una **transizione di stati**, quindi l'*aggiornamento degli stati*, che avviene dopo la computazione di un insieme di *funzioni logiche* assegnate ad ogni variabile del modello, quindi ad ogni nodo del grafo nel nostro caso. Si hanno quindi:

- gli **operatori booleani**, nel caso specifico un sottoinsieme degli stessi composto da \wedge (l'*and logico*), \vee (l'*or logico*) e il \neg (il *not logico*)
- le **espressioni booleane** ottenute tramite gli *operatori booleani*

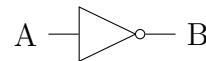
Dopo che si è definita la logica di un modello e si è costruito il modello stesso si possono produrre le cosiddette **traiettorie**, dette anche **pseudo time-courses**, quindi sequenze di stati raggiungibili consecutivamente ognuno a partire dal precedente, e studiare eventuali *attrattori* etc... Ovviamente, poiché il tempo non è correlato al tempo fisiologico/reale, i modelli booleani possono fornire solo una cronologia qualitativa delle attivazioni dei nodi.

4.1 Introduzione alla Logica Booleana

Prima di procedere occorre fare un piccolo ripasso di logica booleana, anche per poterla collegare a situazioni biologiche.

Come anticipato abbiamo tre *operatori logici*:

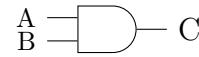
1. il *not*, indicato solitamente con \neg , è un operatore unario che viene rappresentato, a livello circuitale tramite:



e al quale corrisponde la seguente tabella di verità:

A	$B = \neg A$
0	1
1	0

2. l'*and*, indicato solitamente con \wedge , è un operatore binario che viene rappresentato, a livello circuitale tramite:



e al quale corrisponde la seguente tabella di verità:

A	B	$C = A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

3. l'*or*, indicato solitamente con \vee , è un operatore binario che viene rappresentato, a livello circuitale tramite:



e al quale corrisponde la seguente tabella di verità:

A	B	$C = A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Un'espressione booleana è appunto la combinazione degli *operatori booleani* e delle *variabili booleane*. Uno dei problemi è che, date n variabili booleane (quindi $|v| = n$ nodi), si hanno esattamente 2^n possibili combinazioni di valori di stato, avendo 2^n righe nella tavola di verità.

Un esempio potrebbe essere la seguente espressione booleana:

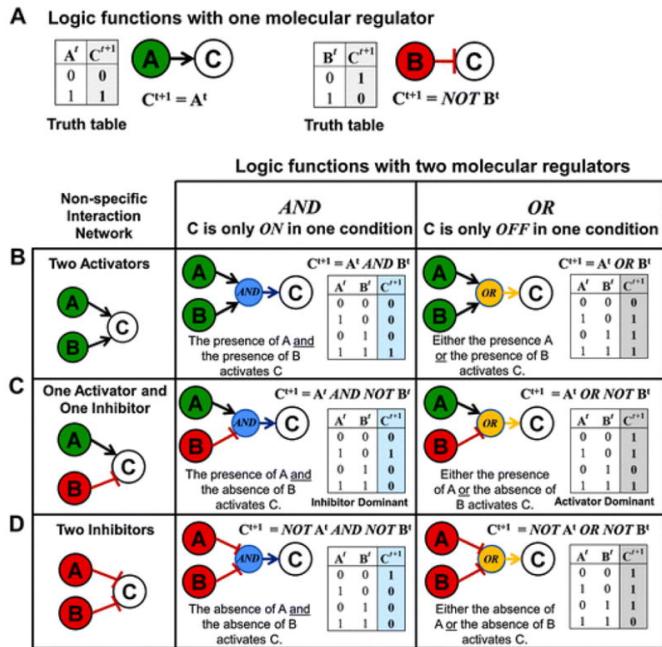
$$C = (A \wedge B) \vee (\neg B)$$

alla quale corrisponde la seguente tabella di verità:

A	B	$A \wedge B$	$\neg B$	C
0	0	0	1	1
0	1	0	0	0
1	0	0	1	1
1	1	1	0	1

4.2 Simulazioni su Modelli Logic-Based

Si riprende per comodità un esempio già visto nell'introduzione ai modelli per avere un esempio di quanto detto rapportato alla regolazione molecolare⁴:



⁴Wynn ML, Consul N, Merajver SD, Schnell S. Logic-based models in systems biology: a predictive and parameter-free network analysis method. Integr Biol (Camb). 2012;4(11):1323-1337. doi:10.1039/c2ib20193c

In questo esempio notiamo come possiamo differenziare regolazioni positive, come quella di A su C , e quelle negative, come quella di B su C , anche a livello grafico. Notiamo anche come il formalismo sia del tipo:

$$C^{t+1} = \neg B^t$$

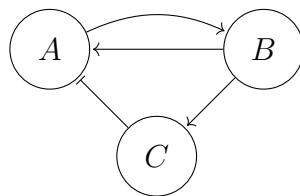
indicando con gli apici lo step temporale, avendo quindi specificato che lo stato risultante in C al tempo $t + 1$ dipende da quello di B al tempo t . Ovviamente questi esempi, con le rispettive tabelle diversità, sono minuscoli, infinitamente più piccoli rispetto a modelli logici reali. Nell'esempio possiamo comunque vedere le varie casistiche dove A attiva C , facendo regolazione positiva, mentre B lo inibisce, facendo regolazione negativa, vedendo i vari risultati possibili nelle varie possibili combinazioni di “input”.

Nel dettaglio, inoltre, lo stato del sistema al tempo t corrisponde ad un vettore booleano consistente nel valore di ogni variabile booleana al tempo t . Purtroppo a volte non abbiamo informazioni biologiche per assegnare la funzione booleana al nodo e ovviamente la questione si complica all'aumentare dei nodi regolatori, positivamente e negativamente, del nodo in questione. Quindi alla semplicità matematica si associa in questo caso anche un “lack” di informazioni biologiche preliminari.

Si procede quindi dallo *stato iniziale*, che viene definito per $t = 0$, e si calcola la traiettoria che si ha a partire da quello stato, che viene composta quindi dall'insieme degli stati a $t = 1, t = 2, t = 3$ etc... dove il tempo è una **variabile discreta**, avendo quindi $\{t, t + 1, t + 2, \dots\}$, e dove si assume aggiornamenti in *modalità sincrona*. Fatte queste premesse è ovvio che lo stato corrente del sistema è identificato univocamente dallo stato precedente, che a sua volta identificava univocamente lo stato corrente come suo successore. Si ha quindi che i *modelli logic-based booleani sincroni* sono **deterministici**, avendo che un certo input produrrà sempre e solo lo stesso output.

Sfruttiamo ora un esempio per caratterizzare meglio *attrattori* e *bacini di attrazione*.

Sia data la seguente rete:



Alla quale vengono aggiunte le seguenti funzioni booleane:

- $f(A) = B \wedge (\neg C)$
- $f(B) = A$
- $f(C) = B$

Si ha quindi la seguente tabella di verità, che avendo 3 variabili/nodi avrà $2^3 = 8$ possibili combinazioni di stati di nodi, avendo che ogni riga della tabella di verità rappresenta uno stato della rete:

A	B	C	$f(A)$	$f(B)$	$f(C)$
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	0	1	0
1	0	1	0	1	0
1	1	0	1	1	1
1	1	1	0	1	1

Possiamo quindi distinguere uno *stato iniziale* che comporta un **ciclo** o un **punto fisso**. Infatti, ad esempio, se parto da $[1, 0, 0]$ avrò la seguente traiettoria:

$$[1, 0, 0] \Rightarrow [0, 1, 0] \Rightarrow [1, 0, 1] \Rightarrow [0, 1, 0] \Rightarrow [1, 0, 1] \Rightarrow \dots$$

Avendo quindi che si ha un **ciclo** tra gli stati $[0, 1, 0]$ e $[1, 0, 1]$, che funge da *attrattore*.

D'altro canto se si seleziona come *stato iniziale* $[1, 1, 0]$ avrò la seguente traiettoria:

$$[1, 1, 0] \Rightarrow [1, 1, 1] \Rightarrow [0, 1, 1] \Rightarrow [0, 0, 1] \Rightarrow [0, 0, 0] \Rightarrow [0, 0, 0] \Rightarrow \dots$$

raggiungendo quindi un **punto fisso**, un *attrattore*, ovvero lo stato $[0, 0, 0]$. Nel dettaglio inoltre, in questo caso semplice e fortuito, il **bacino di attrazione** per l'attrattore $[0, 0, 0]$ è formato da tutte le traiettorie che partono dagli stati presenti nella *traiettoria* che parte dallo stato $[1, 1, 0]$, ovvero dagli stati dell'insieme (**capire se attrattore è nel suo stesso bacino di attrazione**):

$$\{[1, 1, 0], [1, 1, 1], [0, 1, 1], [0, 0, 1], [0, 0, 0]\}$$

Data quindi una *rete booleana* e una traiettoria sufficientemente lunga prima o poi lo stato della rete sarà una ripetizione di una sequenza di stati già incontrata, questo perché si ha un numero di stati totali **finito**, in quanto **discreto**.

Visto l'esempio possiamo raffinare quindi quanto già definito:

- un **attrattore** è definito come **punto fisso (fixed state)** qualora si abbia che un singolo stato appare ripetutamente in una *traiettoria*
- un **attrattore** è definito come **ciclo** se un insieme di stati appare più di una volta in una *traiettoria*

Avendo quindi che un **attrattore** rappresenta uno stato dal quale non è possibile scappare a meno che non si verifichi una *perturbazione* esterna al sistema stesso mentre un **bacino di attrazione** è l'insieme di tutte le traiettorie che terminano in un *attrattore*.

4.2.1 Vari Esempi

Si elencano ora una serie di esempi, tratti da vari paper, per capire meglio la modellazione tramite *modelli logici booleani*.

Un primo esempio è un semplice modello con 12 nodi proposto da Wynn et al.⁵. Questo esempio, come visibile in figura 4.3. Nell'esempio abbiamo infatti, in nella figura *A*, una prima modellazione del sistema tramite una *rete d'interazioni*, con l'aggiunta dei due nodi di *input* e *output*. Passando alla figura *B* abbiamo il passaggio alla *rete booleana* con la rappresentazione esplicita degli operatori booleani tramite nodi. Si noti che il nodo *output* non influenza alcun altro nodo. In figura *C* abbiamo invece l'elenco delle funzioni logiche che permettono l'evoluzione della rete. Poi in figura *D* si hanno vari esempi di transizione dal tempo *t* al tempo *t + 1*. Si hanno infatti 7 possibili stati iniziali e la rappresentazione degli stati è fatta in primis tramite una matrice booleana e quindi tramite l'intero, in base decimale, rappresentato in ogni riga della stessa. Infine, in figura *E*, abbiamo una rappresentazione esplicita dall'insieme e degli stati raggiungibili, rappresentati coi "pallini", e degli attrattori. Si noti che quelli nella figura non sono grafi ma una rappresentazione degli stati dove i "pallini" più esterni rappresentano gli stati iniziali. Si hanno poi le varie traiettorie rappresentate tramite i collegamenti

⁵Wynn ML, Consul N, Merajver SD, Schnell S. Logic-based models in systems biology: a predictive and parameter-free network analysis method. Integr Biol (Camb). 2012;4(11):1323-1337. doi:10.1039/c2ib20193c

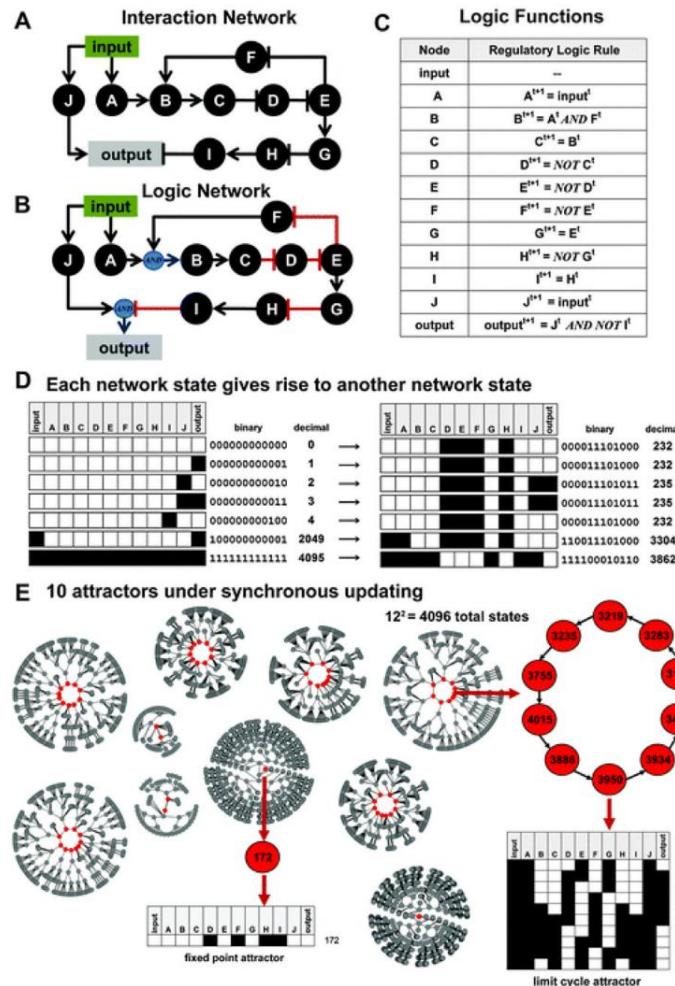


Figura 4.3: Esempio di un *modello logico* tramite *rete booleana*.

tra i vari “pallini”, avendo quindi una rappresentazione dei **bacini di attrazione**, mentre gli **attrattori** sono specificati in rosso, avendo che il singolo “pallino” rappresenta un *fixed point*, mentre se si hanno più “pallini” si ha un *ciclo*, come esplicitato nello zoom a destra. Si comprende ancora meglio come in questo contesto si abbia un vero e proprio concetto di *simulazione*, che porta a ottenere le traiettorie stesse, avendo comunque solo risultati *qualitativi*, per quanto *dinamici*.

Interessante è anche notare come, sempre nel lavoro di Wynn et al.⁶, come

⁶Wynn ML, Consul N, Merajver SD, Schnell S. Logic-based models in systems biology: a predictive and parameter-free network analysis method. Integr Biol (Camb).

visibile in figura 4.4, il variare di una singola funzione logica, nell'esempio *or not* contro *and not*, porti a traiettorie molto diverse con pochissimi "punti di contatto". Non sempre infatti si hanno le basi di conoscenza per poter modellare con certezza un modello, magari non avendo chiara specifica a priori di una certa regolazione. Nel modello, nel dettaglio, si hanno come input, in rosso e verde nella figura:

- sovrappopolazione
- fattori di crescita
- ipossia, ovvero carenza della quantità di ossigeno che raggiunge i tessuti
- danni al *DNA*

In output, come anticipato, si hanno:

- proliferazione
- apoptosis

Il tutto quindi presenta, assumendo *update sincrono*:

- $2^4 = 16$ stati iniziali
- $2^{10} = 1024$ possibili stati della rete
- 16 attrattori

Si nota come questo tipo di situazione sia anche interessante nello studio del variare degli output in base al tuning dei vari input, per portare ad un certo output.

Un altro aspetto fondamentale riguarda l'interpretazione biologica degli attrattori stessi. Tra le varie interpretazioni si hanno quelle dei *fenotipi*, ad esempio nello studio della *differenziazione cellulare*. Si prende ad esempio il paper di Huang⁷, da dove è tratta la figura 4.5. In questa immagine abbiamo la comparazione tra la *rete booleana* e le sue traiettorie con quanto ottenibile tramite varie analisi, ai diversi istanti temporali, con la tecnica dei *microarrays*, che misrano l'espressione genica, e la produzione della *mappa GEDI*. Una **mappa GEDI** (***Gene Expression Dynamics Inspector***) è una rappresentazione visiva di un microarray che riorganizza i geni per creare

2012;4(11):1323-1337. doi:10.1039/c2ib20193c

⁷Huang, Sui. "Reprogramming cell fates: reconciling rarity with robustness." *Bioessays* 31.5 (2009): 546-560.

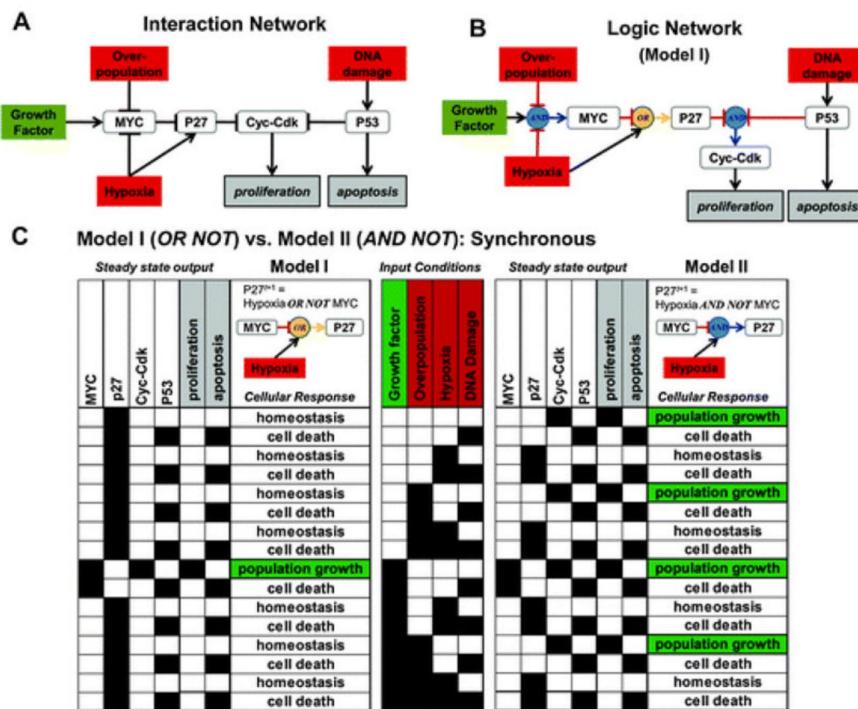


Figura 4.4: Comparazione di due *modelli logici* tramite *rete booleana* per lo studio della *proliferazione* e dell'*apoptosi*, al variare di una singola funzione logica.

pattern caratteristici, che riflettono la somiglianza tra i profili di espressione. In pratica, in modo parallelo al modello logico, si usano altre tecniche per caratterizzare lo stato della rete tramite i vari fenotipi. Da qui, come visibile in figura *E*, posso partire da uno stato cellulare definito a priori e seguire l'evoluzione del modello anche tramite gli studi in laboratorio ai vari step temporali, studiando i fenotipi, che vengono quindi “collegati” al modello booleano, e quindi le caratteristiche emergenti del modello. Quindi, iniziamo con una cellula staminale e durante il processo di differenziazione la cellula cambia e diventa un diverso tipo di cellula. Quindi, lo stato finale è un'attrazione che è la specializzazione della cellula.

Sempre proseguendo sullo stesso discorso sempre Huang⁸ propone anche una rappresentazione grafica degli *ipotetici stati epigenetici*, quindi legato ai fenotipi, i pallini verdi in figura 4.6. I geni portano la cellula verso una certa traiettoria, avendo diverse possibilità di ottenere le diverse espressioni genetiche. Questo visivamente si vede avendo che i “pallini” verdi si vanno a posizionare nelle “conche”. Solo una perturbazione può portare gli stessi fuori da queste conche. Questo si ricollega anche al discorso delle cellule staminali, che restano tali per un certo periodo per poi iniziare a differenziarsi, parlando quindi di *stato quasi-potential* ovvero la cellula ha la potenzialità di cambiare lo stato anche se è potenzialmente stabile. Inoltre in questo contesto il fatto che i sistemi logici siano deterministici può essere limitante nei confronti della stocasticità intrinseca della natura ma, limitandosi al discorso degli *attrattori*, la si può accettare come semplificazione valida.

Per completezza un altro paper interessante è quello di Graf e Enver⁹ dove si ha un discorso analogo a quello fatto.

Vediamo infine un ultimo esempio, tratto dal lavoro di Udyavar et al.¹⁰ dove si è studiata la differenza di fenotipi nel caso del *small cell lung cancer*. Si avevano infatti principalmente due fenotipi, con caratteristiche e quindi terapie associate differenti:

- *neuroendocrin/epithelial, NE*
- *non-neuroendocrin/mesenchymal-like, ML*

⁸Huang, Sui. "Reprogramming cell fates: reconciling rarity with robustness." Bioessays 31.5 (2009): 546-560.

⁹Graf T, Enver T. Forcing cells to change lineages. Nature. 2009;462(7273):587-594. doi:10.1038/nature08533

¹⁰Udyavar AR, Wooten DJ, Hoeksema M, et al. Novel Hybrid Phenotype Revealed in Small Cell Lung Cancer by a Transcription Factor Network Model That Can Explain Tumor Heterogeneity [published correction appears in Cancer Res. 2019 Mar 1;79(5):1014]. Cancer Res. 2017;77(5):1063-1074. doi:10.1158/0008-5472.CAN-16-1467

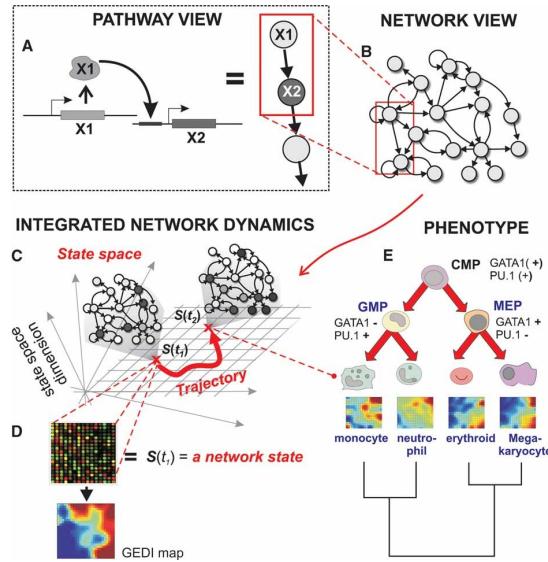


Figura 4.5: Esempio di comparazione tra una *rete booleana* e altre tecniche sperimentali, come le mappe *GEDI*.

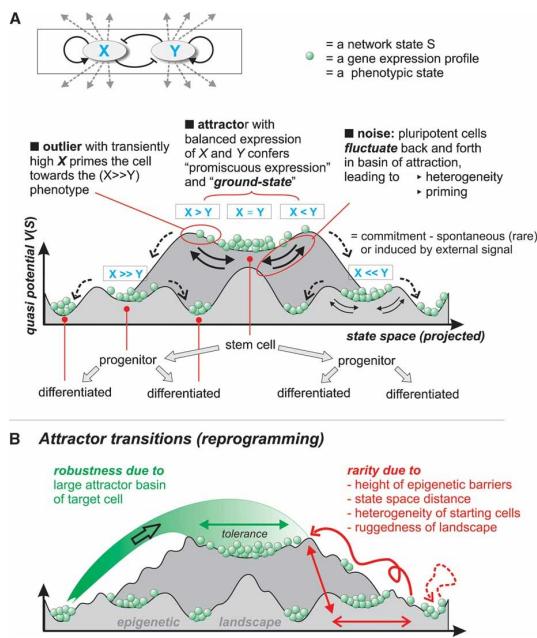


Figura 4.6: Rappresentazione grafica degli *stati epigenetici*.

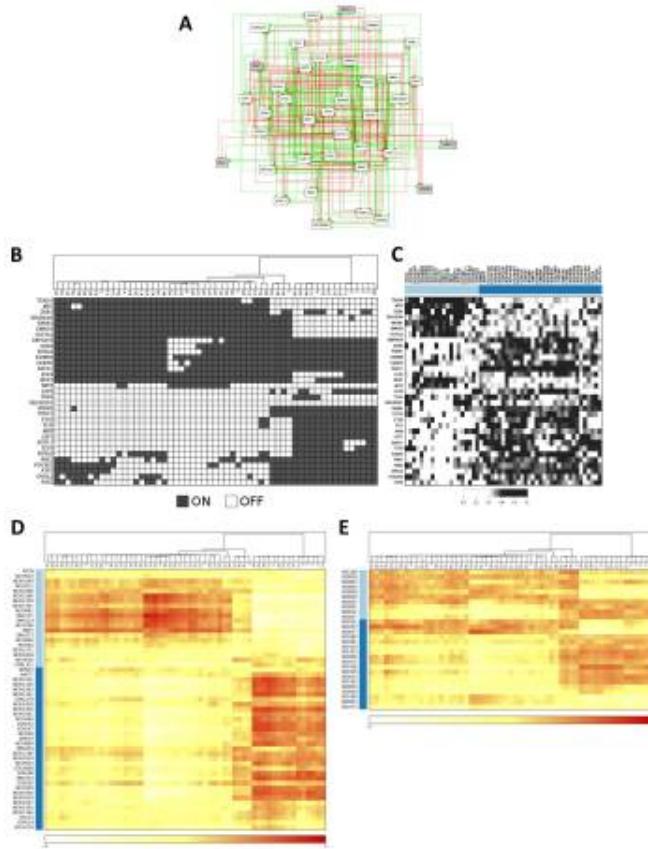


Figura 4.7: Rete booleana, figura *A* e vari grafici, ottenuti computazionalmente nel caso delle figure *B* e *C* e in laboratorio nel caso delle figure *D* ed *E*.

che ci si aspettava fossero ben caratterizzati da due attrattori nel modello, ottenuto tramite 33 fattori di trascrizione, modello che avrebbe poi permesso di regolare meglio i dosaggi delle cure etc....

I risultati dello studio, visibili in figura 4.7, mostrano non solo come i risultati del modello siano comparabili a quelli ottenuti sperimentalmente in laboratorio ma conferma anche l'esistenza di un fenotipo ibrido, un terzo attrattore, conosciuto in letteratura ma non ben caratterizzato, che era quello più aggressivo e difficile da curare. Quindi non solo tali modelli hanno spazio anche in fase di validazione dei dati sperimentali ma possono anche essere l'inizio, come spesso in *systems biology*, per ulteriori studi, con modelli diversi (magari *meccanicistici*) ma anche sperimentali in *wet-lab*, fatti al fine di migliorare la caratterizzazione di quanto scoperto.

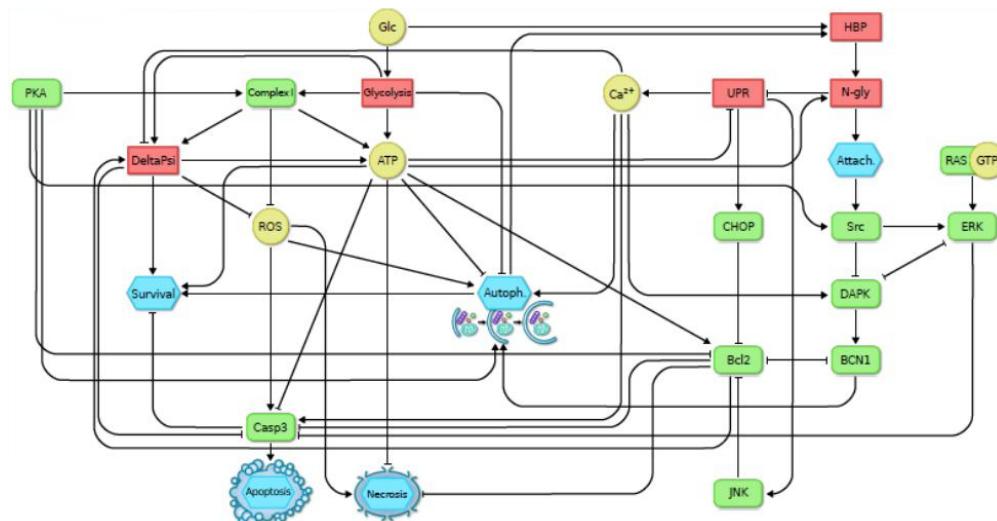
Un altro esempio interessante, qui solo citato, è quello di Offerman et al.¹¹.

4.3 Logica Fuzzy

Sono stati elencati diverse volte i limiti dell'uso della semplice *logica booleana* per la rappresentazione di sistemi biologici quindi il prossimo passaggio è quello di sfruttare come logica la **logica fuzzy**, dove *fuzzy* si può tradurre con il concetto di “sfumatura”.

I sistemi cellulari sono caratterizzati, come già visto, da un alto livello di *eterogeneità*, avendo interazioni orchestrate tra ioni, metaboliti, proteine, complessi macromolecolari, vie di trasduzione del segnale, pathway metabolici, etc... e si sa che lo stato cellulare è normalmente descritto tramite termini qualitativi dai biologi/biotecnologi, che sono soliti usare frasi del tipo “fortemente espresso”, “moderatamente attivo” o simili, evidenziando una forte **incertezza**, ammessa dalla *logica fuzzy*, della descrizione biologica ed evidenziando anche limiti della misura di dati sperimentali.

Vediamo un breve esempio. Si prenda il seguente sistema¹²:



Abbiamo un sistema molto eterogeneo che, avendo magari molecole che regolano processi cellulari o altri regolazioni eterogenee, rappresenta una cellula cancerogena dalla quale si vuole predire il minimo uso di farmaci per massimare la morte cellulare tramite apoptosi. Non potrei in questo caso

¹¹Offermann, Barbara, et al. "Boolean modeling reveals the necessity of transcriptional regulation for bistability in PC12 cell differentiation." *Frontiers in genetics* (2016): 44.

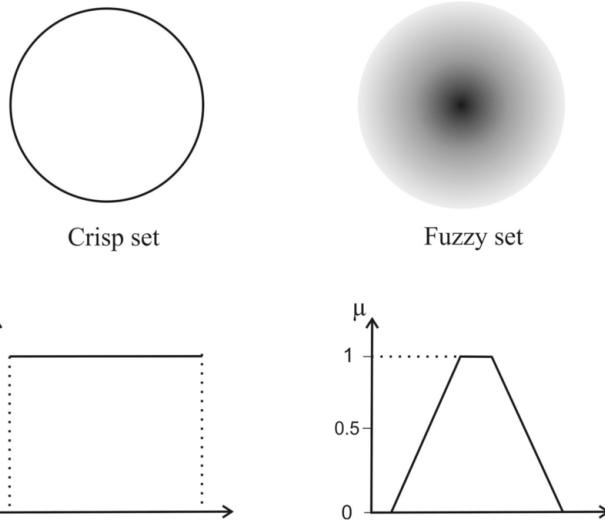
¹²Nobile M.S. et al. Fuzzy modeling and global optimization to predict novel therapeutic targets in cancer cells. *Bioinformatics*, 2019

usare modelli meccanicistici non essendo un sistema *small-scale* e non avendo praticamente informazioni quantitative.

Alla base della *logica fuzzy* si hanno quindi:

- un **grafo diretto** che rappresenta l'insieme dei componenti del sistema, corrispondenti a qualsiasi tipo di biomolecole o interi processi cellulari, nonché i fenotipi di output, e le loro reciproche regolazioni positive o negative
- un insieme di **variabili linguistiche**, coi corrispondenti **termini linguistici**, e le **membership function** (introdotte da Zadeh nel 1965), per i rispettivi **fuzzy set**, che permettono di dare una descrizione qualitativa delle quantità cellulari o dell'attività funzionale. Un altro concetto fondamentale è il **membership degree**, solitamente indicato con μ , che caratterizza il valore attuale rispetto ad una *membership functions*. Tutti questi caratterizzano i nodi della rete.
Le *variabili linguistiche* collegano la descrizione qualitativa alla rappresentazione quantitativa, qualora si abbia (e in assenza si deriva una corrispondenza usando varie tecniche). A differenza dei *modelli meccanicistici* non serve la *parametrizzazione del modello*
- un insieme di **regole logiche fuzzy** che specificano lo stato che assumerà ciascun componente del sistema in base agli stati dei componenti da cui è regolato, permettendo di simulare l'evoluzione del sistema sfruttando algoritmi di ragionamento fuzzy e metodi di “defuzzificazione”, ottenendo appunto un *modello dinamico e quantitativo*, come in natura

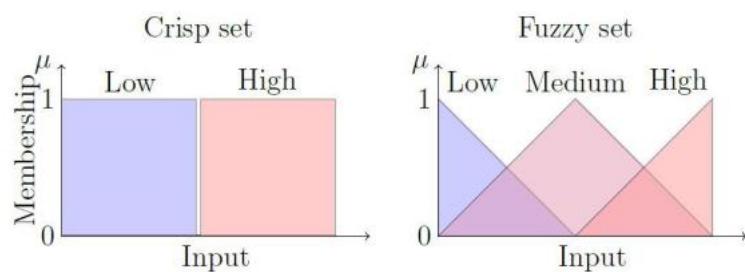
Vediamo ora qualche grafico chiarificatore, dove si comparano *fuzzy set* con *crisp set*, che è un altro nome per descrivere il caso booleano ma usando variabili linguistiche. Vediamo infatti un semplice grafico¹³:



che ci permette di distinguere chiaramente i due casi:

- con il *crisp set* si potrebbe dire che “o è bianco o è nero”, oppure “o si è dentro o si è fuori”, proprio come nel caso booleano 0/1, avendo infatti che posso avere solo o $\mu = 0$ o $\mu = 1$
- con il *fuzzy set* o tutta la casistica intermedia, vendo, per assonanza con l'esempio appena fatto, la “scala di grigi”, con $\mu \in [0, 1]$, avendo che assume valori reali, pur ricordando che non si tratti assolutamente di una probabilità. Inoltre la somma dei vari μ per un certo elemento sull'asse x rispetto a tutti i *fuzzy set* può essere maggiore di 1

Usando i *termini linguistici* potremmo anche avere il seguente esempio:



¹³Toksöz Hozathoğlu, Derya, and İşık Yılmaz. "A Fuzzy Classification Process for Swelling Soils." *Transportation Infrastructure Geotechnology* (2022): 1-14.

Dove possiamo vedere come nel *crisp set* ci sia solo *low/high* per un valore che vive sull'asse delle x mentre per il *fuzzy set* ci siano vari livelli, al variare di μ per i tre *fuzzy set*. Sono proprio i *fuzzy set* a permettere di modellare l'*incertezza* in quanto sfruttano i vantaggi dei *termini linguistici*:

- descrivono la vaghezza dei concetti
- collegano rappresentazioni qualitative e quantitative
- sono vicini al linguaggio naturale

Riprendendo quindi l'esempio:

- *Low*, *Medium* e *High* sono i *termini linguistici*, qualitativi. Potrei avere dei *modificatori linguistici* per aggiungere anche, ad esempio, *very* a *Low* per ottenere *very Low* ma nel caso semplice, che si tratta nel corso, questo sarebbe un altro *fuzzy set* e non qualcosa di ottenibile tramite modificatori
- *Input*, che caratterizza l'asse x è il termine quantitativo (potrei avere una temperatura, l'età etc...) ed è una anch'essa una *variabile linguistica* che rappresenta il mio **universe of discourse**
- per ogni *termine linguistico* si ha il relativo *fuzzy set* che possono avere forme diverse ma devono coprire tutto l'asse delle x tramite la rispettiva *membership function*

Bisogna però capire come siano le regole con la *logica fuzzy*. Si assuma di avere un insieme di nodi $V = \{v_1, v_2, \dots, v_n\}$ ognuno specificato da una variabile linguistica. Lo stato di ognuno di questi nodi è caratterizzato da un *termine linguistico*. Dato quindi un nodo v_i , regolato da un certo insieme di nodi $V' = \{v_k, v_j, \dots, v_l\}$ la *regola fuzzy* per v_i è definita da un insieme di cosiddetti **if-then statement**, avendo che lo stato dei nodi in V' funge da **antecedente** alla regola, essendo nell'*if*, e lo stato del nodo v_i è il **conseguente** della regola, essendo nel *then*.

Si ha un breve esempio, con due input (*service* e *food*) e un output, *tip*, che descrive il processo di scelta di dare o meno una mancia al ristorante, in figura 4.8.

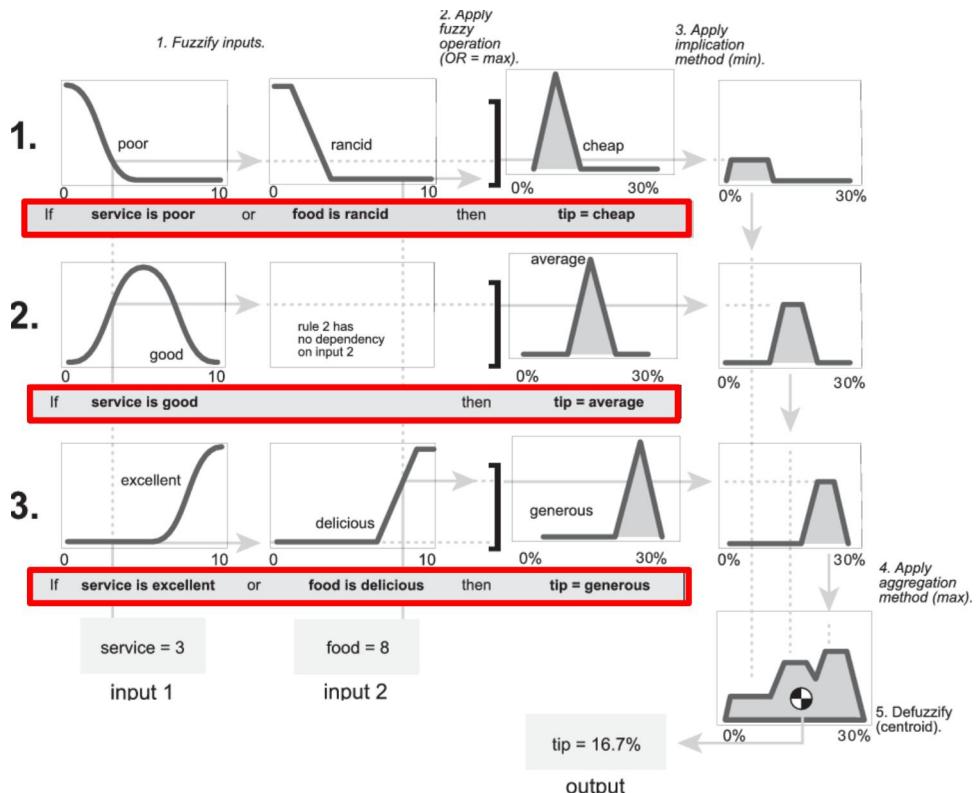


Figura 4.8: Esempio dove, come regola di *or*, si usa il max tra i due valori mentre le tre regole logiche sono bordate in rosso e sono scelte a proprio o in modo *knowledge-driven*, se possibile quindi partendo dalla letteratura, o *data-driven*, soluzione tendenzialmente meno efficace e non preferibile che parte dai dati. Le regole vengono valutate simultaneamente per ottenere lo stato al tempo $t + 1$ partendo dal tempo t . I μ degli input serviranno a calcolare il μ dell'output.

4.4 Seminario Logica Fuzzy

Viene qui messo un riassunto del seminario tenuto dal dottor Simone Spolaor sulla modellistica tramite logica fuzzy.

La **logica fuzzy** ha visto la sua nascita nel 1965 per mano di Lotfi Zadeh del quale si possono citare alcune frasi importanti:

“I had greatly underestimated the difficulty of designing machines that can approximate to the remarkable human ability to reason and make decisions in an environment of uncertainty and imprecision “

“We need a radically different kind of mathematics, the mathematics of fuzzy or cloudy quantities which are not describable in terms of probability distributions“

Negli anni settanta e ottanta, soprattutto in Giappone, si è avuto un “boom” nell’uso della *logica fuzzy*, che si poneva a metà tra i modelli logici e quelli meccanicistici, la quale è stata usata per moltissime soluzioni, tra cui:

- robot autonomi
- vari usi nella metropolitana in Giappone (il sistema frenante è tutt’ora gestito tramite *logica fuzzy*)
- macchinari per le pulizie
- ...

Inoltre tra gli anni ottanta e novanta si sono aggiunti moltissimi lavori teorici che usavano i *fuzzy set* nel paradigma scientifico, infatti la *logica fuzzy* può essere applicata per aumentare l’interpretabilità dei modelli e/o fornire approssimazioni convenienti di comportamenti dinamici noti. Si hanno infatti modelli che sono semplici da spiegare, ad esempio, a biologi/biotecnologi (molto di più di un insieme di *EDO*), avendo quindi che il loro uso ha preso piede sia nell’ambito della modellazione biologica, di natura estremamente eterogenea etc..., che in quello degli studi clinici. Un altro vantaggio è che tale logica è parsimoniosa dal punto di vista dei parametri e quindi si ha meno costo computazionale, avendo di conseguenza meno costo monetario. Come si è già accennato le regole della *logica fuzzy* sono basate sui cosiddetti *if-then statement* dove:

- la parte dell’*if* è detta *antecedente*
- la parte del *then* è detta *consequente*

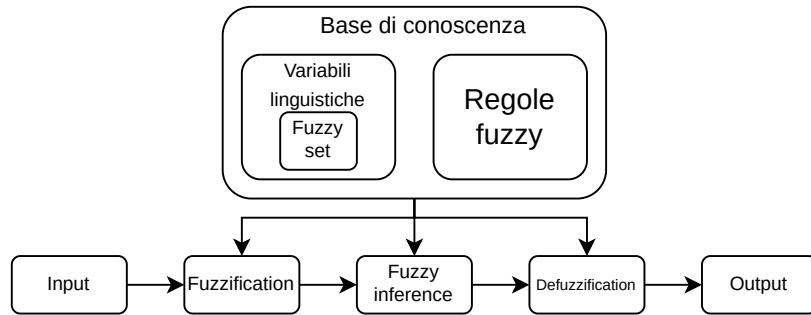


Figura 4.9: Schema generale del funzionamento di un sistema di inferenza fuzzy.

Si hanno quindi statement condizionali in cui gli antecedenti sono soddisfatti in una certa misura, avendo poi la necessità di aggregare l'output di diverse regole studiate in modo simultaneo. Inoltre l'antecedente può essere formato da più variabili che vengono elaborate insieme mediante operatori logici. Il significato di tali operatori logici non è standard ma nella maggior parte delle casistiche si ha:

- $\wedge \Rightarrow \min()$
- $\vee \Rightarrow \max()$
- $\neg \Rightarrow 1 - \mu$

Quindi ad esempio potremmo avere le seguenti regole:

- *IF A is High AND B is Low THEN C is High*
- *IF A is High OR B is Low THEN C is High*
- *IF A is NOT High THEN B is High*

Per lo studio dell'insieme delle varie formule si ha quindi bisogno di un **sistema di inferenza fuzzy**, ovvero un insieme di *variabili linguistiche* e *regole fuzzy*, che quindi consiste nel processo di mapping di un dato input in un output mediante *logica fuzzy*, dove sia l'input che l'output sono *crisp*, nel senso che sono valori precisi della variabile linguistica che vive sull'asse delle x . Questo processo può essere riassunto come in figura 4.9. Si hanno due principali metodi di inferenza:

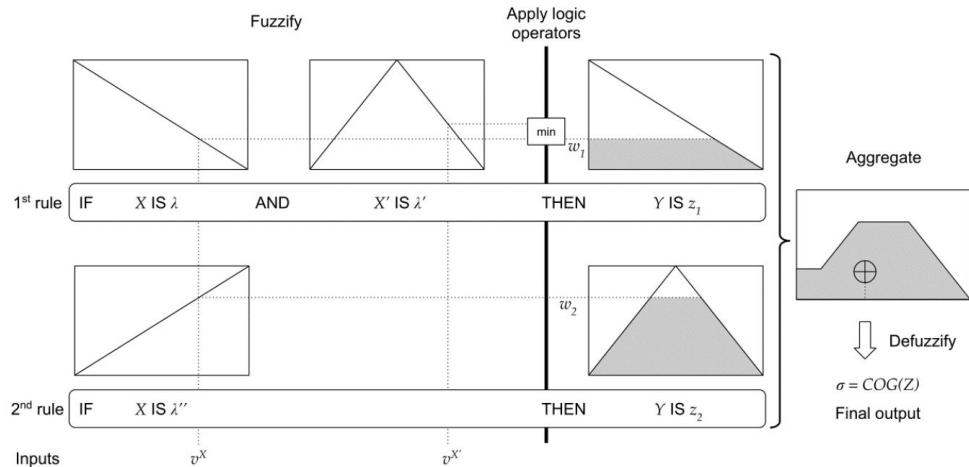
- il **metodo Mandani**
- il **metodo Takagi-Sugeno-Kang (TSK)**, spesso indicato anche solo con **metodo Sugeno**

4.4.1 Metodo Mandani

Il **metodo Mandani** è stato il primo metodo proposto, nel 1974, per l'*inferenza fuzzy* e consta di cinque step:

1. *fuzzificazione dell'input*
2. *applicazione degli operatori fuzzy*
3. *implicazione*
4. *aggregazione degli output*
5. *defuzzificazione*, step opzionale

Possiamo quindi riassumere tutto con un esempio grafico¹⁴:



Nell'esempio si vede l'applicazione di due regole, dati due input v^X e $v^{X'}$. Il conseguente è un *fuzzy set*. Si nota come con la fase di aggregazione si produce un poligono di forma non banale che subisce la fase di defuzzificazione come applicazione del calcolo del *COG*, ovvero *center of gravity*, per ottenere poi, tramite proiezione del risultato sull'asse delle x , l'output dell'intero processo. Il calcolo del *COG* è computazionalmente molto oneroso, soprattutto con aree non banali. Si nota quindi come il metodo dipenda fortemente ai metodi di aggregazione/defuzzificazione e che l'applicazione di molte regole possa dare risultati anche inaffidabili, a causa della creazione di un poligono troppo complesso. Quindi, per quanto il metodo sia di facile interpretazione, solitamente non viene utilizzato.

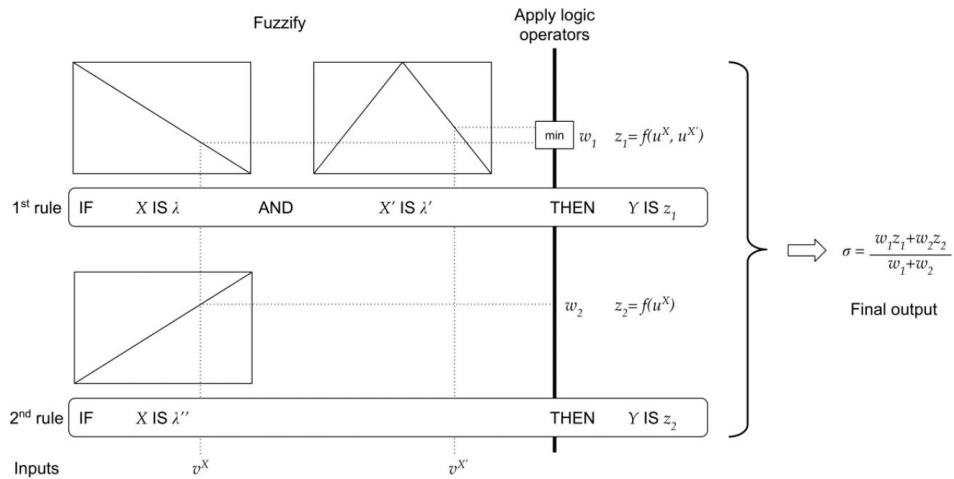
¹⁴Simone Spolaor

4.4.2 Metodo TSK

Il metodo **TSK** (*Takagi-Sugeno-Kang*) è invece uno dei metodi più usati ed è formato da quattro step:

1. *fuzzificazione dell'input*
2. *applicazione degli operatori fuzzy*
3. *implicazione*
4. *aggregazione degli output*

Possiamo quindi riassumere tutto con un esempio grafico¹⁵:



Si nota come, a differenza del *metodo Mandani*, il conseguente sia in questo caso una funzione f , che può essere lineare, gaussiana etc... essendo quindi più o meno complessa a seconda del caso e della necessità. Non si ha inoltre in ogni caso la necessità della defuzzificazione avendo che l'output si ottiene tramite una *media pesata*, riducendo quindi molto il costo computazionale e garantendo un output più affidabile. Purtroppo funzioni di ordine superiore portano a una minore interpretabilità.

¹⁵Simone Spolaor

4.4.3 Modelli Fuzzy Dinamici

Dopo aver introdotto i metodi di inferenza bisogna passare ai modelli. Nel 2010 Gegov propose il concetto di **modello dinamico fuzzy** come un modello basato su una **rete di inferenza fuzzy**, avendo quindi una rappresentazione formale tramite grafo dove i nodi rappresentano le *variabili linguistiche* mentre gli archi le *regole fuzzy*. Si è ottenuto quindi un formalismo che può essere adottato per modellare e simulare l’evoluzione temporale di sistemi eterogenei, a livello di dettaglio macroscopico.

4.4.4 Modello della Morte Cellulare Programmata

Si illustra ora un esempio d’uso dei modelli basati su logica fuzzy tramite una ricerca svolta dal dottor Spolaor, dalla professoressa Besozzi et al.

Lo scopo principale di questa ricerca era rispondere alla domanda:

Come possiamo indurre l’apoptosi nelle cellule tumorali, con una quantità minima di farmaci?

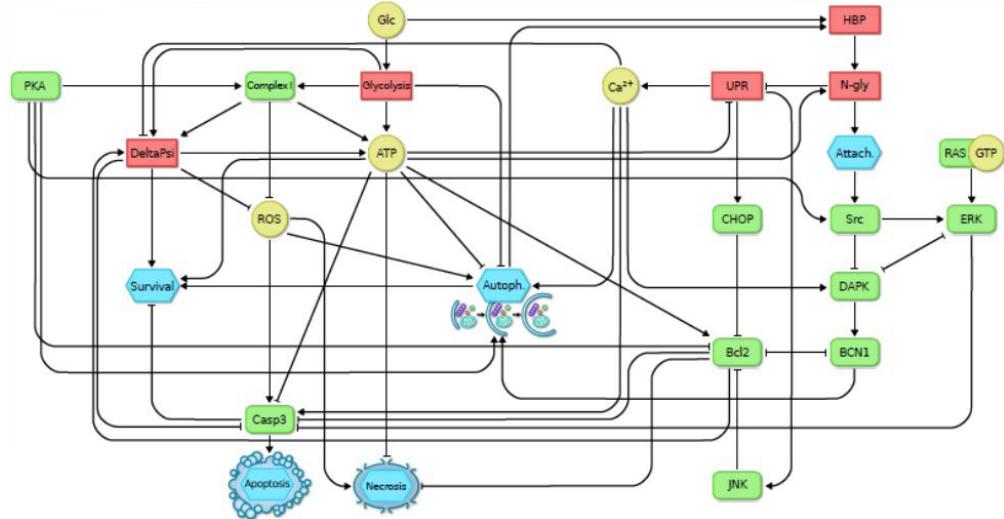
La morte cellulare programmata è infatti un processo biochimico complesso, infatti:

- include diversi distretti cellulari tra cui *ER (endoplasmic reticulum), mitocondri, superficie cellulare, etc...*
- include diversi attori biochimici, che vanno dalle piccole molecole agli organelli interi
- tutte le componenti sono fortemente connesse

Inoltre i dati disponibili, *sull’espressione genica, sull’imaging a fluorescenza etc...*, sono di natura qualitativa ed eterogenea e gli squilibri in questo processo sono coinvolti in diverse malattie complesse, incluso il cancro.

Questo è stato il primo tentativo di modellare tale processo tramite *modelli dinamici fuzzy* in quanto in letteratura erano presenti solo *modelli basati su reti booleane*.

Si è quindi arrivati al modello già visto precedentemente¹⁶:



Dove si contavano 25 *variabili linguistiche* e ben 252 *formule fuzzy* elaborate a mano in modalità *knowledge-driven*. Il modello si è dimostrato buono dopo una validazione tramite studi *in-vitro* e quindi è potuto procedere a studiare le simulazioni del modello in presenza di *perturbazioni*, con l'idea di cercare la combinazione minima di variabili perturbate in grado di massimizzare la morte apoptotica e ridurre al minimo la sopravvivenza, avendo che la *perturbazione* viene scelta tra i termini linguistici disponibili di ciascuna variabile. Si noti che la *perturbazione* di una variabile non avviene mediante il *metodo TSK*.

Per procedere con l'analisi è quindi usato un **algoritmo di ottimizzazione globale**, che studia il cosiddetto *spazio delle soluzioni*. Senza entrare nei dettagli un problema di ottimizzazione consiste nel trovare una soluzione ottimale a un dato problema. Si inoltre ha la **fitness function** che misura la qualità delle soluzioni possibili, funzione della quale bisogna trovare l'ottimo per risolvere un problema di ottimizzazione.

Nel dettaglio della ricerca si aveva uno *spazio delle soluzioni* che conteneva $3^6 \cdot 4^9 = 191102976$ possibili soluzioni. Per gestire uno spazio delle soluzioni così ampio si è scelto di usare uno degli algoritmi più famosi di ottimizzazione globale: l'algoritmo detto **Simulated Annealing**, che ottimizzava la seguente funzione di fitness:

$$F(\pi) = \frac{[s_{apoptosis}^\pi(t_b + \delta) - s_{apoptosis}^\pi(t_b)] - [s_{survival}^\pi(t_b + \delta) - s_{survival}^\pi(t_b)]}{\delta \cdot |\pi|}$$

¹⁶Nobile M.S. et al. Fuzzy modeling and global optimization to predict novel therapeutic targets in cancer cells. Bioinformatics, 2019

Dove:

- π è la perturbazione, avendo $F(\pi)$ come funzione di fitness
- $|\pi|$ è il numero di variabili perturbate da π
- δ è il tempo dopo il quale viene valutato l'effetto della perturbazione
- $[s_{apoptosis}^{\pi}(t_b + \delta) - s_{apoptosis}^{\pi}(t_b)]$ è il cambiamento per quanto riguarda l'apoptosi con l'azione della perturbazione
- $[s_{survival}^{\pi}(t_b + \delta) - s_{survival}^{\pi}(t_b)]$ è il cambiamento per quanto riguarda la sopravvivenza con l'azione della perturbazione

Non si entra nei dettagli della formula, in caso consultare il paper di riferimento¹⁷.

Ovviamente il range delle possibili soluzioni è stato anche testato in *wet-lab* ma si cercavano col modello bersagli per nuovi trattamenti chemioterapici. Tali bersagli erano, nel dettaglio:

- *UPR activation*
- *Complex I inhibition*
- *UPR activation and autophagy inhibition*
- *HBP and N-glycosylation inhibition*
- *N-glycosylation and autophagy inhibition*

I vari studi sono stati molto complessi e difficili da ottenere ma hanno portato a scoperte interessanti.

La domanda iniziale però è stata raffinata:

Possiamo ottenere soluzioni di dimensioni minime, massimizzando l'apoptosi e riducendo al minimo la necrosi?

Si è quindi passati ad un **problema di ottimizzazione multi-obiettivo**. Tale classe di problemi ha solitamente più di una soluzione ottima, che possono essere rappresentate da un cosiddetto *Pareto front*, che è un insieme che rappresenta tutte soluzioni egualmente ottimali, rappresentando anche i vari “tradeoff”, da bilanciare con le varie soluzioni ottime, tra i vari obiettivi.

¹⁷Nobile M.S. et al. Fuzzy modeling and global optimization to predict novel therapeutic targets in cancer cells. Bioinformatics, 2019

Si è quindi ripetuta l'analisi perturbativa sul modello di morte cellulare programmata, questa volta ottimizzando tre diversi obiettivi¹⁸:

1. massimizzare l'apoptosi
2. minimizzare la necrosi
3. minimizzare il numero di variabili perturbate

Questo discorso non viene approfondito.

Interessante notare come l'uso di *modelli dinamici fuzzy* sia cruciale in altri progetti in corso (fatti in collaborazione con la Fondazione Tettamanti e l'Università dell'Insubria), tra cui:

- modellazione dell'insorgenza della *Precursor B cell Acute Lymphoblastic Leukemia (BCP-ALL)* in pazienti pediatrici
- previsione della risposta terapeutica nei pazienti pediatrici affetti da *T cell Acute Lymphoblastic Leukemia (T-ALL)*
- modellazione della rete mitocondriale disfunzionale nella *malattia di Parkinson*

4.4.5 Simpful

Al fine di poter effettuare simulazioni tramite *modelli dinamici fuzzy* il dottor Spolaor ha sviluppato una libreria in *python*, chiamata **simpful**¹⁹ per usare la *logica fuzzy* e di conseguenza modellare *modelli dinamici fuzzy*. Un esempio è visibile al listing 1.

Sono attualmente supportate le seguenti funzionalità:

- *fuzzy set* con funzioni poligonali e/o personalizzate
- definizione di regole fuzzy come stringhe di testo
- regole logiche arbitrariamente complesse costruite con operatori logici classici, \wedge , \vee , \neg
- *metodo TSK* per l'inferenza a qualsiasi ordine
- varie utility per i plot

¹⁸Spolaor S. et al. Screening for Combination Cancer Therapies With Dynamic Fuzzy Modeling and Multi-Objective Optimization. *Frontiers in Genetics*, 2021

¹⁹Spolaor S. et al. Simpful: A User-Friendly Python Library for Fuzzy Logic. *International Journal of Computational Intelligence Systems*, 2020

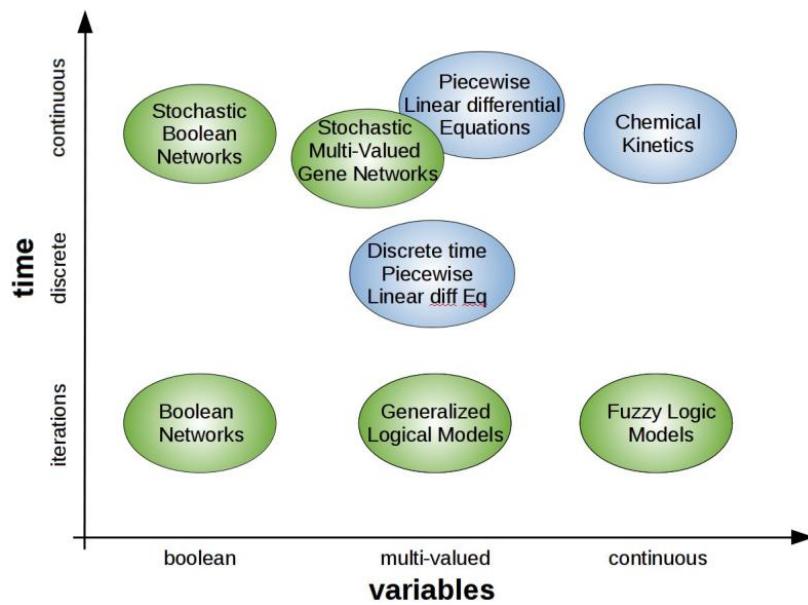


Figura 4.10: Overview delle varie tipologie di modelli logici, in verde, e di alcuni tipi di modelli meccanicistici, in blu.

Il core della libreria consiste nell'eseguire un'analisi ricorsiva delle regole per costruire rappresentazioni eseguibili degli antecedenti, sotto forma di *alberi di derivazione*.

4.5 Note Conclusive

Si mettono ora alcune ultime note riguardo questa categoria di modelli.

Come visto nel capitolo ci si è focalizzati soprattutto su *modelli booleani* e su *modelli basati su logica fuzzy*. In realtà si hanno molte altre varianti, come visibile in figura 4.10²⁰, dove le varie soluzioni vengono catalogate in base a come sono caratterizzati le variabili e a come viene rappresentato il tempo. Nella figura si inizia a mostrare un confronto tra *modelli logici* e *modelli meccanicistici*, confronto che verrà approfondito anche più avanti nel corso.

²⁰Le Novère N. Quantitative and logic modelling of molecular and gene networks. Nat Rev Genet. 2015;16(3):146-158. doi:10.1038/nrg3885

Listing 1 Esempio del repressilator modellato tramite logica fuzzy. Altri esempi sono disponibili presso <https://github.com/aresio/simpful/tree/master/examples>

```

from simpful import *
from copy import deepcopy
import seaborn as sns
import matplotlib.pyplot as plt

# A simple dynamic fuzzy model of the repressilator
# Create a fuzzy reasoner object
FS = FuzzySystem()

# Define fuzzy sets and linguistic variables
LV = AutoTriangle(2, terms=['low', 'high'])
FS.add_linguistic_variable("LacI", LV)
FS.add_linguistic_variable("TetR", LV)
FS.add_linguistic_variable("CI", LV)

# Define output crisp values
FS.set_crisp_output_value("low", 0.0)
FS.set_crisp_output_value("high", 1.0)

# Define fuzzy rules
RULES = []
RULES.append("IF (LacI IS low) THEN (TetR IS high)")
RULES.append("IF (LacI IS high) THEN (TetR IS low)")
RULES.append("IF (TetR IS low) THEN (CI IS high)")
RULES.append("IF (TetR IS high) THEN (CI IS low)")
RULES.append("IF (CI IS low) THEN (LacI IS high)")
RULES.append("IF (CI IS high) THEN (LacI IS low)")
FS.add_rules(RULES)

# Set antecedents values
FS.set_variable("LacI", 1.0)
FS.set_variable("TetR", 0.5)
FS.set_variable("CI", 0.0)

# Set simulation steps and save initial state
steps = 14
dynamics = []
dynamics.append(deepcopy(FS._variables))

# At each simulation step, perform Sugeno inference, update state and save the results
for i in range(steps):
    new_values = FS.inference()
    FS._variables.update(new_values)
    dynamics.append(new_values)

# Plot the dynamics
lac = [d["LacI"] for d in dynamics]
tet = [d["TetR"] for d in dynamics]
ci = [d["CI"] for d in dynamics]
plt.plot(range(steps+1), lac)
plt.plot(range(steps+1), tet)
plt.plot(range(steps+1), ci)
plt.ylim(0,1.05)
plt.xlabel("Time")
plt.ylabel("Level")
plt.legend(["LacI", "TetR", "CI"], loc="lower right", framealpha=1.0)
plt.show()

```

4.6 Software

Per lo studio di modelli logici sono stati sviluppati nel tempo vari tool computazionali, ben riassunti nella seguente tabella²¹:

tool	type of logic	functionality	treatment of time
BooleanNet	Boolean, piecewise linear	simulation and visualization	synchronous, asynchronous, or continuous
GinSim	discrete (multistate)	model building, simulation, and analysis	synchronous, asynchronous, or mixed asynchronous
CellNetAnalyzer	Boolean (multistate)	model simulation, visualization, and network properties analysis	logic steady state
CellNetOptimizer	Boolean	model training and simulation	logic steady state
Odefy	Boolean, logic-based ODEs	model simulation and visualization	synchronous, asynchronous, or continuous
Genetic Network Analyzer	piecewise linear	model building and simulation	continuous
ChemChains	Boolean	model simulation, visualization, and analysis	synchronous or asynchronous
MetaReg	discrete (multistate)	model simulation, visualization, and refinement	logic steady state
SQUAD	standardized qualitative dynamical systems	model simulation and analysis	continuous

La maggior parte dei tool disponibili sono comunque per modelli booleani o multi-stato, non per modelli fuzzy che, come già anticipato, sono stati in buona parte introdotti dal gruppo di ricerca della professoressa Besozzi e che sono stati “wrappati” nella libreria *simpful*. Esistono comunque altre librerie per lavorare con modelli basati su *logiche fuzzy*.

In merito alla questione software si consiglia la lettura del paper di Niarakis e Helikar²², disponibile anche su Moodle, che è un buon paper introduttivo, in sorta una guida pratica, alla simulazione booleana via software.

Su moodle sono indicati vari paper interessanti sui *modelli logic-based*.

²¹Morris, Melody K., et al. "Logic-based models for the analysis of cell signaling networks." Biochemistry 49.15 (2010): 3216-3224.

²²A. Niarakis and T. Helikar, A practical guide to mechanistic systems modeling in biology using a logic-based approach, Brief. Bioinf. 2020

Capitolo 5

Mechanism-Based Modelling

Si introducono ora i modelli **mechanism-based**.

Ricordiamo che tali modelli, come visibile in figura 2.6:

- hanno un sistema di piccole dimensioni, essendo quindi *modelli small-scale*
- presentano tendenzialmente un elevato costo computazionale per le analisi, dovuto all'elevata parametrizzazione delle componenti
- hanno un alto livello di dettaglio, avendo una descrizione dettagliata delle componenti
- presentano difficoltà nella misurazione dei dati conseguente alla necessità di caratterizzare a fondo le componenti del modello stesso

Nel complesso si hanno quindi modelli con altissima *capacità predittiva*, essendo per di più *modelli quantitativi*, che sono come anticipato **fully parameterized** (fattore che rende difficile lo studio di tali modelli), e *dinamici*. Nel dettaglio, per l'aspetto legato alla dinamicità, si ha una *rappresentazione continua del tempo*, rappresentazione che cerca di essere il più vicina possibile alla realtà biologica. Un breve confronto coi *modelli logic-based* è visibile in figura 5.1¹.

Questa è la classe più complessa ed eterogenea di approcci di modellazione, avendo che comprende molti formalismi matematici diversi. In generale le simulazioni usate per studiare l'evoluzione temporale, quindi la dinamica, del sistema sfruttano diversi metodi:

¹Le Novère N. Quantitative and logic modelling of molecular and gene networks. Nat Rev Genet. 2015;16(3):146-158. doi:10.1038/nrg3885

Capitolo 5. Mechanism-Based Modelling

	Quantitative model	Logic model
Suitable for	Time series	Phenotypes
Time representation	Linear representation	Abstract iterations
Variables	Quantitative	Qualitative
Mechanism representation	Yes	No
What can we do?	Compute concentrations and durations; evaluate the effect of parameter values	Compute state transitions and attractors (steady-states and cyclic attractors)
Data necessary to build the model	Molecular species, genes, interactions, biochemical processes	Activities, defined phenotypes, rules linking those
Data to parameterize and validate the model	Amount of molecular species, timecourses, quantitative phenotype	Perturbations of activities such as RNA interference, inhibitors, qualitative phenotypes
Advantages	Quantitative, precise; direct comparison with quantitative measurements; large existing toolkit	Easy to build; easy to compose; easy simulation of perturbations
Weaknesses	Requires quantitative knowledge of initial conditions and kinetics	Cannot provide quantitative predictions; difficult to choose between alternative behaviours

Figura 5.1: Confronto tra modelli quantitativi, come i *modelli mechanism-based*, dove si nota lo studio esplicito delle *time series*, con l’evoluzione nel tempo delle variabili, e modelli qualitativi, come i *modelli logic-based*.

- *deterministici*
- *stocastici*
- *ibridi*, ovvero sia deterministici che stocastici

Inoltre il cuore di tali modelli sono appunto i **parametri** e anche per il loro studio, al fine di calibrare e analizzare il modello, si hanno varie tecniche, tra cui:

- *parameter sweep analysis/scan*, che è il metodo più semplice
- *parameter estimation*, metodo più complesso ma anche più importante, basato sulla risoluzione di problemi di ottimizzazione globale
- *sensitivity analysis*, per l’identificazione delle componenti più fragili o più importanti del sistema stesso

In figura 5.2² troviamo inoltre un breve confronto tra i pro e i contro dei vari approcci modellistici. Se ci si concentra sui *modelli mechanism based* si nota come le difficoltà di ottenere i dati quantitativi, parametrizzare il modello ed effettivamente procedere con la simulazione, che ha un costo computazionale non indifferente, siano fattori da tenere in considerazione in fase di scelta del modello. In questo tipo di modelli si hanno principalmente variabili intere o

²Bordbar, Aarash, et al. "Constraint-based models predict metabolic and associated cellular functions." Nature Reviews Genetics 15.2 (2014): 107-120.

Capitolo 5. Mechanism-Based Modelling

Method	Model systems	Parameterization	Typical prediction type	Advantages	Disadvantages
Stochastic kinetic modelling	Small-scale biological processes	Detailed kinetic parameters	Reaction fluxes, component concentrations and regulatory states	<ul style="list-style-type: none"> Mechanistic Dynamic Captures biological stochasticity and biophysics 	<ul style="list-style-type: none"> Computationally intensive Difficult to parameterize Challenging to model multiple timescales
Deterministic kinetic modelling	Small-scale biological processes	Detailed kinetic parameters	Reaction fluxes, component concentrations and regulatory states	<ul style="list-style-type: none"> Mechanistic Dynamic 	<ul style="list-style-type: none"> Computationally intensive Difficult to parameterize
Constraint-based modelling	Genome-scale metabolism	Network topology, and uptake and secretion rates	Metabolic flux states and gene essentiality	<ul style="list-style-type: none"> Mechanistic Large scale No kinetic information is required 	<ul style="list-style-type: none"> No inherent dynamic or regulatory predictions No explicit representation of metabolic concentrations
Logical, Boolean or rule-based formalisms	Signalling networks and transcriptional regulatory networks	Rule-based interaction network	Global activity states and on-off states of genes	Can model dynamics and regulation	Biological systems are rarely discrete
Bayesian approaches	Gene regulatory networks and signalling networks	High-throughput data sets	Probability distribution score	<ul style="list-style-type: none"> Non-biased Can include disparate and even non-biological data Takes previous associations into account 	<ul style="list-style-type: none"> Statistical Issues of over-fitting Requires comprehensive training data
Graph and interaction networks	Protein-protein and genetic interaction networks	Interaction network that is based on biological data	Enriched clusters of genes and proteins	<ul style="list-style-type: none"> Incorporates prior biological data Encompasses most cellular processes 	Dynamics are not explicitly represented
Pathway enrichment analysis	Metabolic and signalling networks	Pathway databases (for example, KEGG, Gene Ontology and BioCyc)	Enriched pathways	<ul style="list-style-type: none"> Simple and quick Takes prior knowledge into account 	<ul style="list-style-type: none"> Biased to human-defined pathways Non-modelling approach

Figura 5.2: Confronto sommario tra i vari approcci di modellazione.

reali, ad esempio per rappresentare la concentrazione o il numero di molecole. Tutte le variabili devono essere inizializzate, anche a zero (o al valore nullo corrispondente), prima di procedere con la simulazione, altrimenti si avrà solo errori (anche perché dal punto di vista algoritmico è necessario il caso iniziale).

Come detto si ha un range molto ampio di formalismi matematici. Tra i più importanti si hanno:

- *modelli reaction-based*, che verranno approfonditi nel corso, in quanto facili da comprendere e con un semplice formalismo matematico. Sono *modelli quantitativi, dinamici, fully parameterized* e solitamente utilizzati per sistemi biologici su piccola scala, oltre ad essere di facile comprensione anche per biologi/biotecnologi nonostante abbiano diversi vantaggi rispetto ad altri formalismi
- *equazioni differenziali ordinarie (EDO)*
- *equazioni differenziali parziali (EDP)*
- *modelli rule-based*, che verranno approfonditi in un seminario
- *reti di Petri*

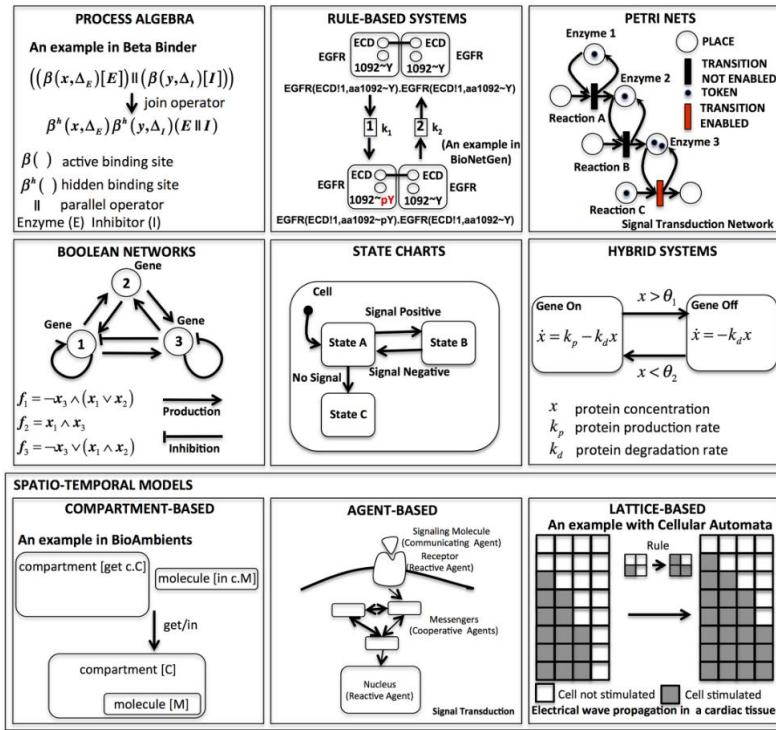


Figura 5.3: Vari toy examples per esempi di modellazione, meccanicistica e non.

- *algebre di processi*, tendenzialmente da evitare in quanto inutilmente complesse

Brevi esempi di alcuni di questi formalismi sono visibili in figura 5.3³.

5.1 Reaction-Based Models

Approfondiamo quindi i *modelli reaction-based*.

Dato un sistema biologico con qualsiasi tipo di componenti interagenti (che possono essere molecole, cellule, persone, animali etc...), una formalizzazione delle sue componenti e delle loro interazioni può essere fatta specificando:

- un *insieme di specie*, nel nostro caso un *insieme di specie molecolari* $\mathcal{S} = \{S_1, S_2, \dots, S_n\}$

³Bartocci, Ezio, and Pietro Lió. "Computational modeling, formal analysis, and tools for systems biology." PLoS computational biology 12.1 (2016): e1004591.

- un *insieme di reazioni*, nel nostro caso un *insieme di reazioni biochimiche* $\mathcal{R} = \{R_1, R_2, \dots, R_m\}$
- un *insieme di quantità iniziali* $\mathcal{X} = \{X_1, X_2, \dots, X_n\}$, avendo quindi una quantità per ogni elemento di \mathcal{S} . Si ha che o $X_i \in \mathbb{N}$ o $X_i \in \mathbb{R}$
- un *insieme di costanti*, nel nostro caso un *insieme di costanti cinetiche* $\mathcal{K} = \{K_1, K_2, \dots, K_m\}$ che caratterizzano ogni singola reazione in \mathcal{R} tramite le proprietà fisico/chimiche
- si assume di avere un volume \mathcal{V} dove avvengono le reazioni. Tale volume è assunto come definito a priori e costante. Inoltre è *well-stirred (ben mescolato)*, avendo che le componenti si distribuiscono in modo uniforme al suo interno. Queste, volume costante/distribuzione uniforme, non sono assunzioni realistiche, anche se a volte lo sono di più che in altre (basti pensare al caso della *diffusione* dove non si ha assolutamente una distribuzione uniforme). Sono comunque approssimazioni accettabili nella maggioranza dei casi e, si vedrà, potranno anche essere “superate” in determinate situazioni

Si è parlato di *specie* e nel dettaglio essere caratterizzano ogni componente del sistema, quindi, nel caso biologico, si potrebbe parlare di:

- ioni
- metaboliti
- geni
- proteine
- ...

Bisogna inoltre considerare i *complessi* che si formano tramite legami tra più componenti. A livello notazionale una componente la identifichiamo con p_i mentre per un complesso potremmo usare vari formalismi, basta che siano coerenti per tutto il modello. Dati P_1 e p_2 potremmo avere:

- $p_1 p_2$
- $p_1 * p_2$
- $p_1 - p_2$

- $p_1 : p_2$
- ...

Si noti che $p_1 p_2$ è diverso da $p_2 p_1$, sia per motivazioni sintattiche che semantiche (magari dovute alla matematica implicita che non commuta).

In questi modelli le quantità di specie possono essere fornite come numero di molecole, via numeri interi, o come concentrazioni, via numeri reali e il loro uso varia a seconda del metodo:

- nei modelli stocastici si usa il numero di molecole
- nei modelli deterministi si usano le concentrazioni

In ogni caso si può passare da uno all'altro usando il volume \mathcal{V} e il numero di Avogadro $N_A = 6,02214076 \times 10^{23}$ avendo che:

$$\text{numero di specie in } \mathcal{S} = \text{concentrazione di } \mathcal{S} \cdot N_A \cdot \mathcal{V}$$

Parlando invece delle costanti legate alle reazioni si ha che sono caratterizzati da un numero reale non negativo e la loro specifica dipende anch'essa dall'interpretazione del modello:

- nei modelli stocastici si ha la probabilità che avvenga una reazione. In questo caso le costanti si indicano con c
- nei modelli deterministi si ha il *reaction rate*. In questo caso le costanti si indicano con k

Si ha una generalizzazione per passare da c a k , che dipende dal numero di reagenti e dal loro ordine, ma per ora basta la seguente tabella riassuntiva:

caso	formula
1 reagente	$c = k$
2 reagenti di specie diverse	$c = \frac{k}{N_A \cdot \mathcal{V}}$
2 reagenti di specie uguali	$c = \frac{2k}{N_A \cdot \mathcal{V}}$

A questo punto possiamo descrivere una reazione R come un insieme di reagenti e un insieme di prodotti, entrambi formati da elementi di \mathcal{S} , avendo che:



dove:

- a sinistra del \rightarrow si hanno i reagenti e a destra i prodotti della reazione
- i vari + segnalano la coesistenza di reagenti/prodotti (bisogna avere tutti quei reagenti e si ottengono tutti quei prodotti)
- ogni reagente e ogni prodotto sono caratterizzati da un *coefficiente stechiometrico* a_i per i reagenti e b_i per i prodotti. Tali coefficienti possono essere nulli. Qualora $a_i = 1$ o $b_i = 1$ solitamente si omette nel formalizzare la reazione (scrivo S_i e non $1S_i$)

Grazie al *coefficiente stechiometrico* si passa ad una descrizione non astratta del meccanismo che si ha dietro una reazione.

Esempio 2. Per chiarire meglio le idee vediamo un esempio.

Si assume:

$$\mathcal{S} = \{A, B, C, AB, ABC\}$$

Posso quindi avere le seguenti reazioni (dove per semplicità si omette il k), ad esempio:



Da questo esempio semplificato si notano varie cose:

- i composti devono essere conosciuti a priori e specificati in \mathcal{S}
- le reazioni possibili sono conosciute a priori nel modello
- è preferibile avere una catena di **reazioni di secondo ordine** (quindi con solo due componenti come reagenti). Questo viene fatto sia per fedeltà alla realtà biologica che per praticità nel formalismo matematico
- non sono modellate le inibizioni che quindi dovranno essere modellate in modo esplicito come reazioni precise nel modello

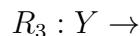
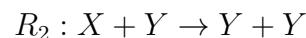
Il primo vantaggio dei *modelli reaction-based* è che sfruttano il linguaggio della biochimica essendo di semplice comprensione/modellazione anche per biologi/biotecnologi in quanto non richiede forti basi matematiche e aiuta quindi la comunicazione con i modellisti. Questo tipo di formalizzazione è abbastanza generale da descrivere qualsiasi tipo di processo determinato da componenti interagenti, a condizione che la semantica appropriata sia assegnata all'insieme delle specie e all'insieme delle reazioni.

5.1.1 Sistema di Lotka-Volterra

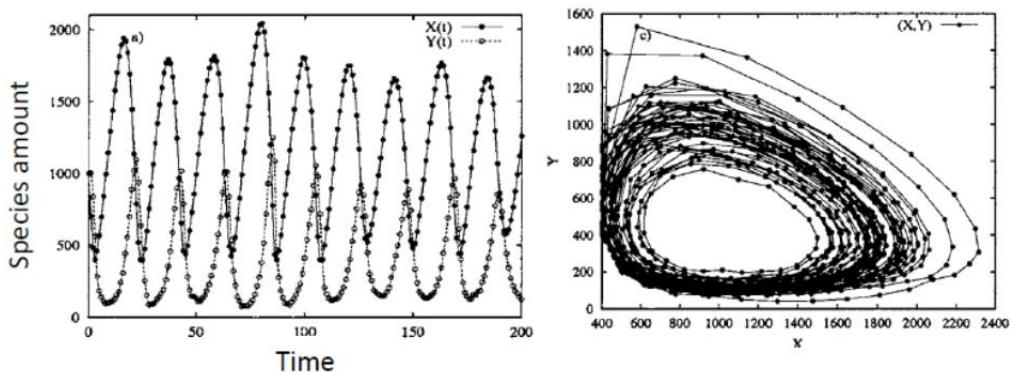
Un esempio famoso, per quanto di ambito ecologico, è quello del **sistema di Lotka-Volterra**. In questo sistema si hanno tre specie:

1. l'insieme A per il cibo
2. l'insieme X per le prede
3. l'insieme Y per i predatori

Il comportamento è quello che le prede, in presenza di cibo, aumentano di numero. I predatori, in presenza di prede, aumentano di numero mentre in assenza delle stesse spariscono. La quantità di cibo rimane costante osservando quindi un comportamento oscillatorio, sfasato, per le altre due specie. Si hanno, per rappresentare tutto ciò, le seguenti reazioni:



Si assumono $k_1 \geq 0$, $k_2 \geq 0$ e $k_3 \geq 0$. Come detto $|A|$ è fissato, magari ad esempio $|A| = 100$, mentre $|X|$ e $|Y|$ sono variabili, entrambi con stato iniziale fissato a 1000. Si ottengo quindi ad esempio i seguenti risultati di una simulazione stocastica, con i grafici che sono due rappresentazioni analoghe del comportamento emergente dinamico del sistema, dato lo stato iniziale appena definito:



Dove nel grafico a sinistra si ha lo stesso output che si otterrebbe, ad esempio, con lo studio di un sistema di EDO, avendo quindi un comportamento oscillatorio non in fase (in quanto al crescere dei predatori diminuiscono le

prede e mano a mano che diminuiscono le prede diminuisce anche il numero di predatori, per poi cominciare a crescere il numero di prede etc. . .), mentre nel grafico a destra si plottano le coppie (X_i, Y_i) , avendo il cosiddetto **spazio delle fasi**. Il tempo non viene esplicitamente rappresentato in questo caso ma viene rappresentato in modo implicito, in quanto la curva (che si noti si sviluppa in senso antiorario) che si viene a generare può essere caratterizzata in modo dinamico, ogni nuovo punto specifica il trascorrere del tempo. In questo caso il plot è ottenuto tramite metodo stocastico in quanto con un metodo deterministico si avrebbe un'unico “ovale”, una *curva chiusa*, in quanto si tornerebbe, in modo periodico, nelle stesse coordinate (non variando ampiezza e frequenza come nel caso stocastico). Si nota quindi come il sistema si vada ogni volta a bilanciare, prima crescono le prede, poi i predatori calano le prede, poi calano i predatori aumentando le prede etc. . .

L'ampiezza diverso nel grafico a sinistra è anch'essa dovuta alla simulazione stocastica. Anche se meno visualizzabile anche la frequenza è variabile a causa della simulazione stocastica.

In questo contesto risulta interessante la **teoria delle biforazioni**, non approfondita nel corso. La *teoria delle biforazioni* è una teoria matematica che si occupa dello studio dei cambiamenti qualitativi o della struttura topologica di integrali di un campo vettoriale o, equivalentemente, dalla soluzione di un'equazione differenziale. In pratica si studia, al variare dei parametri, il variare comportamento del sistema (magari passando da lineare a oscillatorio).

Giusto per completezza, dato il tempo t , $X(t)$ il numero di prede al tempo t e $Y(t)$ il numero di predatori sempre al tempo t , si avrebbe il seguente sistema di equazioni differenziali ordinarie⁴:

$$\begin{cases} \frac{dX}{dt} = (A - B \cdot Y) \cdot X \\ \frac{dY}{dt} = (C \cdot X - D) \cdot Y \end{cases}$$

dove A, B, C, D sono parametri positivi che descrivono l'interazione tra le due specie.

5.1.2 Dinamica dei Modelli Reaction-Based

Vediamo quindi più nel dettaglio l'aspetto della dinamica nei *modelli reaction-based*.

Dato un *modello reaction-based* possiamo determinare come cambia lo stato del sistema biologico nel tempo, in pratica vediamo cosa succede effettuando

⁴https://it.wikipedia.org/wiki/Equazioni_di_Lotka-Volterra

delle *simulazioni*, capendo come partire da uno *stato iniziale*, a $t = 0$, dove si ha la quantità iniziale di tutte le specie del sistema. Si procede quindi partendo dallo stato al tempo t per ottenere lo stato al tempo $t + 1$, esattamente come visto nei modelli logic-based.

Definiamo quindi lo stato del sistema al tempo t come $X(t)$, che è un vettore che contiene le quantità di ogni specie al tempo t :

$$X(t) = (X_1(t), X_2(t), \dots, X_n(t))$$

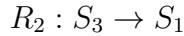
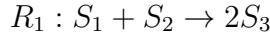
dove $X_1(t), X_2(t), \dots, X_n(t)$ sono le quantità al tempo t delle specie S_1, S_2, \dots, S_n . Lo stato del sistema si aggiorna quindi automaticamente mediante un *algoritmo di simulazione*.

Esempio 3. Si vede un breve *toy example* per capire meglio il discorso, “nascondendo sotto il tappeto” gli aspetti più matematici della simulazione vera e propria.

Sia dato:

$$\mathcal{S} = \{S_1, S_2, S_3\}$$

e le seguenti reazioni:



Si assume il seguente stato al tempo t :

$$X(t) = (X_1(t), X_2(t), X_3(t)) = (15, 7, 23)$$

Le reazioni vengono studiate in modo sequenziale, una reazione alla volta, sommando il numero di molecole dei prodotti e sottraendo il numero di molecole dei reagenti. Non si ragiona in termini di moli ma di brutale numero di molecole, quindi con numeri in \mathbb{N} . Ci sono algoritmi stocastici che permettono lo studio parallelo di più reazioni, anche se con qualche “effetto collaterale”, ma non è il caso di questo esempio, che semplifica la massimo l’algoritmo stocastico di Gillespie.

Si simula quindi R_1 , ottenendo:

$$X(t+1) = (15 - 1, 7 - 1, 23 + 2) = (14, 6, 25)$$

Si simula quindi R_2 , partendo dallo stato appena ottenuto al tempo $t + 1$, ottenendo:

$$X(t+2) = (14 + 1, 6 + 0, 25 - 1) = (15, 6, 24)$$

Questo è una semplificazione di quanto succede con una simulazione stocastica mentre nel caso di una simulazione deterministica si avrebbe che tutte le specie verrebbero modificate contemporaneamente.

Uno dei vantaggi dei *modelli reaction-based* è quello che sono formalizzati in modo tale da supportare sia simulazioni deterministiche che simulazioni stocastiche. Secondo la formulazione stocastica della cinetica chimica⁵, la formalizzazione di una rete biochimica come un insieme di specie e un insieme di reazioni può essere utilizzata direttamente per eseguire algoritmi di simulazione stocastica, previa descrizione del modello mediante, appunto, insiemi di reazioni.

5.1.3 Dalle Reazioni alle Equazioni Differenziali

È possibile trasformare qualsiasi *modello reaction-based* nel corrispondente sistema di equazioni differenziali ordinarie, tendenzialmente in modo automatico sfruttando la **legge di azione di massa (Law of Mass-Action)**. Tali *EDO* possono essere:

- *del primo ordine*, ovvero equazioni differenziali che stabiliscono una relazione tra una variabile indipendente x , la funzione incognita $y = f(x)$, e la derivata prima y'
- *accoppiate*, se, date due o più *EDO*, non è possibile risolverle singolarmente

Il *sistema di EDO* viene poi simulato tramite **algoritmi di integrazione numerica**. Nel dettaglio ogni equazione differenziale, che in abito biologico/biochimico viene anche nominata *rate equation*, descrive la variazione nel tempo della centrale di ogni specie molecolare X , considerando ogni singola interazione con altre specie. La concentrazione di X viene specificata con $[X]$. Si hanno quindi:

- *variazioni positive*, che accrescono la concentrazione di X
- *variazioni negative*, che diminuiscono la concentrazione di X

Facendo un breve esempio:

$$\frac{d[X]}{dt} = [\dot{X}] = \text{sintesi} - \text{fosforilazione} + \text{defosforilazione} - \text{legami} + \dots$$

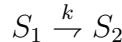
Quindi, a differenza di quanto visto nel toy example precedente ogni specie è legata ad una singola *EDO* e lo studio non prevede di simulare una reazione per volta ma bensì tutte le *EDO* in modo parallelo, studiando quindi

⁵Exact stochastic simulation of coupled chemical reactions, Daniel T. Gillespie The Journal of Physical Chemistry 1977 81 (25), 2340-2361 DOI: 10.1021/j100540a008

contemporaneamente tutte le variazioni di tutte le specie.

Si vedono quindi vari esempi di passaggio dalla sintassi per le reazioni alle EOD.

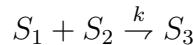
Esempio 4. Si prenda la semplice reazione, dove S_1 viene consumato, a velocità k , aumentando la concentrazione di S_2 :



Questa verrà modellata da:

$$\begin{cases} \dot{[S_1]} = -k[S_1] \\ \dot{[S_2]} = k[S_1] \end{cases}$$

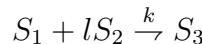
Esempio 5. Si prenda la semplice reazione:



Questa verrà modellata da:

$$\begin{cases} \dot{[S_1]} = -k[S_1][S_2] \\ \dot{[S_2]} = -k[S_1][S_2] \\ \dot{[S_3]} = k[S_1][S_2] \end{cases}$$

Esempio 6. Si prenda la semplice reazione:



Questa verrà modellata da:

$$\begin{cases} \dot{[S_1]} = -k[S_1][S_2]^l \\ \dot{[S_2]} = -kl[S_1][S_2]^l \\ \dot{[S_3]} = k[S_1][S_2]^l \end{cases}$$

Dove si nota che il coefficiente stechiometrico passa all'esponente, oltre che a moltiplicare il k per la EDO di S_2 .

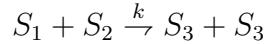
Esempio 7. Si prenda la semplice reazione:



Questa verrà modellata da:

$$\begin{cases} \dot{[S_1]} = -2k[S]^2 \\ \dots \end{cases}$$

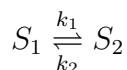
Esempio 8. Si prenda la semplice reazione:



Questa verrà modellata da:

$$\begin{cases} \dots \\ [\dot{S}_3] = 2k[S_1][S_2] \end{cases}$$

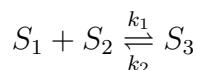
Esempio 9. Si prenda la semplice reazione:



Questa verrà modellata da:

$$\begin{cases} [\dot{S}_1] = -k_1[S_1] + k_2[S_2] \\ [\dot{S}_2] = k_1[S_1] - k_2[S_2] \end{cases}$$

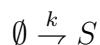
Esempio 10. Si prenda la semplice reazione:



Questa verrà modellata da:

$$\begin{cases} [\dot{S}_1] = -k_1[S_1][S_2] + k_2[S_3] \\ [\dot{S}_2] = -k_1[S_1][S_2] + k_2[S_3] \\ [\dot{S}_3] = k_1[S_1][S_2] - k_2[S_3] \end{cases}$$

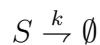
Esempio 11. Si prenda la semplice reazione:



Questa verrà modellata da:

$$[\dot{S}] = k$$

Esempio 12. Si prenda la semplice reazione:



Questa verrà modellata da:

$$[\dot{S}] = -k[S]$$

Come già anticipato bisogna tendenzialmente usare reazioni al più di secondo ordine, quindi con due reagenti.

In alcuni casi posso avere anche *EDO* non derivate dalle reazioni tramite la *Law of Mass-Action* ma si possono anche usare altre equazioni arbitrarie, come ad esempio sinusoidali o esponenziali, sempre atte a rappresentare la variazione di concentrazione nel tempo. Un esempio di tale funzione è la famosa **equazione di Hill**, che verrà approfondita a breve.

Si vede ora un esempio più interessante.

Esempio 13. Si prenda la seguente reazione:



dove:

- *S* è il substrato
- *E* è l'enzima
- *ES* è il complesso enzima-substrato, formato quindi da *E* e *S*
- *P* è il prodotto della reazione
- $k_f \geq 0$, $k_r \geq 0$ e $k_{cat} \geq 0$, avendo che quest'ultima è irreversibile

In pratica è come se avessimo tre reazioni:

1. $R_1 : E + S \xrightarrow{k_f} ES, k_f \geq 0$
2. $R_2 : ES \xrightarrow{k_r} E + S, k_r \geq 0$
3. $R_3 : ES \xrightarrow{k_{cat}} E + P, k_{cat} \geq 0$

Questa reazione viene modellata dal seguente sistema di *EDO*, dove per ogni specie si ha una singola equazione complessiva:

$$\begin{cases} \frac{d[E]}{dt} = -k_f[E][S] + k_r[ES] + k_{cat}[ES] = -k_f[E][S] + (k_r + k_{cat})[ES] \\ \frac{d[S]}{dt} = k_f[ES] - k_r[S][E] \\ \frac{d[ES]}{dt} = k_f[E][S] - k_r[ES] - k_{cat}[ES] = k_f[E][S] - (k_r + k_{cat})[ES] \\ \frac{d[P]}{dt} = k_{cat}[ES] \end{cases}$$

Bisogna analizzare un secondo anche i vari k_i . Nell'approccio deterministico, l'unità di misura della costante di qualsiasi equazione differenziale dipende dall'ordine della corrispondente reazione biochimica, ovvero dal numero di molecole reagenti. Si ha quindi, limitandosi ai primi ordini di reazione:

Ordine della reazione	Unità	Unità equivalente con $M = \frac{mol}{L}$
1	$\frac{1}{s}$	
2	$\frac{L}{mol \cdot s}$	$\frac{1}{M \cdot s}$
3	$\frac{L^2}{mol^2 \cdot s^2}$	$\frac{1}{M^2 \cdot s^2}$

Questo non vale con l'*approccio stocastico* dove si studia il numero di molecole e non le concentrazioni delle stesse.

I valori di k_i sono raramente disponibili in letteratura, non avendo molta utilità calcolarli in wet-lab.

Avendo dato quindi un primo sguardo anche all'uso delle equazioni differenziali possiamo identificare un ulteriore **pro** dei *modelli reaction-based*, ovvero che forniscono una descrizione dettagliata e accurata delle interazioni molecolari e dei meccanismi di controllo (inclusi feedback o regolazione feedforward) che avvengono nei processi cellulari, prevenendo le possibilità del cosiddetto *hard-wire* del comportamento del sistema, ovvero impedendo di modellare il comportamento del sistema in fase di specifica dello stesso. Potendo modellare solo le reazioni non posso modellare tramite formalismi matematici altri comportamenti, avendo quindi che i comportamenti del sistema saranno solo emergenti dalla simulazione dello stesso. Ad esempio, se si sa che la concentrazione di una specie oscilla nel tempo, e si formalizza l'equazione differenziale di quella specie come funzione sinusoidale (ad esempio con una funzione del tipo $f(x) = \alpha \sin(x) + \beta - \dots$, si otterranno oscillazioni grazie a questa funzione sinusoidale e ma queste non saranno ottenibili come proprietà emergente del sistema stesso. Per di più un'assunzione di tale tipo, che "forza" il sistema, potrebbe essere errata.

Si ha quindi un ulteriore **pro**, in quanto poiché tutte le specie molecolari e le loro reazioni reciproche appaiono nel modello come "entità atomiche", possono essere analizzate indipendentemente l'una dall'altra o in combinazione con altri componenti, al fine di determinare la corrispondente influenza sul comportamento del sistema. Con questo tipo di modellazione, basata solo sulle reazioni (ed eventualmente sulle *EDO* generate automaticamente da esse), si evita l'uso di funzioni cinetiche approssimative, come la **legge di frequenza di Michaelis-Menten** per i processi enzimatici, o le **funzioni di Hill** per il legame cooperativo, che sono spesso sfruttate nelle *EDO* ma

impediscono la possibilità di associare l'effetto della perturbazione dei singoli componenti sul sistema complessivo il comportamento. Questo è quindi un ulteriore **pro**.

Un altro **pro** ancora è quello che, contrariamente ad altri formalismi, un *modello reaction-based* può essere facilmente perfezionato o esteso senza alcun aggiustamento laborioso nella formalizzazione della versione precedente del modello, basta infatti aggiungere la reazione. Se si partisse da un sistema modellato a priori con delle *EDO* (che non sono generate automaticamente) si avrebbe che l'aggiunta di nuove specie o di nuove reazioni richiederebbe la modifica di molte delle sue equazioni differenziali, avendone una per specie che "contiene" tutte le informazioni sulle interazioni con essa. La *modellazione reaction-based* è adatta quindi alla costruzione modulare di modelli sempre più grandi, per cui un nucleo iniziale di specie e reazioni può essere esteso per tenere conto di altri processi.

Legge di Michaelis-Menten

Per approfondire quanto detto si introduce brevemente la **legge di Michaelis-Menten**.

Questa legge ha valenza sse si trova nella cosiddetta **approssimazione allo stato quasi stazionario (*quasi-steady-state approximation*)**, che, riprendendo l'esempio 13, sarebbe:

$$k_f[E][S] = k_r[ES] + k_{cat}[ES]$$

Qualora questa approssimazione sia valida si può scrivere la vera e propria *legge di Michaelis-Menten*:

$$\frac{dP}{dt} = [\dot{P}] = \frac{V_{max}[S]}{k_M + [S]}$$

dove:

- $V_{max}[S] = k_{cat}[E_{tot}]$
- $k_M = \frac{k_r + k_{cat}}{k_f}$

Si ha quindi una sola equazione dipendente da k_M e V_{max} che posso perturbare senza però sapere da cosa poi dipenda, nel dettaglio rispetto alla reazione iniziale, una variazione del risultato. Avendo una sola equazione si parla di **modelli ridotti**, che sono comodi per alcuni aspetti ma che impediscono perturbazioni mirate, in quanto le varie componenti vengono "nascoste" da questa singola equazione.

Conti extra e dimostrazione dell'ottenimento della legge negli appunti del corso di *Data & Computational Biology*.

Equazione di Hill

Vediamo quindi anche un breve approfondimento sull'**equazione di Hill**. Questa equazione descrive il cosiddetto **legame/binding cooperativo** che consiste nel rappresentare il fenomeno per cui il legame di un ligando con una macromolecola è talvolta “potenziato” se altri ligandi sono già presenti sulla stessa macromolecola.

Si ha quindi la vera e propria equazione:

$$\sigma = \frac{[L]^n}{K_A^n + [L]^n}$$

dove:

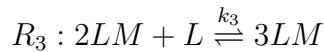
- σ è la frazione della concentrazione della macromolecola legata al ligando
- $[L]$ è la concentrazione del ligando libero, quindi non ancora legato
- n è il **coefficiente di Hill** che quantifica il grado di interazione tra i siti di legame del ligando in due modi:
 1. *legame/binding cooperativo positivo* se $n > 1$
 2. *legame/binding cooperativo negativo* se $n < 1$
 3. se si avesse $n = 1$ non si avrebbero legami che cooperano
- K_A è la **costante di dissociazione**

Si può descrivere questo tipo di interazione molecolare per mezzo di reazioni biochimiche. Definendo:

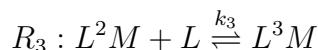
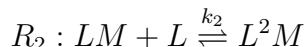
- L come il ligando
- M come la macromolecola
- un *legame cooperativo positivo*

posso rappresentare lo stesso comportamento con le seguenti reazioni biochimiche, limitate al secondo ordine (aumentando l'ordine sarebbero meno):





Avrei potuto scrivere anche con la seguente notazione, ma sempre limitando al secondo ordine:



Imponendo $k_1 > k_2 > k_3$ (ignorando per semplicità i valori numerici) defino come si “favorisce” ogni volta l’attacco di un’altra L , definendo così il *legame/binding cooperativo*. Il punto chiave è appunto lavorare sulle costanti.

Informazioni aggiuntive sull’equazione negli appunti del corso di *Data & Computational Biology*.

5.1.4 Esempio del Pathway RAS/CAMP/PKA

Si vede ora un esempio esteso e “reale” di modellazione del **pathway RAS/-CAMP/PKA**, che è un *signal trasduction pathway*, mediante un *modello reaction-based*. Questo esempio permetterà di confrontare “sul campo” un modello su cui effettuare sia *simulazioni stocastiche* che *simulazioni deterministiche*, permettendo anche di introdurre la tecnica del **parameter sweeping**.

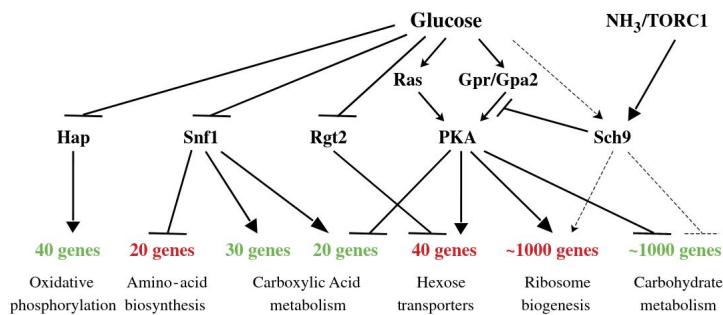
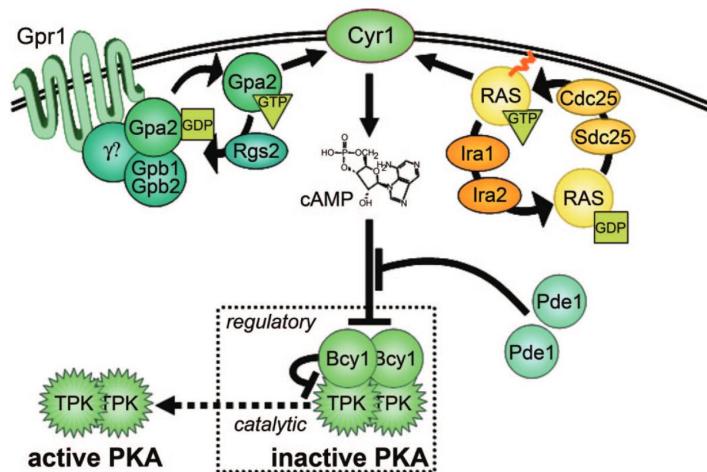
Nel dettaglio si ha a che fare con il **glucose signaling** nel lievito, che comprende cinque pathway *interbloccati*, visibili in figura 5.4⁶ che comportano enormi cambiamenti trascrizionali. Di questi cinque pathway si ha che il *pathway RAS/CAMP/PKA* è quello principale, in quanto coinvolge circa 2000 geni. Inoltre tale pathway ha un ruolo centrale:

- nella regolazione del metabolismo
- nella resistenza allo stress
- nella progressione del ciclo cellulare

Possiamo inoltre vedere una raffigurazione più dettagliata in figura 5.5⁷. In alto a destra notiamo come *RAS* venga attivato da *GTP* e disattivato da

⁶Zaman, Shadia, et al. "Glucose regulates transcription in yeast through a network of signaling pathways." Molecular systems biology 5.1 (2009): 245.

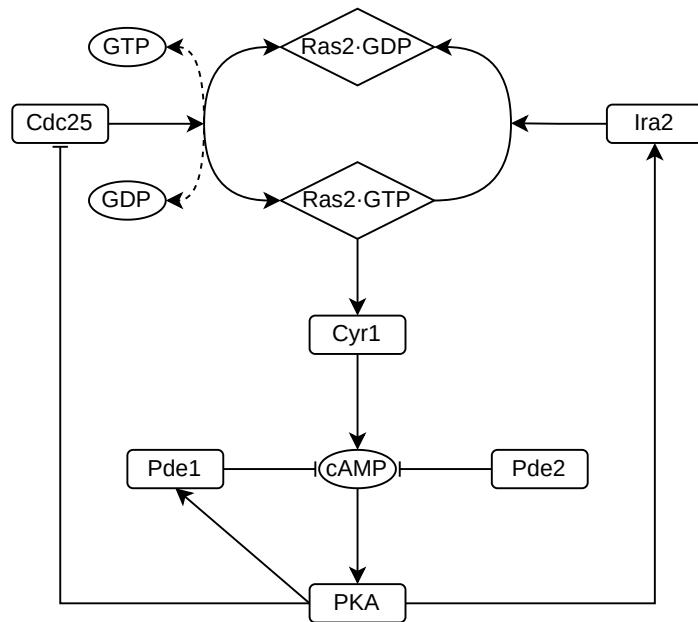
⁷Santangelo GM. Glucose signaling in *Saccharomyces cerevisiae*. Microbiol Mol Biol Rev. 2006;70(1):253-282. doi:10.1128/MMBR.70.1.253-282.2006

Figura 5.4: Schema rassuntivo dei pathway legati al *glucose signaling*.Figura 5.5: Raffigurazione più dettagliata del *pathway RAS/CAMP/PKA*, che consiste nella zona in alto a destra, in centro e in basso dell'immagine.

GDP. Si hanno inoltre *Ira1* e *Ira2* che inducono l'inattivazione di *RAS* mentre *Cdc25* e *Sdc25* inducono l'attivazione. L'attivazione di *RAS* induce quindi *Cyr1* che comporta *cAMP*. Con quattro *cAMP* si ha il rilascio dei domini catalitici di *PKA* inattivo, rilasciando due *PKA*, formati appunto da *TPK*, scindendola dai domini regolatori formati da due *Bcy1*. Si rilascia quindi *PKA* attivo. Si hanno inoltre *Pde1* e *Pde2* (tipo nell'immagine) che sono i *degradatori* di *cAMP*, senza in quali il pathway sarebbe sempre attivo. Per completezza, in alto a sinistra, si ha un comportamento simile a quanto si ha con *RAS*, avendo in più la rappresentazione dei recettori transmembrana (*Gpr1*).

Possiamo inoltre produrre uno schema più “pulito” molto utile per avere un’immediata trascrizione dello stesso in un insieme di reazioni.

RAS viene indicato con Ras2 mentre con · si specificano i composti⁸:



Per quanto all'apparenza semplice questo modello non è scontato da studiare. Inoltre non vengono rappresentate componenti che sono date "per scontate" in questo tipo di modello biologico, come l'ATP. Facendo quindi un piccolo "recap" si hanno:

- la proteina *Ras2*, con *GTP*, che viene regolata positivamente da *Cdc25* e regolata negativamente da *Ira2*
 - *Ras2·GTP* che attiva le *attiva le adenilato ciclasi Cry1*
 - *Cyr1* che induce la sintesi di *cAMP*
 - *cAMP* che attiva *PKA*
 - *cAMP* che viene degradato da due *fosfodiesterasi*, ovvero *Pde1* e *Pde2*
 - *PKA* che esercita tre *feedback*:

⁸Besozzi D, Cazzaniga P, Pescini D, Mauri G, Colombo S, Martegani E. The role of feedback control mechanisms on the establishment of oscillatory regimes in the Ras/-cAMP/PKA pathway in *S. cerevisiae*. EURASIP J Bioinform Syst Biol. 2012;2012(1):10. Published 2012 Jul 20. doi:10.1186/1687-4153-2012-10

1. una regolazione positiva su *Pde1*, che a livello del sistema complessivo risulta essere un *feedback negativo* in quanto *Pde1* inattiva *cAMP*
2. una regolazione positiva su *Ira2*, che a livello del sistema complessivo risulta essere un *feedback negativo* in quanto si ha regolazione negativa su *Ras2*
3. una regolazione negativa su *Cdc25*

Nel complesso si può quindi parlare di **totally non linear system**.

Questo progetto di modellazione è stato proposto al laboratorio della professoressa Besozzi dal professor Martegani e si poneva varie questioni su questo pathway alla luce di conoscenze pregresse scarne sullo stesso, in quanto consistevano praticamente nella solo evidenza indiretta di oscillazioni, come in figura 5.6⁹, nelle cellule del lievito come *spostamento nucleo-citoplasmatico del fattore di trascrizione Msn2* (bersaglio finale di *PKA*), in accordo allo *stato di fosforilazione*. In altri termini ci si attendeva un andamento oscillatorio ma senza nemmeno saperne le vere cause, non sapendo cosa accadesse di preciso all'interno del pathway stesso. Si possono quindi elencare le principali questioni scientifiche che si hanno dietro questo modello:

1. quali sono le condizioni che assicurano la presenza di oscillazioni in questo pathway?
2. qual è il ruolo svolto dai modulatori *Ras2* (*Cdc25/Ira2* e *GTP/GDP*) sull'insorgere dei regimi oscillatori?
3. il *rumore biologico* (discorso che verrà anche approfondito nel corso) ha un ruolo rilevante in questo pathway?

⁹Medvedik, Oliver, et al. "MSN2 and MSN4 link calorie restriction and TOR to sirtuin-mediated lifespan extension in *Saccharomyces cerevisiae*." PLoS biology 5.10 (2007): e261.

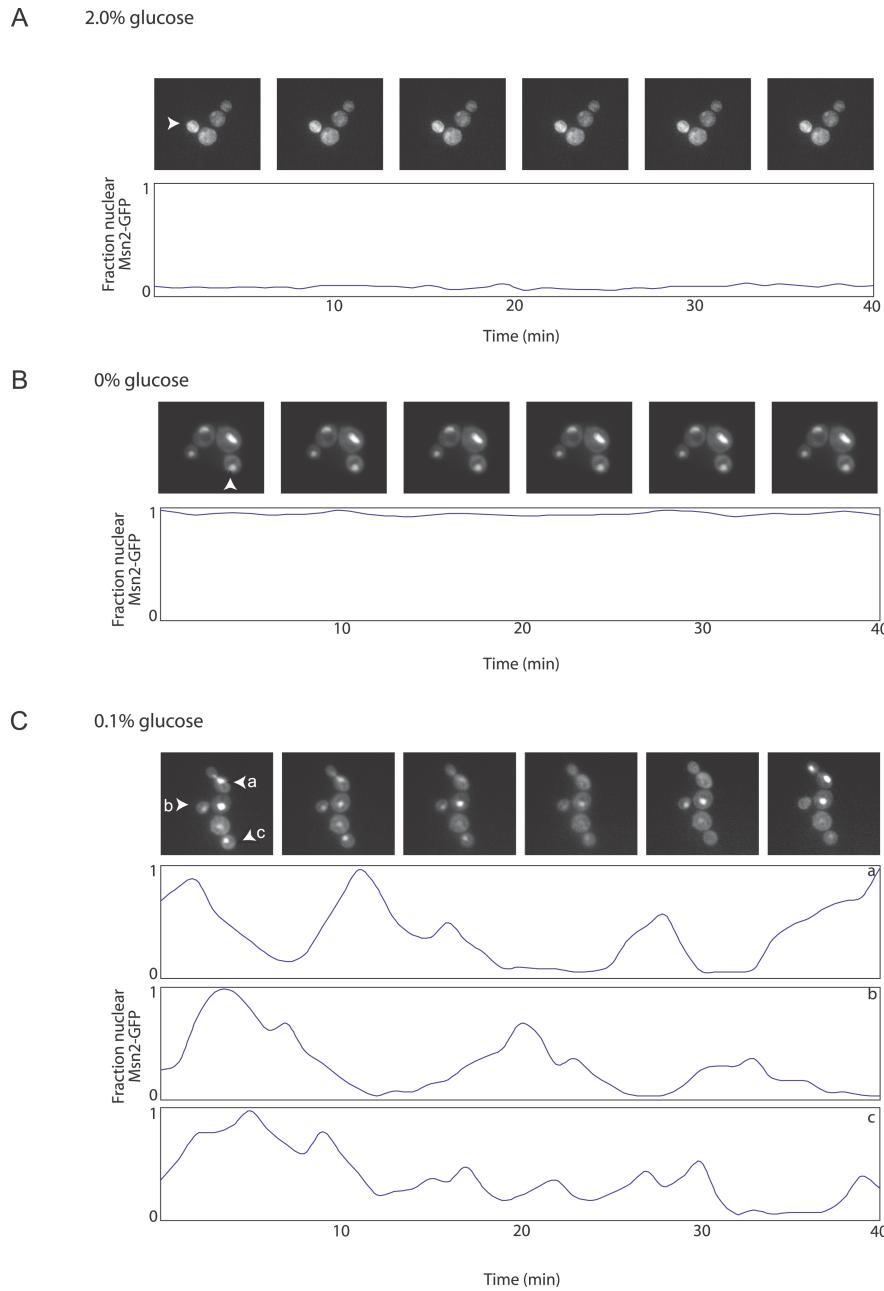


Figura 5.6: Immagini e grafici che mostrano i risultati sperimentali, ottenuti in *wet-lab*, per l'andamento oscillatorio atteso del *pathway RAS/CAMP/P-KA*.

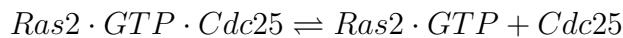
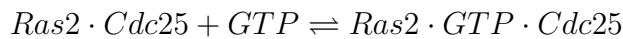
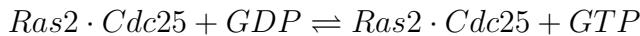
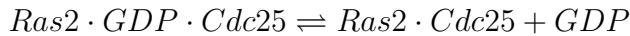
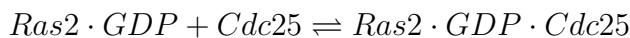
Si è quindi arrivati ad un modello con **33 specie** (composti inclusi) e **39 reazioni**¹⁰ (con conseguenti costanti):

No.	Reagents	Products	Constant c_i
r_1	Ras2-GDP + Cdc25	Ras2-GDP-Cdc25	1.0
r_2	Ras2-GDP-Cdc25	Ras2-GDP + Cdc25	1.0
r_3	Ras2-GDP-Cdc25	Ras2-Cdc25 + GDP	1.5
r_4	Ras2-Cdc25 + GDP	Ras2-GDP-Cdc25	1.0
r_5	Ras2-Cdc25 + GTP	Ras2-GTP-Cdc25	1.0
r_6	Ras2-GTP-Cdc25	Ras2-Cdc25 + GTP	1.0
r_7	Ras2-GTP-Cdc25	Ras2-GTP + Cdc25	1.0
r_8	Ras2-GTP + Cdc25	Ras2-GTP-Cdc25	1.0
r_9	Ras2-GTP + Ira2	Ras2-GTP-Ira2	$*1.0 \times 10^{-2}$
r_{10}	Ras2-GTP-Ira2	Ras2-GDP + Ira2	$*2.5 \times 10^{-1}$
r_{11}	Ras2-GTP + Cyr1	Ras2-GTP-Cyr1	1.0×10^{-3}
r_{12}	Ras2-GTP-Cyr1 + ATP	Ras2-GTP-Cyr1 + cAMP	2.1×10^{-6}
r_{13}	Ras2-GTP-Cyr1 + Ira2	Ras2-GDP + Cyr1 + Ira2	1.0×10^{-3}
r_{14}	cAMP + PKA	cAMP-PKA	1.0×10^{-5}
r_{15}	cAMP + cAMP-PKA	(2cAMP)-PKA	1.0×10^{-5}
r_{16}	cAMP + (2cAMP)-PKA	(3cAMP)-PKA	1.0×10^{-5}
r_{17}	cAMP + (3cAMP)-PKA	(4cAMP)-PKA	1.0×10^{-5}
r_{18}	(4cAMP)-PKA	cAMP + (3cAMP)-PKA	1.0×10^{-1}
r_{19}	(3cAMP)-PKA	cAMP + (2cAMP)-PKA	1.0×10^{-1}
r_{20}	(2cAMP)-PKA	cAMP + cAMP-PKA	1.0×10^{-1}
r_{21}	cAMP-PKA	cAMP + PKA	1.0×10^{-1}
r_{22}	(4cAMP)-PKA	C + C + R-2cAMP + R-2cAMP	1.0
r_{23}	R-2cAMP	R + cAMP + cAMP	1.0
r_{24}	R + C	R-C	7.5×10^{-1}
r_{25}	R-C + R-C	PKA	1.0
r_{26}	C + Pde1	C + Pde1p	1.0×10^{-6}
r_{27}	cAMP + Pde1p	cAMP-Pde1p	1.0×10^{-1}
r_{28}	cAMP-Pde1p	cAMP + Pde1p	1.0×10^{-1}
r_{29}	cAMP-Pde1p	AMP + Pde1p	7.5
r_{30}	Pde1p + PPA2	Pde1 + PPA2	1.0×10^{-4}
r_{31}	cAMP + Pde2	cAMP-Pde2	1.0×10^{-4}
r_{32}	cAMP-Pde2	cAMP + Pde2	1.0
r_{33}	cAMP-Pde2	AMP + Pde2	1.7
r_{34}	C + Cdc25	C + Cdc25p	1.0
r_{35}	Cdc25p + PPA2	Cdc25 + PPA2	1.0×10^{-2}
r_{36}	Ira2 + C	Ira2p + C	1.0×10^{-3}
r_{37}	Ras2-GTP + Ira2p	Ras2-GTP-Ira2p	1.25
r_{38}	Ras2-GTP-Ira2p	Ras2-GDP + Ira2p	2.5
r_{39}	Ira2p	Ira2	10.0

¹⁰Besozzi D, Cazzaniga P, Pescini D, Mauri G, Colombo S, Martegani E. The role of feedback control mechanisms on the establishment of oscillatory regimes in the Ras/-cAMP/PKA pathway in *S. cerevisiae*. EURASIP J Bioinform Syst Biol. 2012;2012(1):10. Published 2012 Jul 20. doi:10.1186/1687-4153-2012-10

Nel dettaglio:

- le reazioni dalla 1 alla 10, inclusa, rappresentano lo *switch cycle* di *Ras2*, quindi tutta la parte del superiore del modello (*Cdc23*, *GTP*, *GDP*, *Ras2·GDP*, *Ras2·GTP* e *Ira2*), ovvero (*si noti che tutte queste reazioni potrebbero essere scritte in una riga ma per praticità sono state mandate a capo ripetendo il prodotto della reazione i nel reagente della reazione $i+1$* ; le costanti delle reazioni non sono indicate):



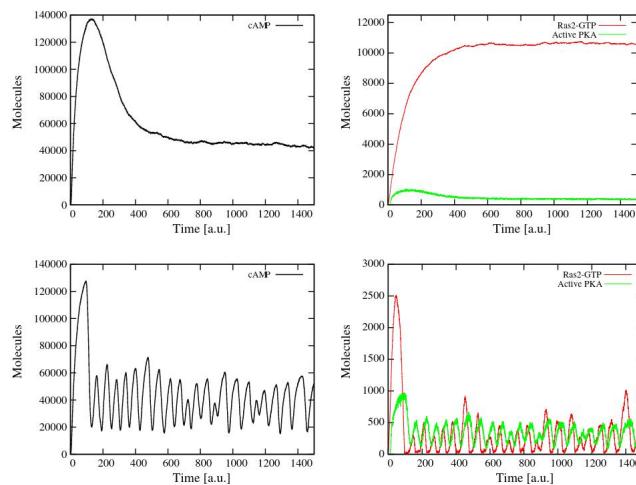
- le reazioni dalla 11 alla 13, inclusa, rappresentano la *sintesi di cAMP*, quindi la parte centrale del modello (*Ras2·GTP*, *Cyr1*, *cAMP*, con l'aggiunta di *ATP*)
- le reazioni dalla 14 alla 25, inclusa, rappresentano l'*attivazione di PKA*, quindi la parte centrale/inferiore del modello (*cAMP*, *PKA*). Si noti che per il discorso di mantenere reazioni al più del secondo ordine le varie reazioni che implicano quattro *cAMP* sono state spezzate. Inoltre si hanno:
 - *C* come la *catalytic subunit*, ovvero i due *TPK* di *PKA* inattivo
 - *R* come la *regulatory subunit*, ovvero i due *Bcy1* di *PKA* inattivo
- le reazioni dalla 26 alla 33, inclusa, rappresentano l'*attività delle fosfodiesterasi*, quindi la parte inferiore del modello (*cAMP*, *PKA*, *Pde1*, *Pde2*). Si noti l'aggiunta di *PPA2* per la defosforilazione di *Pde1*
- le reazioni 34 e 35 rappresentano il *feedback su Cdc25*, quindi la parte sinistra del modello (*PKA* e *Cdc25*, con l'aggiunta di *C* e *PPA2*). Si noti come, a livello di modello, basti creare *Cdc25^P*,

dove P sta per fosforilato (ed è presente anche in altre reazioni), per rappresentare eventuali inibizioni, in quanto si sta creando una variabile aggiuntiva, che quindi non “interagisce” come la “vecchia” variabile per $Cdc25$ (avendo che $Cdc25 \neq Cdc25^P$). Si sta quindi modellando in modo semplice i *feedback negativi* senza fare *hard wiring*

- le reazioni dalla 36 alla 39, inclusa, rappresentano il *feedback su Ira2*, quindi la parte destra del modello (PKA e $Ira2$, con l’aggiunta di C). In questo caso si noti come la differenza della costante delle reazioni, avendo una costante maggiore in presenza di $Ira2$ non fosforilata, permetta la modellazione del *feedback negativo*

Lo *stato iniziale* prevede di avere un “sistema spento”, avendo $cAMP$ con concentrazione nulla. Per le restanti componenti si segnala come le quantità iniziali siano state scelte anche per rappresentare una sorta di “range stocastico”, atto a rappresentare la realtà biologica. Inoltre si noti che tutti i composti hanno concentrazione nulla all’inizio, in quanto, in caso contrario, non si sarebbero potuti studiare alcuni comportamenti emergenti.

Vediamo quindi il risultato di una prima simulazione, dove sopra si hanno i comportamenti dinamici di $cAMP$, $Ras2\text{-GTP}$ e PKA attivo avendo un feedback solo su $Cdc25$ (avendo quindi posto a 0 la costante per $Ira2$), mentre sotto avendo entrambi i feedback attivi (sia su $Cdc25$ che su $Ira2$)¹¹:



¹¹Besozzi D, Cazzaniga P, Pescini D, Mauri G, Colombo S, Martegani E. The role of feedback control mechanisms on the establishment of oscillatory regimes in the Ras/-cAMP/PKA pathway in *S. cerevisiae*. EURASIP J Bioinform Syst Biol. 2012;2012(1):10. Published 2012 Jul 20. doi:10.1186/1687-4153-2012-10

Si nota come queste siano *simulazioni stocastiche* (infatti le curve non sono “precise”). Si nota come nel primo caso si raggiunga una sorta di *steady state stabile*. Nel primo caso, inoltre, non si ha un comportamento oscillatorio (e si avrebbe lo stesso comportamento anche togliendo il feedback di *Cdc25*).

Si ha un grafico per ogni singola componente del sistema.

5.1.5 Parameter Sweep Analysis

Possiamo sfruttare l'esempio del *pathway RAS/CAMP/PKA* anche per introdurre la tecnica detta **Parameter Sweep Analysis (PSA)**.

Con questa tecnica si ha l'analisi automatica dell'effetto di un insieme di diverse condizioni iniziali, attraverso la variazione sistematica di un dato parametro entro un range fisso (rispetto al valore di riferimento). Nel caso dell'esempio quindi si variano le concentrazioni iniziali e/o le costanti delle reazioni. Possiamo dividere le tecniche di *PSA* in due categorie:

1. **PSA a singolo parametro (*PSA-1D*)**, che a sua volta prevede due tecniche di sampling:

- *sampling lineare*, usato quando il parametro è la concentrazione di una componente (solitamente non molto piccole) in quanto, basandosi su una distribuzione uniforme, qualora si abbia a che fare con valori prossimi allo zero, si ottengono solo valori “schiaffiati” verso lo zero, ottenendo quindi valori tendenzialmente in un subset del range di valori voluto e quindi non sarebbe utilizzabile per le costanti delle reazioni
- *sampling logaritmico*, usato quando il parametro è la costante di una reazione (solitamente molto piccole) in quanto, basandosi su una distribuzione uniforme logaritmica, copre l'intero range dei valori.

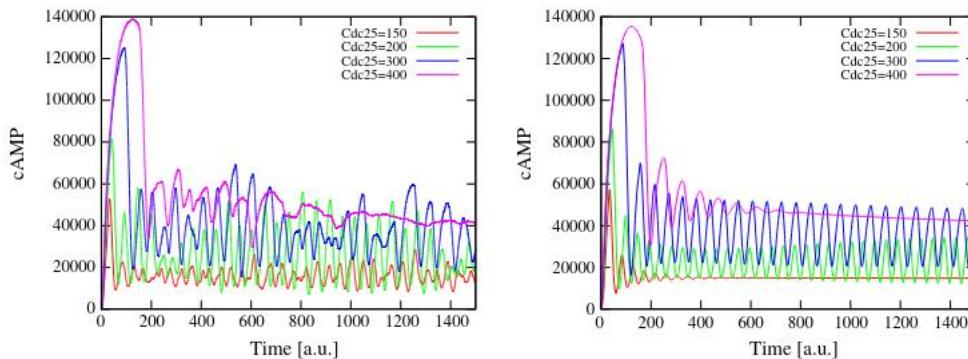
2. **PSA a doppio parametro (*PSA-2D*)**

Inoltre la generazione dei samples può essere fatta in due modi:

1. usare un **Pseudo-Random Number Generator (PRNG)**, come ad esempio il *Mersenne Twister*, che però rischia di produrre sample non equamente distribuiti sullo spazio di ricerca, avendo quindi un *basso grado di equidistribuzione*
2. usare un **low-discrepancy series method**, come la *sequenza di Sobol*, che genera numeri che sono meglio equidistribuiti rispetto

ai numeri pseudocasuali in un dato spazio di ricerca, avendo una distribuzione migliore rispetto ai numeri generati da un *PRNG* anche per valori piccoli, avendo quindi un *alto grado di equidistribuzione*. Tali metodi dovrebbero permettere comunque l'uso di un *seed* per generare sequenze uguali

Riprendendo quindi l'esempio del *pathway RAS/cAMP/PKA* possiamo vedere un esempio di perturbazione su *Cdc25*, variandone la concentrazione nel range [125, 400], per quattro sole casistiche per semplicità, studiando la variazione del comportamento oscillatorio di *cAMP*. Si hanno quindi a sinistra la *simulazione stocastica* mentre a destra la *simulazione deterministica*¹²:



Si notano varie cose:

- sia all'aumentare che al diminuire di *Cdc25* si perde il comportamento oscillatorio
- alcuni comportamenti oscillatori della simulazione stocastica, come per la curva rossa relativa a *Cdc25* = 150 sono dovuti unicamente al rumore stocastico, e questo è verificabile nella simulazione deterministica dove è una linea piatta. L'uso della simulazione deterministica ha comunque poco spazio, non essendo realistica rispetto alla natura, ma permette, ad esempio, di vedere come, per *Time* = 800 la linea viola relativa a *Cdc25* = 400 si smorzi (comportamento non facilmente individuabile nella simulazione stocastica), avendo quindi una visualizzazione più facile di variazioni del comportamento qualitativo del sistema

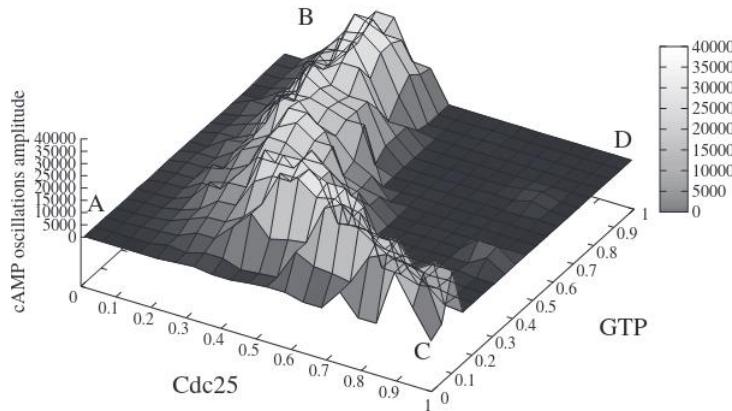
¹²Besozzi D, Cazzaniga P, Pescini D, Mauri G, Colombo S, Martegani E. The role of feedback control mechanisms on the establishment of oscillatory regimes in the Ras-/cAMP/PKA pathway in *S. cerevisiae*. EURASIP J Bioinform Syst Biol. 2012;2012(1):10. Published 2012 Jul 20. doi:10.1186/1687-4153-2012-10

Usando quindi la *PSA-1D* e studiando l'ampiezza delle oscillazioni si è scoperto che il comportamento oscillatorio è mantenuto per:

$$150 < Cdc25 < 400$$

avendo per valori minori fluttuazioni stocastiche attorno a uno stato stazionario stabile mentre per valori maggiori oscillazioni smorzate e poi uno stato stazionario. Si è quindi ottenuta risposta ad uno dei quesiti iniziali.

Con questo tipo di simulazioni si studiano quindi molti comportamenti risparmiando soldi e fatica in *wet-lab*. Facendo varie perturbazioni si può scoprire ad esempio come le variazioni sulla concentrazione di *GTP* non sortiscano alcun effetto sul comportamento oscillatorio di *cAMP*, con $Cdc25 = 300$ (condizione standard) mentre solo un piccolo range di quantità di *GTP* mantiene il comportamento oscillatorio di *cAMP*, con $Cdc25 = 500$ (overespressione). Parlando invece di *PSA-2D* si possono studiare, per esempio, contemporaneamente come le variazioni su *GTP*, avendo $GTP \in [1.9 \times 10^4, 5 \times 10^6]$ (ovvero da nutrienti ridotti a crescita normale), e *Cdc25*, avendo $GDC25 \in [0, 600]$ (ovvero dalla delezione al *2-fold expression*), variano il comportamento oscillatorio di *cAMP*. Si ottiene un grafico 3D del tipo ¹³ (*si ignorino le scale*):



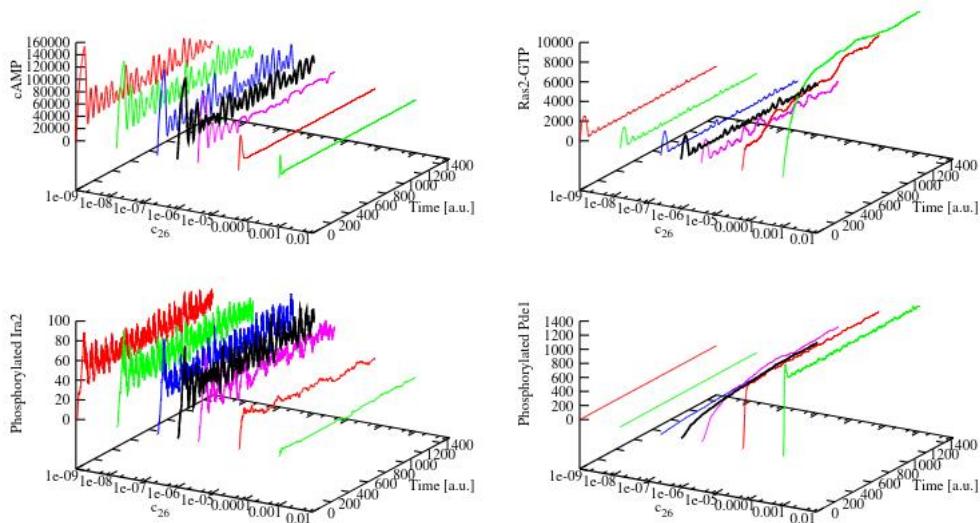
Per ottenere un tale plot si è fatto uso di *calcolo parallelo su GPU*, in quanto, per questo modello:

- su *CPU* in due ore si hanno circa 200 simulazioni

¹³Besozzi D, Cazzaniga P, Pescini D, Mauri G, Colombo S, Martegani E. The role of feedback control mechanisms on the establishment of oscillatory regimes in the Ras/cAMP/PKA pathway in *S. cerevisiae*. EURASIP J Bioinform Syst Biol. 2012;2012(1):10. Published 2012 Jul 20. doi:10.1186/1687-4153-2012-10

- su *GPU* in due ore si hanno circa 65000 simulazioni

Un esempio di *PSA-1D* su una costante di reazione si ha invece con lo studio della perturbazione della fosforilazione di *Pde1*, avendo, per la reazione 26, $c_{26} \in [1.0 \times 10^{-9}, 1.0 \times 10^{-3}]$, avendo $[1.0 \times 10^{-6}$ come valore di reference. Si sono studiate quindi le variazioni in *cAMP*, *Ras2·GTP*, *Ira2^P* e *Pde1^P*, ottenendo¹⁴:



Dove si noti come nonostante Ras2·GTP aumenti si ha un “collasso” di *cAMP*. Questo accade in quanto si ha poca concentrazione di *Ira2* quindi *Ras2* resta nello stato attivo.

Qui si coglie il vantaggio di questi modelli che permettono di studiare la dinamica di tutte le componenti.

Per concludere con questo modello:

- si è mostrata l’ enfasi sul ruolo svolto dai modulatori *Ras* (*Cdc25/Ira2* e *GTP/GDP*) e dalle fosfodiesterasi, avendo la predizione della dinamica dei sistemi in diverse condizioni, sia fisiologiche che perturbate
- per il significato biologico delle oscillazioni si è pensati all’ ipotesi del “frequency modulated” signaling system introdotta da Cai et al¹⁵

¹⁴Besozzi D, Cazzaniga P, Pescini D, Mauri G, Colombo S, Martegani E. The role of feedback control mechanisms on the establishment of oscillatory regimes in the Ras/-cAMP/PKA pathway in *S. cerevisiae*. EURASIP J Bioinform Syst Biol. 2012;2012(1):10. Published 2012 Jul 20. doi:10.1186/1687-4153-2012-10

¹⁵Cai L, Dalal CK, Elowitz MB. Frequency-modulated nuclear localization bursts coordinate gene regulation. Nature. 2008;455(7212):485-490. doi:10.1038/nature07292

- nel *pathway RAS/CAMP/PKA*, secondo il professor Martegani (???), le oscillazioni potrebbero estendere l'intervallo di regolamentazione del sistema, avendo che *PKA* è conosciuto per controllare il 90% dei geni regolati dal glucosio nel lievito

5.2 Simulazioni Deterministiche

Si introduce qui la tematica delle **simulazioni deterministiche**, che, ricordando come un *sistema reaction-based* possa essere convertito in un **sistema di equazioni differenziali ordinarie (EDO)** (tramite la *law of mass-action*), di fatto comporta il parlare di **metodi di integrazione numerica**. Si ricorda inoltre come il sistema di EDO contenga un'equazione differenziale per ogni specie dell'insieme delle specie $\mathcal{S} = \{S_1, S_2, \dots, S_n\}$, che vengono rappresentate a loro volta tramite numeri reali rappresentati la loro concentrazione. Nel dettaglio ogni equazione differenziale è della forma:

$$\frac{dS_i}{dt} = f_i(S_1, \dots, S_n)$$

dove la funzione f_i è un'arbitraria funzione matematica che include all'interno anche le costanti k_j per i *kinetic rates*. Come funzione f_i si possono anche usare funzioni che approssimano la cinetica, come le funzioni di Michaelis-Menten o di Hill, al fine di semplificare il modello e ridurre il numero di equazioni differenziali. Ovviamente per quanto ci sia $f_i(S_1, \dots, S_n)$ non implica che si abbiano tutte le S_i nell'equazione in quanto banalmente potrei avere degli $0 \cdot S_j$.

Un esempio più completo di uso delle *EDO* nel contesto della *systems biology* è visualizzabile in figura 5.7¹⁶.

Si hanno alcune proprietà nei modelli deterministic, dove appunto la soluzione del sistema di *EDO* altro non è che la dinamica, quindi la variazione, delle concentrazioni di ogni specie del modello nel tempo (ovvero il *comportamento emergente*):

- la dinamica del sistema è univocamente determinata dai valori dei parametri e dalla condizione iniziale data, avendo quindi che, a parità di parametri e condizioni iniziali, multiple simulazioni porteranno sempre allo stesso risultato, in quanto nessun evento casuale (ad esempio dovuto a collisioni molecolari) è coinvolto nella determinazione degli stati futuri del sistema

¹⁶Aldridge BB, Burke JM, Lauffenburger DA, Sorger PK. Physicochemical modelling of cell signalling pathways. Nat Cell Biol. 2006;8(11):1195-1203. doi:10.1038/ncb1497

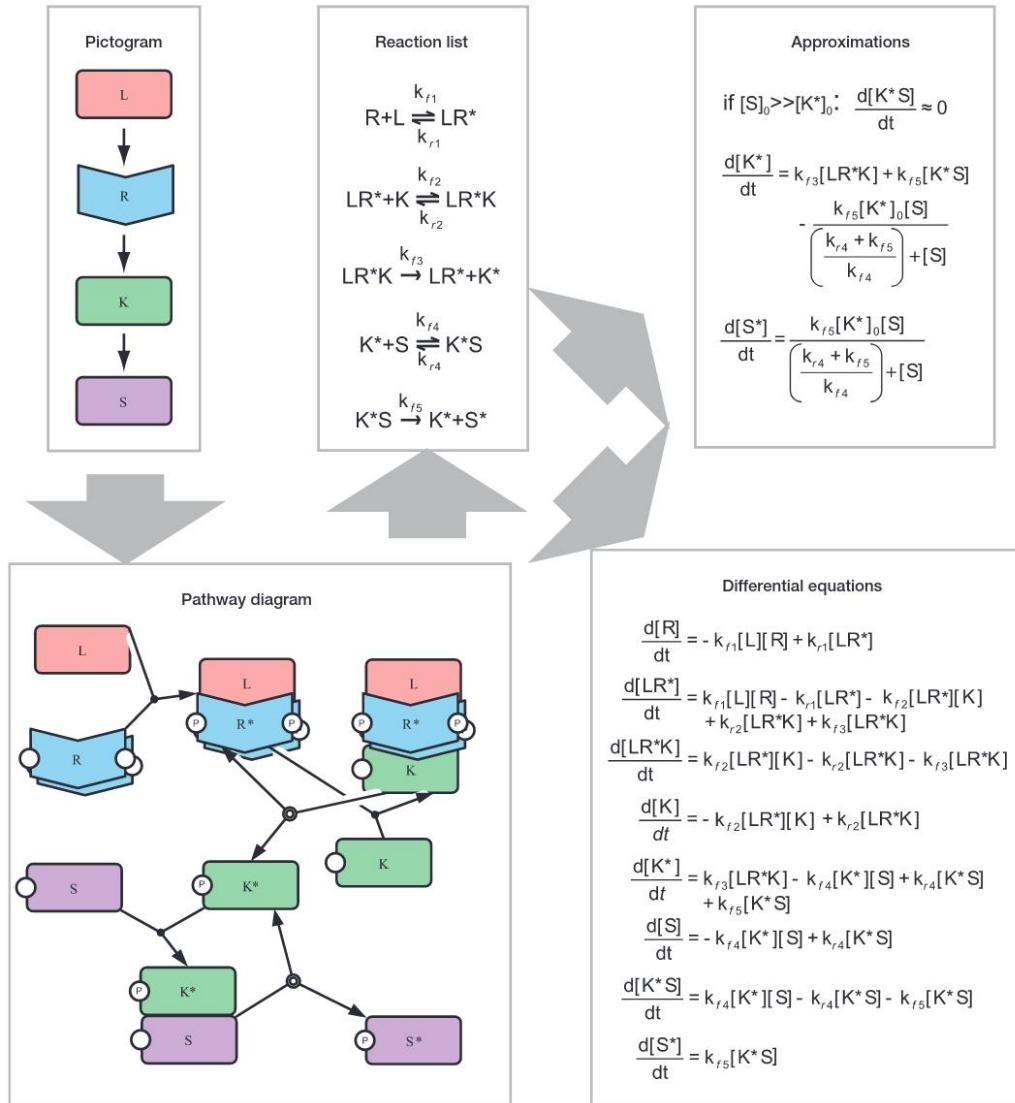


Figura 5.7: Esempio di un semplice pathway *ligando-recettore-chinasi-substrato*, con $|\mathcal{S}| = 8$. Si nota come dal modello si possano produrre sia il sistema con 8 EDO che un sistema ridotto di sole 3 EDO, tramite approssimazioni delle funzioni grazie a precise assunzioni, principalmente di stampo biochimico.

- tali modelli sono efficaci (e quindi dovrebbero essere usati solo in tal caso) quando le quantità di specie molecolari sono elevate, in modo che la casualità di ogni interazione molecolare sia “smorzata” facendo la media su un gran numero di eventi di interazione molecolare. Infatti se ogni interazione molecolare avviene molte volte, la casualità viene coperta dal comportamento medio del sistema

Parlando in generale di risolvere un *sistema di EDO* si hanno due possibili vie:

1. la **soluzione analitica**, dove la dinamica ottenuta rappresenta la soluzione esatta del sistema. Tale metodo è solitamente inapplicabile in quanto, soprattutto in *systems biology*, si lavora con sistemi troppo complessi per i quali sarebbe praticamente impossibile trovare la soluzione analitica
2. i **metodi di integrazione numerica**, dove la dinamica ottenuta rappresenta una soluzione approssimata del sistema. Si hanno moltissimi metodi, tra cui:
 - il **metodo di Eulero**
 - il **metodo di Runge-Kutta**
 - *metodi impliciti ed espliciti*
 - *metodi multi-step*
 - *metodi adattivi*
 - ...

Nel corso verranno approfonditi solo i primi due ma si avranno anche citati altri metodi tendenzialmente “più efficaci”.

5.2.1 Metodi di Integrazione Numerica

Si parla quindi di **metodi di integrazione numerica** anche se, per semplicità, si considera non un *sistema di EDO* ma bensì una singola *EDO* del primo ordine, con una data condizione iniziale. Più formalmente si studierà il caso:

$$\begin{cases} \frac{dy}{dt} = f(y(t), t) \\ y(t_0) = y_0 \end{cases}$$

Dove y è la funzione non nota che si vuole approssimare ma di cui si conosce:

- il rateo con cui cambia, ovvero $\frac{dy}{dt}$, che in pratica è la “pendenza” della funzione in un certo punto
- un suo valore iniziale, ovvero $y(t_0)$, che nel nostro caso specifico è la concentrazione iniziale

In questo contesto si vuole determinare la soluzione approssimativa in un dato intervallo di tempo $[t_0, t_n]$, considerando un insieme di $n + 1$ istanti di tempo t_0, t_1, \dots, t_n tale che:

$$t_i = t_0 + i\Delta t \text{ con } \Delta t = \frac{t_n - t_0}{n}$$

Avendo quindi che si suddivide il tempo totale in $n + 1$ istanti di tempo equi-distanziati, con una distanza pari a Δt .

Metodo di Eulero

Il metodo che si introduce ora è detto **metodo di Eulero** ed è il metodo più semplice, ma anche meno efficace, di integrazione numerica.

L'obiettivo è quindi determinare la forma di una curva incognita che soddisfi l'*EDO*, considerando l'unico valore noto, ovvero la condizione iniziale. Per ottenere ciò si può immaginare l'*EDO* come la formula con cui calcoliamo la pendenza della retta tangente alla curva, in qualsiasi punto della curva, una volta che conosciamo la posizione di quel punto, quindi, anche se la curva non è nota, utilizziamo la condizione iniziale nota e la funzione data nell'*EDO*, cioè f , per valutare la pendenza e la retta tangente alla curva alla condizione iniziale. L'intero metodo viene quindi ripetuto ad ogni successivo punto temporale, ottenendo una curva poligonale che approssima la forma della curva incognita.

Per capire come funzioni tale metodo si ricorda che la derivata di una funzione in un dato punto è la pendenza della retta tangente alla curva della funzione in quel punto, avendo quindi il cosiddetto **limite incrementatale**:

$$\frac{dy}{dt} = \lim_{\Delta t \rightarrow 0} \frac{y(t_i + \Delta t) - y(t_i)}{\Delta t}$$

Ma noi sappiamo che:

$$\frac{dy}{dt} = f(y(t), t)$$

Possiamo quindi riscrivere:

$$dy = f(y(t), t) dt$$

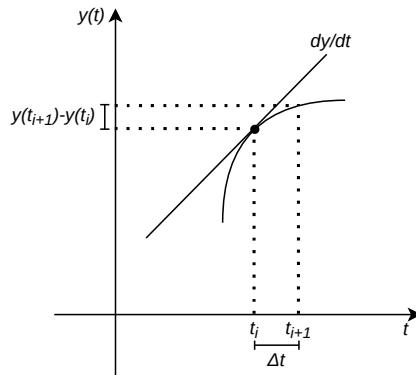
Ma quindi si può dire che:

$$y(t_{i+1}) - y(t_i) = f(y(t_i), t_i)(t_{i+1} - t_i)$$

Riscrivendo sapendo che $\Delta t = t_{i+1} - t_i$ si ottiene la formula generale del **metodo di Eulero** (dove si noti che tutti i termini a destra dell'uguale sono noti):

$$y(t_{i+1}) = y(t_i) + f(y(t_i), t_i)\Delta t, \quad \forall i = 0, 1, \dots, n-1$$

Graficamente si avrebbe:



Si ah quindi a che fare con un *algoritmo iterativo single-step*, in quanto y_{i+1} si ottiene unicamente partendo dal valore di y_i , con un solo calcolo e si itera per tutti i t_i in $[t_0, t_n]$.

Si noti che solitamente si sceglie un Δt molto minore di 1, avendo che è nell'ordine di 10^{-n} , con $n \gg 1$.

Questo metodo è comunque molto “approssimativo” e basta anche una semplice *EDO* da approssimare per mostrarne i limiti. Più formalmente infatti questo è un metodo del **primo ordine** e quindi si ha (da considerare con $\Delta t < 1$) che:

- l'*errore locale di troncamento*, ovvero quello causato da una singola iterazione, è nell'ordine di $O(\Delta t^2)$
- l'*errore globale accumulato*, ovvero l'errore cumulativo causato da molte iterazioni o l'accumulo dell'errore di troncamento locale su tutte le iterazioni, è nell'ordine di $O(\Delta t)$

Metodo Runge-Kutta del Quart'Ordine

Il secondo metodo che si mostra è il **metodo di Runge-Kutta**, nel dettaglio il **metodo di Runge-Kutta del quart'ordine (*RK4*)**.

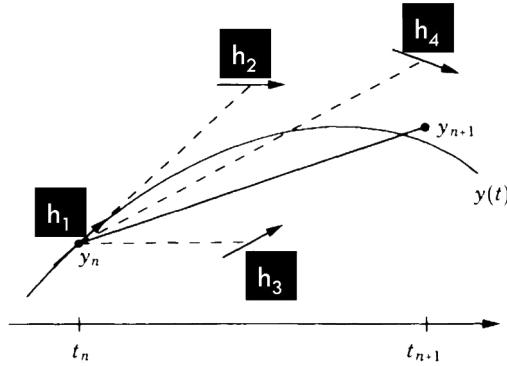


Figura 5.8: Rappresentazione grafica dei valori h_1, h_2, h_3 e h_4 .

Nel dettaglio tale metodo è un *algoritmo iterativo one-step* che, ad ogni passo di integrazione, calcola il valore $y_i + 1$ come valore corrente y_i più la media pesata di altri quattro valori (h_1, h_2, h_3, h_4) moltiplicata per Δt , che in pratica “aggiustano il tiro”. Più formalmente si ha la seguente formulazione generale:

$$y(t_{i+1}) = y(t_i) + \left(\frac{h_1}{6} + \frac{h_2}{3} + \frac{h_3}{3} + \frac{h_4}{6} \right) \Delta t, \quad \forall i = 0, 1, \dots, n-1$$

Avendo:

$$\begin{aligned} h_1 &= f(y(t_i), t_i) \\ h_2 &= f\left(y(t_i) + \frac{h_1}{2}, t_i + \frac{\Delta t}{2}\right) \\ h_3 &= f\left(y(t_i) + \frac{h_2}{2}, t_i + \frac{\Delta t}{2}\right) \\ h_4 &= f(y(t_i) + h_3, t_i + \Delta t) \end{aligned}$$

Che in pratica sono quattro pendenze, come visibile in figura 5.8, che corrispondono a quattro posizioni, specificate dalla funzione a destra dell’equazione differenziale. In pratica permettono di calcolare la pendenza media ad ogni t_i , avendo che:

- h_1 è la posizione a t_i (e infatti si noti che è quella che si usa anche nel *metodo di Eulero*)
- h_2 e h_3 sono le posizioni intermedie tra t_i e t_{i+1}
- h_4 è la posizione a t_{i+1}

Si noti che ogni h_j viene calcolata da h_{j-1} , tranne ovviamente h_1 .

Questo metodo è quindi ovviamente meno efficiente dal punto di vista computazione del *metodo di Eulero* ma comporta risultati molto più validi, avendo che è appunto un metodo del **quart'ordine**, avendo che:

- l'*errore locale di troncamento* è nell'ordine di $O(\Delta t^5)$
- l'*errore globale accumulato* è nell'ordine di $O(\Delta t^4)$

Alla luce di ciò, facendo un veloce confronto con il *metodo di Eulero* si avrebbe che, con $\Delta t = 0.001$ l'errore globale sarebbe:

- nell'ordine di 10^{-3} con il *metodo di Eulero*
- nell'ordine di 10^{-12} con il *metodo di Runge-Kutta al quart'ordine*

Si noti che esistono altri metodi di Runge-Kutta, per ordini diversi.

Metodo Adam-Moulton

Per completezza si cita anche il **metodo di Adam-Moulton** che è un algoritmo multi-step implicito che sfrutta le informazioni da diversi punti precedenti e valori derivati piuttosto che scartarli come nei metodi appena visti, che usavano solo il precedente. Nel dettaglio questo metodo sfrutta un numero fissato di step precedenti ma esistono anche metodi in grado di calcolare dinamicamente il numero di step utili.

Tra quelli visti il *metodo di Adam-Moulton* è il più accurato ma anche il più oneroso dal punto di vista computazionale.

5.2.2 Errori dei Risolutori Numerici

Se pensiamo ai metodi appena descritti si può dimostrare come il *metodo di Eulero* può divergere così tanto dalla soluzione esatta da portare addirittura a fraintendimenti, pensando magari che una funzione abbia un comportamento oscillatorio. Ogni risolutore numerico cerca di soddisfare la condizione per cui la differenza tra la soluzione approssimata e quella esatta debba rimanere in un certo *range di tolleranza*. In termini più tecnici il “massimo errore” viene impostato direttamente dall'utilizzatore del risolutore. Ci sono essenzialmente due valori da impostare:

1. la **relative tollerance (rtoi)**, per gestire l'**errore relativo**
2. l'**absolute tollerance (atoi)**, per gestire l'**errore assoluto**

Impostando quindi questi due valori si imposta l'accuratezza dell'algoritmo di integrazione numerico in qualsiasi software di simulazione (tra cui *COPASI*). Dal punto di vista più teorico si ha che errore e tolleranza sono i due lati di un'espressione di disegualanza nota come **criterio di convergenza**, che dice che un algoritmo di integrazione numerica converge se la soluzione approssimativa si avvicina alla soluzione esatta per $\lim_{\Delta t \rightarrow 0}$. Quindi una minor tolleranza, scelta alla fine mediante soglie psicologiche, genera una soluzione più accurata ma al costo di un maggior tempo macchina.

Parlando più formalmente si ha che, denotando con y_a la soluzione approssimata e con y_e la soluzione esatta, che solitamente non si conosce, si definiscono:

- l'**errore assoluto** come $|y_a - y_e|$
- l'**errore relativo** come $\frac{|y_a - y_e|}{|y_e|}$

Ricordando che solitamente y_e non è nota si procede fissando valori bassi per la tolleranza in modo da poter ragionare indirettamente sui due tipi di errore. L'*errore relativo* è generalmente una misura migliore dell'errore rispetto all'*errore assoluto*, sebbene l'*errore assoluto* sia necessario per determinare l'accuratezza della soluzione approssimativa quando ci si avvicina allo zero, avendo $|y_e|$ a denominatore. Si noti che in alcuni risolutori l'*errore relativo* è usato per controllare la qualità della soluzione approssimata riducendo la grandezza del time-step Δt .

5.2.3 Problema della Stiffness

Il **problema della Stiffness (rigidità)** è uno dei problemi di efficienza della *systems biology*. Per tale problema non esiste ancora una vera e propria definizione formale ma dipende da:

- il problema stesso
- il risolutore
- i parametri d'errore fissati

Un algoritmo è *numericamente instabile* se l'errore si propaga durante il processo iterativo, quindi la soluzione approssimativa calcolata avrà una grande deviazione dalla soluzione esatta. In pratica la soluzione approssimata non converge a quella esatta.

Un'equazione differenziale è detta **stiff** se i metodi di integrazione numerica risultano essere *instabili* a meno di scegliere dei valori di Δt molto piccoli,

portando quindi ad una “esplosione” dal punto di vista del tempo macchina. Il problema è che non si può sapere a priori se un’equazione sia *stiff* ma lo si capisce solo eseguendo l’algoritmo di integrazione numerica, vedendo che ad un certo punto si “impasta” impiegando tantissimo tempo per avanzare nelle iterazioni, avendo che l’algoritmo tenta di ridurre il Δt per restare nel range di tolleranza richiesto ma comportando tempi macchina ingestibili. Quello che succede è che si riduce moltissimo il Δt avendo la soluzione approssimata che continua ad oscillare intorno alla soluzione esatta, avendo quindi tantissimo tempo macchina per una soluzione che non è nemmeno accettabile. Fortunatamente si sono sviluppati algoritmi che sono in grado di riconoscere la *stiffness* e a procedere di conseguenza, adattando il Δt anziché ridurlo troppo.

LSODA

Uno dei risolutori in grado di gestire la *stiffness* è **LSODA**¹⁷ che è un algoritmo di integrazione numerico adattivo caratterizzato da un processo di riconoscimento automatico della *stiffness*. Questo risolutore è ormai diventato uno standard nella *systems biology* ed è basato su due algoritmi di integrazione numerica (*che non vengono approfonditi nel corso*):

1. il **metodo di Adams** per le regioni non *stiff*
2. la **backward differentiation formula (BDF)** per le regioni *stiff*

In pratica *LSODA* switcha tra i due algoritmi a seconda della situazione che viene analizzata in modo dinamico.

5.3 Simulazioni Stocastiche

Si introducono ora le **simulazioni stocastiche** che, in contesto biologico, risultano essere più utili per poter studiare comportamenti emergenti rispetto alle simulazioni deterministiche. Nel dettaglio ci si concentrerà soprattutto sui *modelli stocastici discreti* (si noti che verranno spesso indicati solo con *modelli stocastici*).

Si ricorda che, a differenza dei *modelli deterministic*, in questo caso si modella la quantità di elementi di ogni specie, tramite numeri interi, e non la

¹⁷L. Petzold, Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations, SIAM Journal on Scientific and Statistical Computing 4(1):136-148, 1983

concentrazione delle stesse.

In questo contesto si ha la già citata **well-stirred assumption**, infatti si assume che **eventi molecolari non reattivi/elastici**, che randomizzano le posizioni delle molecole, si verificano molto più frequentemente degli **eventi molecolari reattivi/anelastici**, che modificano le quantità di popolazione delle varie specie molecolari, avendo appunto che questa circostanza produce una ridistribuzione uniforme delle molecole all'interno del volume, a priori rispetto ogni collisione reattiva. In termini più pratici possiamo dire che con tali modelli si riesce a studiare cosa avviene a livello molecolare simulandone il comportamento non deterministico (o meglio “non determinitico” per le nostre conoscenze attuali di come funzionino i sistemi biologici). La formulazione stocastica si riduce alla formulazione deterministica nel limite termodinamico che si ha quando il numero di molecole di ciascuna specie e il volume del sistema si avvicinano all'infinito. Un confronto coi *modelli deterministici* e coi *modelli stocastici continui* è visibile in figura 5.9¹⁸. Inoltre, coi *modelli stocastici*, si considera il cosiddetto **rumore biologico**. Il discorso necessita di un approfondimento. Il comportamento cellulare e l'ambiente extra cellulare sono stocastici, infatti i circuiti genetici che regolano le funzioni cellulari sono soggetti a fluttuazioni stocastiche, o rumore, nei livelli delle loro componenti. La causa principale è la **fluttuazione dell'espressione genica**. In termini poco tecnici si ha che nel DNA ogni gene è presente, in termini di molecola, una volta (o al più due) e quindi si hanno pochi eventi in cui si ha il legame effettivo col promotore (non tutto nelle cellule è subito “pronto all'uso”). Con l'*espressione genica* si ha quindi un comportamento stocastico sia a livello di *mRNA* che a livello di *proteine* si ottiene l'**eterogeneità intracellulare**, infatti anche a parità di cloni cellulari essi non saranno mai nella stessa condizione in termini di *espressione genica*. Questo comportamento “randomico” è quello che permette l'*evoluzione delle specie*, in quanto è ciò che permette l'*adattamento*, infatti la stocasticità rappresenta un mezzo benefico sfruttato dagli organismi viventi per rispondere ad ambienti in continuo mutamento adattando i propri comportamenti. Si ha che i fenotipi cellulari variano tra le popolazioni isogeniche (ovvero con individui di identica costituzione genetica) e nelle singole cellule nel tempo. Le misurazioni dell'espressione genica in singole cellule hanno rivelato *burst* (*esplosioni*) stocastiche sia di *mRNA* che di *sintesi proteica* in molti organismi, portando quindi all'**eterogeneità intracellulare**. Le cellule che rispondono stocasticamente possono competere con le cellule che sono sopraffatte dai cambiamenti ambientali e questo comporta che, nelle scale temporali più lun-

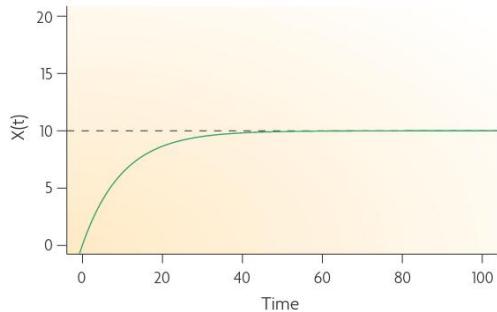
¹⁸Wilkinson, D. Stochastic modelling for quantitative description of heterogeneous biological systems. Nat Rev Genet 10, 122–133 (2009). <https://doi.org/10.1038/nrg2509>

Continuous deterministic model (RRE):

$$\frac{dX}{dt} = \alpha - \mu X$$

Solution: $X_t = \frac{\alpha}{\mu} (1 - e^{-\mu t})$

Equilibrium: $X_\infty = \alpha/\mu$



Discrete stochastic model:

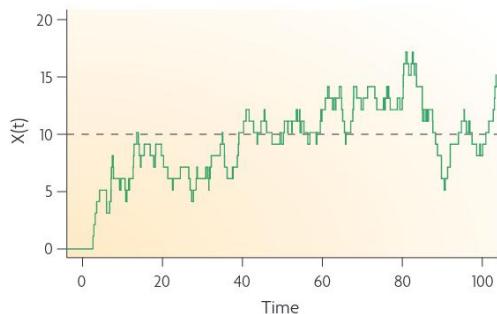
$$\Pr(X_{t+dt} = x+1 | X_t = x) = \alpha dt$$

$$\Pr(X_{t+dt} = x-1 | X_t = x) = \mu x dt$$

Solution: $X_t \sim \text{Poisson} \left(\frac{\alpha}{\mu} [1 - e^{-\mu t}] \right)$

Equilibrium distribution: $X_\infty \sim \text{Poisson} (\alpha/\mu)$

$$E(X_\infty) = \text{Var}(X_\infty) = \alpha/\mu$$



Continuous stochastic model (CLE):

$$dX_t = (\alpha - \mu X_t) dt + \sqrt{\alpha + \mu X_t} dW_t$$

At equilibrium: $E(X_\infty) = \text{Var}(X_\infty) = \alpha/\mu$

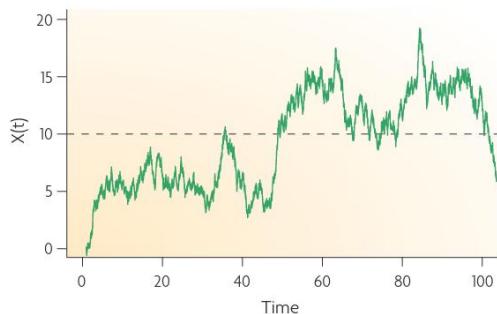


Figura 5.9: Esempio di confronto tra *modelli deterministici, stocastici discreti e stocastici continui*. Si nota come il *modello deterministico* sia più “pulito” ma anche per questo meno “utile”.

ghe, il rumore può facilitare le transizioni evolutive.

Prima di procedere bisogna introdurre alcuni concetti.

Molti processi biochimici coinvolgono un basso numero di molecole o interazioni poco frequenti, dando origine a fluttuazioni stocastiche. Si hanno quindi due concetti:

1. il **concepto di burst (esplosione)** che si ha in quanto i promotori genici possono passare stocasticamente tra gli stati “off” e “on”, provocando burst/esplosioni di produzione di mRNA. Ogni singolo RNA messaggero, inoltre, è tipicamente tradotto molte volte per produrre molte proteine, generando i corrispondenti burst di proteine
2. il **concepto di propagation** che si ha in quanto i tassi di espressione genica sono influenzati direttamente dai livelli dei fattori di trascrizione e di altri componenti a monte, che sono essi stessi soggetti a bursting e media temporale, di conseguenza, fluttuazioni nell'espressione di un gene si propagano per generare fluttuazioni nei geni a valle

Si noti che a livello di *pathway metabolici* si ha meno possibilità di burst, e in generale di comportamenti stocastici, in quanto le specie necessarie ai vari meccanismi, come l'ATP, sono presenti in grandi quantità. In tal caso si ha quindi meno *rumore biologico*.

QUESTA PRIMA PARTE VA RISENTITA

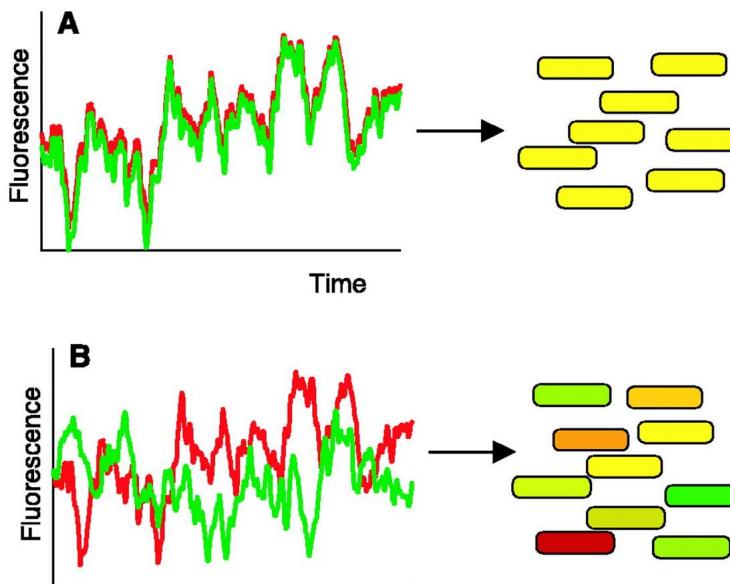
È possibile distinguere due tipi di *rumore biologico*:

1. il **rumore intrinseco** che è inherente ai processi di *espressione genica* e dipende dal fatto che si hanno effetti dati da un passo *copy-number*, parlando di un ordine di 10^0 per i geni, di 10^1 per i fattori di trascrizione e di 10^2 per le proteine. Date queste premesse il rumore si ricollega poi sia ai fenomeni di *diffusione molecolare* che di *propagazione del rumore*, non potendo essere quindi assolutamente trascurato, essendo direttamente legato all'*espressione genica*
2. il **rumore estrinseco** che è dovuto alle *fluttuazioni* in altre componenti cellulari, allo stadio del ciclo cellulare, alle caratteristiche fisiche dell'ambiente (temperatura, pressione etc...), alla distribuzione degli organelli (ad esempio dei mitocondri), al rumore “ereditato” (ad esempio durante il partizionamento nella fase di *mitosi* parlando di *divisione cellulare*). Riassumendo questo tipo di rumore regola *indirettamente* l'*espressione genica*

Elowitz, tra i più importanti ricercatori nel campo del *rumore biologico*, propose, nel 2002, una misurazione di entrambe le tipologie di rumore, usando la fluorescenza su due geni, controllati dalle stesse sequenze regolatorie:

1. *cfp*, colorato di verde
2. *yfp*, colorato di rosso

La colorazione comporta che le cellule con una quantità uguale dei due geni risulteranno colorate di giallo mentre, se esprimono più un gene che l'altro, risulteranno di conseguenza colorate come il gene maggiormente espresso. Tramite un semplice grafico si possono distinguere le due situazioni possibili¹⁹:



A in questo caso si ha assenza di *rumore intrinseco*, avendo che le due proteine fluorescenti fluttuano in modo correlato nel tempo nelle singole cellule. Si ha quindi che in una popolazione, ogni cellula avrà la stessa quantità di entrambe le proteine, sebbene tale quantità differrà da cellula a cellula a causa del *rumore estrinseco*

B in questo caso, ovviamente più vicino al caso reale, si ha la presenza di *rumore intrinseco*, avendo che l'espressione dei due geni può diventare non correlata nelle singole cellule, avendo che nella popolazione alcune cellule esprimeranno più di una proteina

¹⁹Elowitz MB, Levine AJ, Siggia ED, Swain PS. Stochastic gene expression in a single cell. *Science*. 2002;297(5584):1183-1186. doi:10.1126/science.1070919

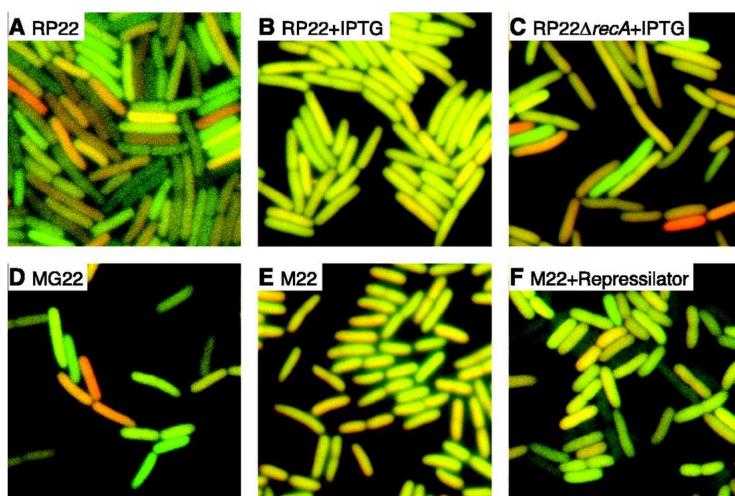


Figura 5.10: Insieme di immagini relative al rumore in *E. Coli*, studiato tramite le fluorescenze dei geni *cfp* e *yfp*. In figura A, nel ceppo *RP22*, con i promotori repressi dal gene *lacI* wild-type, il rosso e il verde indicano quantità significative di *rumore intrinseco*. In figura B si ha *RP22* cresciuto in presenza di induttore *lac*, *IPTG* 2 mM ed entrambe le proteine fluorescenti sono espresse a livelli più alti e le cellule esibiscono meno rumore. In figura C si ha una situazione analoga alla figura B tranne per il fatto che il gene *recA* è stato eliminato, aumentando il *rumore intrinseco*. In figura D si ha un altro ceppo wild-type, *MG22*, che mostra caratteristiche di rumore simili a quelle di *RP22*. In figura E si ha che livelli di espressione e il rumore nel ceppo *M22* non represso sono simili a quelli dei ceppi *inlacI+* indotti con *IPTG*. In figura F si hanno le cellule di *M22* regolate dal *Repressilator*, una rete oscillatoria che amplifica il *rumore intrinseco*.

fluorescente rispetto alle altre. Si ottiene quindi una *popolazione eterogenea*

In termini più biologici l'esperimento è stato effettuato tramite dei ceppi di *E. Coli*, come visibile in figura 5.10²⁰. Le fluttuazioni stocastiche a livello molecolare possono indurre diversi comportamenti macroscopici a livello cellulare (che possono essere la fluorescenza ma anche altro):

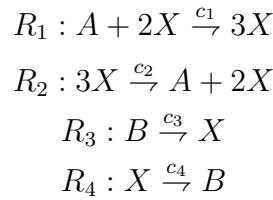
- la **bistabilità** ma anche la **multistabilità**
- ogni tipo di **fenomeno di switching**

²⁰Elowitz MB, Levine AJ, Siggia ED, Swain PS. Stochastic gene expression in a single cell. Science. 2002;297(5584):1183-1186. doi:10.1126/science.1070919

5.3.1 Il modello di Schlögl

Al fine di comprendere il concetto di *bistabilità* introduciamo il **modello di Schlögl**, che è considerato uno dei principali prototipi di sistemi bistabili. Essendo un prototipo quindi non rappresenta un reale sistema biologico ma è un comodo modello matematico.

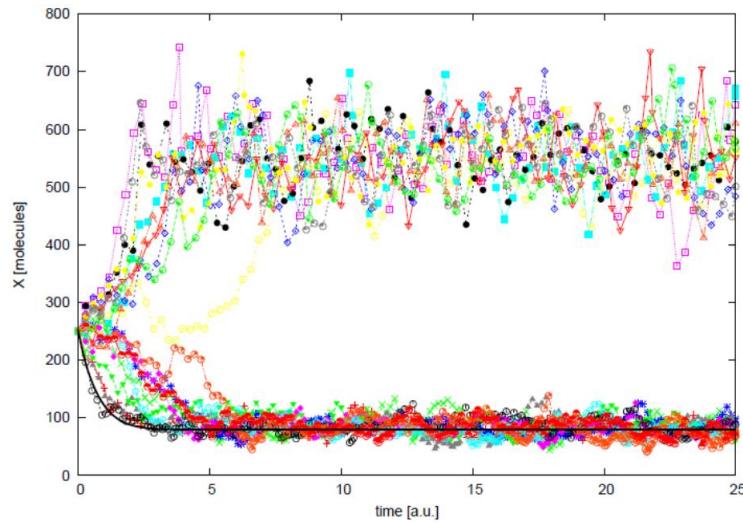
Si definiscono, date tre specie $\mathcal{S} = \{A, B, X\}$, quattro reazioni (in realtà sarebbero due reazioni reversibili):



Dove si hanno:

- le quantità di A e B che sono costanti
- la quantità di X che può raggiungere due stati stazionari, da qui il concetto di *bistabilità*

Si ha quindi che A e B dipendono da X , che risulta quindi, essendo anche l'unica specie a non avere una quantità costante, la più interessante studiare. Fissiamo quindi $X(0) = 250$ (le quantità iniziali di A e B non ci interessa specificarle qui essendo costanti) ed eseguiamo sia la simulazione deterministica (ne basta una in quanto si ha sempre lo stesso risultato) che diverse simulazioni stocastiche. Nel seguente risultato la linea nera in basso rappresenta la simulazione deterministica, mentre tutte le altre colorate sono le varie simulazioni stocastiche:

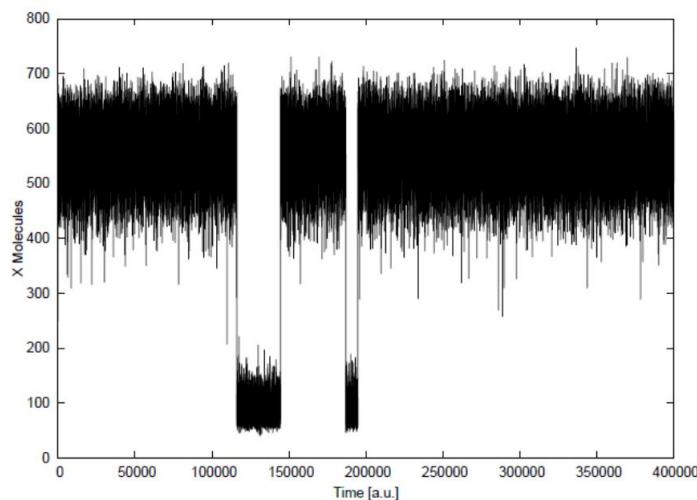


Possiamo quindi notare come le simulazioni stocastiche evidenzino la presenza di due stati stazionari, anche se ovviamente non in modo preciso:

1. in basso si hanno più risultati di simulazioni stocastiche che oscillano intorno allo steady state rappresentato dalla simulazione deterministica
2. in alto si hanno meno risultati di simulazioni stocastiche e più oscillanti. Si noti che in questo caso non si ha il risultato della simulazione deterministica in quanto essa può avere uno e un solo risultato

Si evidenzia quindi si sia potuto assistere al comportamento della bistabilità solo grazie alla simulazione stocastica, mostrando come si abbia una forte perdita d'informazione con la simulazione deterministica in quanto il *rumore biologico* può portare a diversi fenotipi, come in questo caso. Le simulazioni stocastiche sono quindi essenziali nel contesto biologico. Si sarebbe comunque potuto ottenere lo steady state superiore (quindi non quello inferiore), con la simulazione deterministica, variando $X(0)$, avendo che quindi in tali simulazioni cambia il *risultato qualitativo*.

Interessante è anche notare come, per il *modello di Schlögl*, mediante le simulazioni stocastiche, si possa ottenere anche un altro fenomeno facilmente riscontrabile in biologia, ovvero la *transizione tra stati stazionari*, come nel seguente grafico, dove si hanno diversi passaggi tra i due stati stazionari:



Questo risultato, che si noti è impossibile da ottenere con una simulazione deterministica, si ottiene da una singola simulazione stocastica effettuata su

un lungo periodo di tempo. Questo tipo di comportamento è tipico in contesti biologici in quanto, ad esempio, modella il *riadattamento cellulare* dopo una perturbazione biologica/biochimica, permettendo magari alla cellula di sopravvivere. Si può parlare in questo caso di *fenomeno di switching* (?? CHIEDERE A PROF).

5.3.2 Chemiotassi Batterica

Vediamo quindi anche un altro esempio per mostrare meglio il **fenomeno di switching**, studiando la **chemiotassi batterica** che è il fenomeno con cui i corpi cellulari, batteri ed altri organismi uni-cellulari o multi-cellulari direzionano i loro movimenti a seconda della presenza di alcune sostanze chimiche nel loro ambiente. Questo comportamento è fondamentale per la vita del batterio in quanto permette sia di scappare da ambienti tossici che di cercare ambienti ricchi di nutrienti.

I batteri si muovono infatti tramite i *flagelli* che, se si muovono tutti in modo coordinato, simultaneo e in senso anti-orario (counterclockwise, *CCW*) permettono al batterio di compiere una *run*, ovvero di andare dritto in una direzione. Questo accade, dal punto di vista biochimico sse la proteina transmembrana *CheY* fosforilata, quindi *CheYp*, non sta interagendo con le proteine che regolano il “motore” dei flagelli. Qualora invece i flagelli girano in senso orario (clockwise, *CW*) e in modo non coordinato (ne basta anche solo uno che non è coordinato con gli altri) si ha il cosiddetto *tumbling movement*, ovvero il batterio inizia a ruotare fino a scegliere una nuova direzione nella quale proseguire con una *run*. Qualora infatti sia in un ambiente passivo il batterio deve scappare da esso, tramite una lunga *run* mentre se si trova in un ambiente ricco di nutrienti, posto che prima deve muoversi per trovarlo, vuole restarci a lungo, sfruttando brevi *run* e *tumbling movement* per tornare dentro tale ambiente qualora ne sia uscito per mezzo di una *run*. Si ha quindi un **fenomeno di switching**.

Si ha quindi a che fare con un piccolo pathway che conta pochissime specie, avendo 8 proteine intracellulari nel “motore” dei flagelli più i ligandi extracellulari. Inoltre non solo si ha a che fare con poche specie ma esse sono presenti in bassa quantità, comportando così molto *rumore biologico*.

Tramite una simulazione stocastica si ottengono i seguenti risultati visibili in figura 5.3.2²¹. Ovviamente anche questo risultato non sarebbe ottenibile con una simulazione deterministica.

²¹Besozzi, D., Cazzaniga, P., Dugo, M., Pescini, D., & Mauri, G. (2009). A study on the combined interplay between stochastic fluctuations and the number of flagella in bacterial chemotaxis. arXiv preprint arXiv:0910.1415.

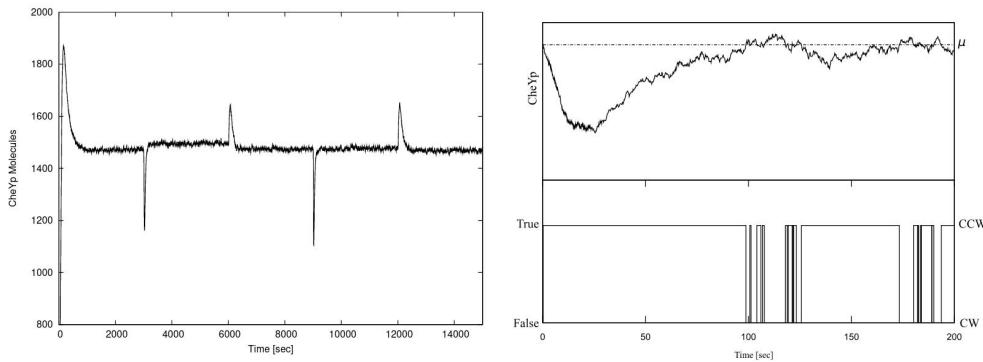


Figura 5.11: Risultati della simulazione stocastica effettuata per la chemiotassi batterica. Nel grafico a sinistra è possibile notare come i picchi siano le *run* più lunghe mentre gli steady state siano le situazioni run-tumbling. Nel grafico a destra, che è per lo più uno zoom su uno steady state, si indica con μ la media dei valori. Qualora la curva stocastica si trovi sopra tale valore μ si ha un movimento di *tumbling*, quindi con rotazione *CW* dei flagelli, come indicato nel grafico sotto, altrimenti si ha un movimento di *run*, quindi con rotazione *CCW*.

5.3.3 Altri Esempi

La lista di paper relativi all'uso di simulazioni stocastiche in contesti biologici per studiare il *rumore biologico* è pressoché infinita. Si riportano giusto alcuni titoli:

- *Functional roles for noise in genetic circuits*, di Eldar e Elowitz
- *Cellular decision making and biological noise: from microbes to mammals* di Balázsi, Van Oudenaarden e Collins
- *Noise in Gene Expression Determines Cell Fate in *Bacillus subtilis**, di Maamar, Raj e Dubnau. Questo è uno dei paper principali sul tema del *rumore biologico*, studiando l'*uptake del DNA* e la tematiche delle *competenze batteriche*
- *Nanog-dependent feedback loops regulate murine embryonic stem cell heterogeneity*, paper chiave per la tematica dell'*embryonic development*, di MacArthur et al.
- *Stochasticity of metabolism and growth at the single-cell level*, di Kiviet et al. Questo paper è interessante in quanto ricorda come bisogni considerare la stocasticità anche a livello metabolico,

anche se normalmente le **reti metaboliche** vengano modellate assumendo uno steady state del metabolismo, nonostante le reazioni metaboliche subiscano anch'esse il *rumore biologico*

5.3.4 Verso l'Algoritmo di Gillespie

Bisogna quindi vedere come simulare il *rumore biologico* per mezzo di *algoritmi stocastici*, parlando, nel dettaglio, dell'**algoritmo di Gillespie**.

Prima di parlare dell'algoritmo in se bisogna introdurre alcune basi.

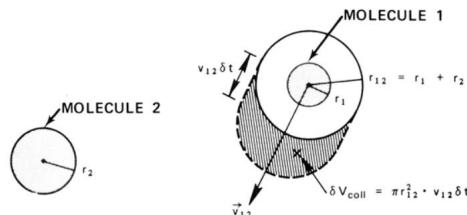
Definizione 34. *Data una reazione R_μ , con la sua costante cinetica stocastica c_μ , la cui unità di misura è sempre $\left[\frac{1}{\text{unità del tempo } dt}\right]$ (diversificandosi quindi dall'unità di misura variabile che si aveva per le costanti di reazioni nel caso deterministico), si definisce l'**ipotesi fondamentale della formulazione stocastica della cinetica chimica** che, formalmente, dice che:*

$$c_\mu dt = \bar{\mathcal{P}}$$

Dove $\bar{\mathcal{P}}$ è la probabilità media (che quindi non basta per poter descrivere la probabilità esatta nell'istante di tempo attuale, non prendendo in considerazione tutte le altre molecole in quel preciso istante temporale) che qualsiasi combinazione particolare di molecole che appaiono come reagenti nella reazione R_μ reagisca di conseguenza, nel prossimo intervallo di tempo infinitesimale dt .

Il secondo aspetto fondamentale per poter capire l'*algoritmo di Gillespie* è capire le motivazioni fisiche per le quali l'approccio stocastico è effettivamente validabile.

Si assume di lavorare nel contesto della *well-stirred hypothesis*, in un volume costante \mathcal{V} , e si considerino due molecole, una della specie S_1 e una della specie S_2 che reagiscono tra loro. In pratica si vuole studiare cosa succede quando queste due molecole “collidono”. Sfruttiamo quindi direttamente il semplice modello proposto da Gillespie nel 1977²²:



²²Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. The journal of physical chemistry, 81(25), 2340-2361.

Si hanno quindi (tenendo la stessa notazione dell'immagine):

- S_1 e S_2 approssimate come due *sfere dure*, rispettivamente di raggio r_1 e r_2 . Si noti che sarebbe comunque troppo complesso simulare direttamente la posizione delle due molecole nello spazio
- si denota $r_{12} = r_1 + r_2$ come la distanza da centro a centro che deve essere raggiunta affinché le due molecole entrino in collisione
- si denota v_{12} come la velocità relativa di S_1 rispetto a S_2
- si denota $\delta\mathcal{V}_{coll}$ come il *volume di collisione* che ci dice che se il centro della molecola S_2 si trova in un istante temporale t all'interno di questo spazio (che in pratica è lo spazio a forma di cilindro in cui si muove la molecola), allora le due molecole si scontreranno nel prossimo intervallo di tempo infinitesimale δt , quindi nell'intervallo $[t, t + \delta t]$. Come detto tale volume altro non è che un cilindro il cui volume è:

$$\delta\mathcal{V}_{coll} = \pi r_{12}^2 v_{12} \delta t$$

Si noti inoltre che se $\delta t \rightarrow 0$ allora c'è solo una probabilità trascurabile che una collisione tra le due molecole nel tempo δt sia impedita da una collisione precedente di una delle due molecole con un'altra molecola

- nell'ipotesi di *omogeneità spaziale*, il valore $\frac{\delta\mathcal{V}_{coll}}{\mathcal{V}}$ rappresenta la probabilità che il centro di una molecola di S_2 si trovi all'interno di $\delta\mathcal{V}_{coll}$

Fatte queste premesse si ha che la probabilità media che il centro di una molecola di S_2 si trovi all'interno di $\delta\mathcal{V}_{coll}$ è:

$$\overline{\left(\frac{\delta\mathcal{V}_{coll}}{\mathcal{V}}\right)} = \mathcal{V}^{-1} \pi r_{12}^2 \overline{v_{12}} \delta t$$

Che è quindi la probabilità media che avvenga una collisione tra S_1 e S_2 . Inoltre, se, al tempo t , si hanno X_1 molecole S_1 e X_2 molecole S_2 , si ha che esistono $X_1 X_2$ possibili combinazioni distinte delle stesse, avendo quindi che la probabilità esatta, denotata con \mathcal{P}_{coll} di avere una collisione tra una molecola S_1 e una molecola S_2 nel prossimo intervallo di tempo infinitesimale $(t, t + dt)$ altro non è che:

$$\mathcal{P}_{coll} = X_1 X_2 \mathcal{V}^{-1} \pi r_{12}^2 \overline{v_{12}} dt$$

Si possono dire varie cose su questa equazione:

- per calcolare $\overline{v_{12}}$ ci si affida alla cinetica di Maxwell e alla sua (e di Boltzmann) *distribuzione di velocità*, avendo che:

$$\overline{v_{12}} = \left(\frac{8kT}{\pi m_{12}} \right)^{\frac{1}{2}}$$

dove:

- k è la *costante di Boltzmann*
- T è la *temperatura assoluta*
- $m_{12} = \frac{m_1 m_2}{m_1 + m_2}$ è la *massa ridotta*

- $\mathcal{V}^{-1} \pi r_{12}^2 \overline{v_{12}} dt$ altro non è che l'**ipotesi fondamentale della formulazione stocastica della cinetica chimica**
- $\mathcal{V}^{-1} \pi r_{12}^2 \overline{v_{12}}$ nel dettaglio è la *costante stocastica cinetica* c_μ , avendo quindi che la costante stocastica tiene conto di tutte le proprietà chimiche e fisiche della reazione

Ci si avvicina quindi alla spiegazione dell'algoritmo. Data una serie di reazioni e lo stato attuale del sistema, vogliamo rispondere a due domande:

1. *quando avverrà la prossima reazione?* Chiedendosi quindi quanto tempo dobbiamo aspettare prima che la prossima reazione si verifichi da qualche parte all'interno del volume, avendo quindi che si passa ad un Δt non fisso come nel caso deterministico, avendo quindi che può passare ogni volta un tempo diverso tra una reazione e un'altra. Si noti però che, passato quel tempo, si assume che la reazione in se avvenga in tempo istantaneo
2. *quale sarà la prossima reazione?* Avendo che non per forza la reazione con più probabilità di essere la prossima sarà necessariamente quella che avverrà

Si prenda quindi un *sistema reaction-based*, con M reazioni e N specie distinte. Si ha che ogni reazione R_μ , con $1 \leq \mu \leq M$ è caratterizzata da:

- lo **state-change vector** (*vettore di cambiamento di stato*), che è un vettore di interi che denota la variazione del sistema dopo che è avvenuta una e una sola precisa reazione
- la **propensity function** (*funzione di propensione*), che è necessaria per calcolare la probabilità che occorra una certa reazione, dato lo stato attuale del sistema

State-Change Vector

Si comincia con il descrivere lo **state-change vector**. Data la reazione R_μ , in un sistema con M reazioni e N specie distinte, si ha che lo *state-change vector*, v_μ , è definito come:

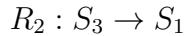
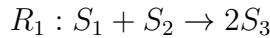
$$v_\mu = (v_{1\mu}, v_{2\mu}, \dots, v_{N\mu})$$

dove $v_{i\mu}$ denota la variazione della quantità di specie S_i , con $1 \leq i \leq N$, dovuta al verificarsi della reazione R_μ . Viene quindi specificato un vettore lungo a priori N ma se non si ha una certa specie per una certa reazione si indica la cosa tramite lo zero, avendo che di fatto tale specie non viene condizionata dalla reazione stessa.

Esempio 14. Sia dato:

$$\mathcal{S} = \{S_1, S_2, S_3\}$$

e le seguenti reazioni:



Si ottengono quindi due state-change vectors:

$$v_1 = (-1, -1, +2)$$

$$v_2 = (+1, 0, -1)$$

Si assume il seguente stato al tempo t :

$$X(t) = (X_1(t), X_2(t), X_3(t)) = (15, 7, 23)$$

Se si ha la reazione R_1 al tempo $t+1$ si avrà:

$$X(t+1) = X(t) + v_1 = (15 - 1, 7 - 1, 23 + 2) = (14, 6, 25)$$

Se si ha la reazione R_2 al tempo $t+2$ si avrà:

$$X(t+2) = X(t+1) + v_2 = (14 + 1, 6 + 0, 25 - 1) = (15, 6, 24)$$

Propensity Function

La definizione della **propensity function** è più complessa. Data la reazione R_μ in un sistema con M reazioni e N specie distinte, si ha che la *propensity function*, denotata a_μ , della reazione è definita come:

$$a_\mu(X(t)) = c_\mu h_\mu(t)$$

dove:

- c_μ è sempre la *costante stocastica cinetica* della reazione R_μ
- $h_\mu(t)$ è una funzione combinatoria che caratterizza la reazione, sia in base al suo ordine (che si ricorda non dovrebbe superare il secondo) che al tipo di reagenti. In pratica rappresenta tutte le possibili combinazioni di reazioni che si hanno al momento t al variare della tipologia della reazione. Possiamo quindi vedere alcuni esempi per il calcolo di h_μ , che è basato solo sui reagenti della reazione e non sui prodotti²³:

Reagenti	h_μ
.	$h_\mu = 1$
S_j	$h_\mu = X_j$
$S_j + S_k, j \neq k$	$h_\mu = X_j X_k$
$2S_j$	$h_\mu = \frac{X_j(X_j-1)}{2}$
$S_i + S_j + S_k, i \neq j \neq k \neq i$	$h_\mu = X_i X_j X_k$
$S_j + 2S_k, j \neq k$	$h_\mu = \frac{X_j X_k (X_k-1)}{2}$
$3S_j$	$h_\mu = \frac{X_j(X_j-1)(X_j-2)}{6}$

Probabilmente esiste una formulazione generale ma non l'ho trovata nel paper. Avendo quindi che i vari X_l cambiano nel tempo si ha che ad ogni step bisogna ricalcolare tali valori.

Possiamo inoltre dire che, avendo:

$$a_\mu(X(t)) dt = c_\mu dt h_\mu(t)$$

si ha che $a_\mu(X(t)) dt$ rappresenta il numero totale delle possibili combinazioni distinte dei reagenti di R_μ presenti nel volume \mathcal{V} al tempo t , cioè è uguale alla probabilità che, dato lo stato del sistema $X(t)$ (avendo quindi che h_μ dipende direttamente dallo stato corrente), avvenga una reazione di tipo R_μ da qualche parte in \mathcal{V} nell'intervallo di tempo $[t, t + dt]$.

Si ricorda inoltre che si ha una generalizzazione per passare da c , la *costante stocastica cinetica*, a k , la *costante deterministica cinetica*, per passare da simulazioni stocastiche a deterministiche e viceversa, che dipende dal numero di reagenti e dal loro ordine, ma per ora basta la seguente tabella riassuntiva:

²³Gillespie, D. T. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. Journal of computational physics, 22(4), 403-434.

caso	formula
1 reagente	$c = k$
2 reagenti di specie diverse	$c = \frac{k}{N_A \cdot \mathcal{V}}$
2 reagenti di specie uguali	$c = \frac{2k}{N_A \cdot \mathcal{V}}$

si ha che le costanti stocastiche c delle reazioni unimolecolari e bimolecolari sono “diverse” tra loro, avendo che:

- per le reazioni unimolecolari, c è indipendente dal volume del sistema
- per le reazioni bimolecolari c è inversamente proporzionale a \mathcal{V} , riflettendo il fatto che due molecole reagenti avranno più difficoltà a trovarsi all’interno di un volume maggiore

5.3.5 Chemical Master Equation

Il passaggio successivo per arrivare all’*algoritmo di Gillespie* è parlare della **Chemical Master Equation (CME)**.

Poiché stiamo descrivendo il sistema in termini probabilistici, ci interessa determinare $P(X, t | X_0, t_0)$, cioè la probabilità di essere nello stato X al tempo t , a partire dallo stato X_0 al tempo t_0 . In questo contesto la *CME* è l’equazione che determina la probabilità che ogni specie abbia una quantità molecolare specificata in un dato momento futuro ed è formalizzabile come, in un sistema con M reazioni e N specie distinte:

$$\frac{\partial P(X, t | X_0, t_0)}{\partial t} = \sum_{\mu=1}^M [a_\mu(X - v_\mu)P(X - v_\mu, t | X_0, t_0) - a_\mu(X)P(X, t | X_0, t_0)]$$

avendo che:

- avendo $\sum_{\mu=1}^M$ si stanno considerano tutte le reazioni del sistema
- a_μ è la *propensity function* della reazione R_μ
- v_μ è lo *state-change vector* della reazione R_μ

- $a_\mu(X - v_\mu)P(X - v_\mu, t|X_0, t_0)$ indica la probabilità di arrivare allo stato X tramite la reazione μ , avendo che lo stato $X - v_\mu$ è quello esattamente prima all'applicazione delle reazione R_μ
- $a_\mu(X)P(X, t|X_0, t_0)$ indica la probabilità di arrivare allo stato X senza che avvenga alcuna reazione
- con la *CME* si ha il calcolo della probabilità esatta che il sistema sia in un certo stato X al tempo t

Dal punto di vista più matematico però la *CME* è in realtà un sistema di *EDO accoppiate*, con un'equazione per ogni possibile combinazione di molecole reagenti. Il problema diventa quindi il numero di possibili stati che un sistema può avere. Si prenda ad esempio un catena di n semplici reazioni di isomerizzazione:

$$S_1 \rightleftharpoons \cdots \rightleftharpoons S_n$$

e uno stato iniziale:

$$X = (Z, 0, 0, \dots, 0)$$

Si ha quindi che il numero di possibili stati è:

$$\frac{(Z + n - 1)!}{Z!(n - 1)!} \approx \left(\frac{eZ}{n - 1} \right)^{n-1}$$

Avendo quindi che il numero di stati possibili è esponenziale, essendo un $O(Z^{n-1})$, avendo quindi anche un numero esponenziale di *EDO*. Si è quindi nella situazione in cui non solo risulta impossibile, a meno di particolarissimi casi ad hoc che vengono regolarmente studiati in un ramo di ricerca praticamente solo dedicato ad essi, trovare una soluzione analitica ma anche una approssimata tramite algoritmi di integrazione numerica. Possiamo quindi dire che la *CME* è **unfeasible**, praticamente “irrisolvibile”.

Si è quindi cercata una via alternativa alla risoluzione “diretta” della *CME*, cercando un modo alternativo per costruire realizzazioni numeriche di $X(t)$, cioè le *traiettorie simulate* (quindi le variazioni) del stato del sistema $X(t)$ nel tempo t che corrispondono alla *CME*. La chiave per generare traiettorie simulate di $X(t)$ non è la funzione $P(X, t|X_0, t_0)$, ma piuttosto una nuova funzione di probabilità congiunta $P(\tau, \mu|X, t)$, che è chiamata **funzione densità di probabilità della reazione**. Tale funzione di probabilità congiunta è la *funzione densità di probabilità* di due variabili casuali:

- τ , che è il tempo dopo il quale avverrà reazione successiva
- μ , che è l'indice della prossima reazione

Tramite vari massaggi matematici, volendo disponibili nel paper di Gillespie del 1976²⁴, è possibile ricavare una formula esatta per $P(\tau, \mu|X, t)$ applicando le leggi della probabilità, ottenendo:

$$P(\tau, \mu|X, t) = a_\mu(X) e^{-a_0(X)\tau}$$

Avendo che:

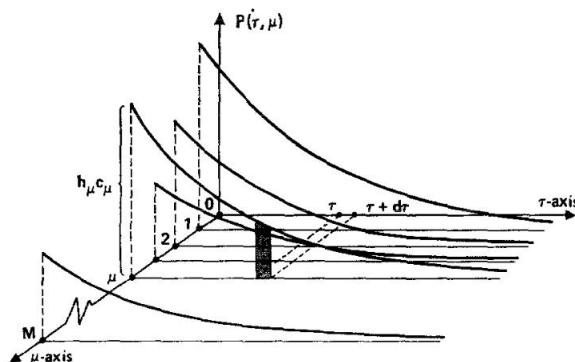
$$a_0(X) = \sum_{\mu=1}^M a_\mu(X)$$

Tale valore non cambia fino a che non cambia lo stato del sistema X , ovvero fino a che non si fa un altro step di simulazione.

Scomponendo l'equazione si nota che:

- $a_\mu(X)$ è praticamente la probabilità che una reazione R_μ avverrà dopo un intervallo di tempo lungo τ
- $e^{-a_0(X)\tau}$ è praticamente la probabilità che non avverrà nessuna reazione dopo un intervallo di tempo lungo τ , avendo che tale probabilità decresce esponenzialmente nel tempo

Si ha quindi che si ottiene praticamente una possibile soluzione della *CME*. Dal punto di vista grafico potremmo ottenere, sempre dal paper di Gillespie del 1976, un grafico tipo:



L'area sotto ogni curva rappresenta la probabilità che la reazione si verifichi nell'istante temporale successivo. L'area ombreggiata in centro, uguale a $P(\tau, \mu|X, t)$, è la probabilità che, all'istante t , la reazione successiva in \mathcal{V}

²⁴Gillespie, D. T. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. Journal of computational physics, 22(4), 403-434.

avvenga nel tempo infinitesimo $d\tau$ e sarà una reazione di tipo R_μ . Si noti quindi che tale probabilità dipende da tutte le altre probabilità relativamente rispettive tutte le altre reazioni del sistema. Si noti inoltre che l'area sottesa alle curve del grafico, intesa come somma delle stesse in un certo intervallo di tempo, ovvero il rispettivo integrale da 0 a $+\infty$, è pari a 1, avendo quindi la non indipendenza delle varie funzioni.

Si è quindi arrivati alla formula che è alla base dell'*approccio di simulazione stocastico*, avendo che:

- τ è una variabile casuale esponenziale con media (ma anche deviazione standard) pari a $\frac{1}{a_0(X)}$
- μ è una variabile intera causale statisticamente indipendente con *point probability* pari a $\frac{a_\mu(X)}{a_0(X)}$

Esistono quindi vari modi per generare questi due valori rispetto alle loro due distribuzioni di probabilità. Fatta questa premessa possiamo riscrivere l'equazione come:

$$P(\tau, \mu | X, t) = P_1(\tau)P_2(\mu)$$

Avendo:

- $P_1(\tau) = a_0(X)e^{-a_0(X)\tau}$
- $P_2(\mu) = \frac{a_\mu(X)}{a_0(X)}$

Resta solo quindi capire come determinare τ e μ .

Si inizia con il determinare τ , generando in primis un numero pseudocasuale r_1 a partire da una distribuzione uniforme sull'intervallo $[0, 1]$. A questo punto si ha che (capendo anche perché r_1 non può essere pari a 1):

$$\tau = \frac{1}{a_0(X)} \ln \frac{1}{r_1}$$

Avendo quindi che τ è inversamente proporzionale ad $a_0(X)$, avendo quindi che all'aumentare di $a_0(X)$ si avranno step temporali τ più piccoli e quindi un maggior tempo macchina per l'intera simulazione. Per questo motivo, per evitare questo costo macchina, solitamente si fanno simulazioni stocastiche in presenza di sistemi con poche quantità di tutte le specie del sistema, che è anche la situazione in cui si ha comunque maggior *rumore biologico*.

Si determina infine μ , anche in questo caso generando in primis un numero pseudocasuale r_2 a partire da una distribuzione uniforme sull'intervallo $[0, 1]$. Si ha quindi che μ è quell'intero tale per cui:

$$\sum_{j=1, \dots, \mu-1} a_j(X) \leq r_2 a_0(X) \leq \sum_{j=1, \dots, \mu} a_j(X)$$

In altri termini si prende ogni valore di $a_\mu(X)$ e lo si divide per $a_0(X)$ e, mantenendo per praticità l'ordine delle reazioni (ma non è l'unica soluzione), si determina il sotto-intervallo tra 0 a 1 che rappresenta la probabilità che ciascuna reazione ha di accadere. Praticamente si fa una *normalizzazione*. Ad esempio, avendo tre reazioni, se $\frac{a_1X}{a_0(X)} = 0.7$, $\frac{a_2X}{a_0(X)} = 0.2$ e $\frac{a_3X}{a_0(X)} = 0.1$ si avrebbe che se r_1 è un valore in $[0, 0.7)$ allora si sceglierà la reazione 1, in $[0.7, 0.9)$ la 2 o altrimenti la 3. Si noti come non per forza sarà la più probabile la reazione che si andrà a scegliere. Infatti si ha che la probabilità che venga scelta R_μ è proporzionale ad $a_\mu(X) = c_\mu h_\mu(t)$, avendo che questa probabilità è proporzionale alla *costante stocastica cinetica* moltiplicata per il numero di molecole reagenti attualmente presenti nel sistema, comportando che il tempo di esecuzione della simulazione sarà maggiore se il sistema contiene un numero elevato di molecole.

Concludendo possiamo quindi dire che τ serve ad aggiornare il valore del tempo t mentre μ a scegliere con quale reazione, e nel dettaglio quindi con quale *state-change vector*, aggiornare lo stato del sistema X .

Algoritmo di Gillespie

Si è quindi arrivati a descrivere il vero e proprio **algoritmo di Gillespie**, o meglio il **Stochastic Simulation Algorithm (SSA)**, proposto appunto da Gillespie nel 1976.

SSA è un *algoritmo iterativo* che consente di simulare l'evoluzione temporale di un dato modello di reazione, generando realizzazioni della corrispondente *CME*, a partire da una condizione iniziale. Tale algoritmo si fonda su un *processo Markoviano*, avendo che il sistema procede di fatto applicano una reazione per volta e come output produce l'insieme dei vari stati del sistema, quindi, isolando una determinata specie (quindi studiando la quantità della stessa in ogni stato prodotto in output), è possibile studiarne l'evoluzione durante la simulazione, ovvero la sua traiettoria. Il flowchart dell'algoritmo è visualizzabile in figura 5.12. L'*algoritmo SSA* ha comunque dei limiti, così come ovviamente risolvere sistemi di *EDO*. Nel dettaglio *SSA* deve ricalcolare ad ogni step le *propensity function* e inoltre si ha l'assunzione di lavorare in un ambiente *well-stirred*. Parlando di *propensity functions* si ha che sono utilizzate per valutare la reazione successiva R_μ e il tempo τ prima che la reazione si verifichi quindi:

- minore è τ e maggiore è il tempo di simulazione
- le *propensity functions* sono proporzionali al prodotto del numero di molecole reagenti e quindi *SSA* è rallentato in sistemi caratterizzati da un elevato numero di molecole

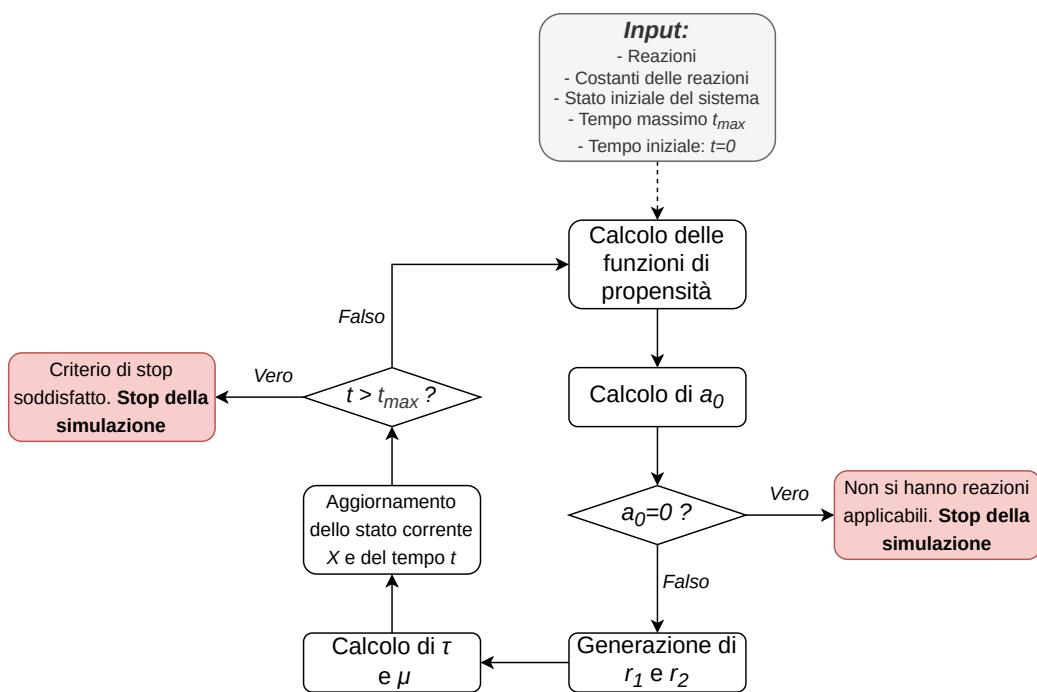


Figura 5.12: Flowchart di *SSA*, con specificati gli input e i possibili casi di uscita dall'iterazione.

Una prima soluzione è data dal fatto che si potrebbe calcolare le *propensity functions* solo in presenza di reagenti la cui quantità è mutata rispetto allo step precedente, infatti nel caso contrario la *propensity function* non cambia quindi è inutile sprecare tempo per computarla nuovamente.

5.4 Deterministico VS Stocastico

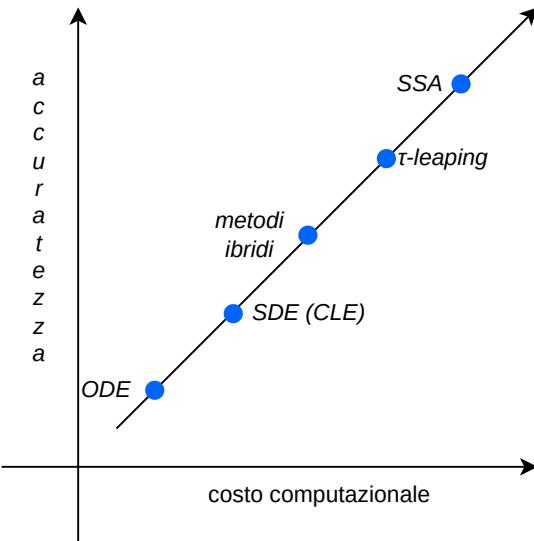
Risulta comodo questo punto un confronto diretto tra le *simulazioni deterministiche* e le *simulazioni stocastiche*.

Categoria	S. stocastiche	S. deterministiche
Tempo di simulazione	lento, in quanto le reazioni sono scelte casualmente e applicate una dopo l'altra	veloce, in quanto vengono integrate le equazioni differenziali simultaneamente (ottenendo il comportamento medio)
Intervalli temporale	Il tempo, τ tra due reazioni è scelto in modo casuale	Lo step di integrazione, Δt , è fisso per tutta l'esecuzione ed è un parametro dell'algoritmo. Con soluzioni adattive potrebbe cambiare durante l'esecuzione
“Unità di misura” delle molecole	lo stato del sistema è caratterizzato da quantità molecolari discrete, quindi tramite numeri in \mathbb{N}	lo stato del sistema rappresenta la media di quantità molecolari, quindi tramite le concentrazioni che sono numeri in \mathbb{R}
Rumore biologico	Si considerano gli effetti del rumore biologico	Non si considerano gli effetti del rumore biologico
Output	Data una certa condizione iniziale si generano output qualitativamente diversi, essendo stocastici, per ogni simulazione	Data una certa condizione iniziale si genera sempre lo stesso output, essendo deterministico, per ogni simulazione
Algoritmi	SSA, τ -leaping, ...	Eulero, Runge-Lutta, LSODA, ...

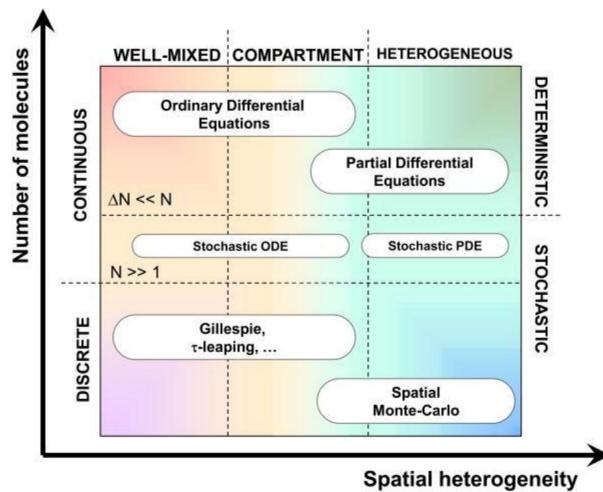
5.5 Altri Algoritmi

In questo capitolo si sono approfonditi solo alcuni dei principali algoritmi deterministici e stocastici per effettuare simulazioni su modelli biologici. Nel dettaglio, con SSA e con gli algoritmi di integrazione numerica di *sistemi di EDO*, abbiamo approfondito i “casi limite”, dal punto di vista dell'**accuratezza della simulazione** e del **costo computazionale**. Infatti

possiamo cedere come in questo grafico ci siano molti altri algoritmi, che si approfondiranno brevemente (anche spiegando le sigle), che si piazzano nel mezzo:

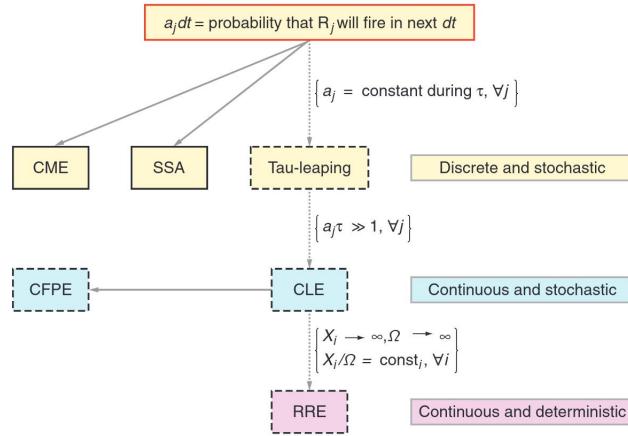


Lo stesso discorso si può verificare in quest’altro grafico²⁵ dove ci si basa sull’eterogeneità spaziale (avendo che *well-mixed* corrisponde all’assunzione di *well-stirred*):



²⁵Radhakrishnan, Krishnan et al. “Quantitative understanding of cell signaling: the importance of membrane organization.” Current opinion in biotechnology vol. 21,5 (2010): 677-82. doi:10.1016/j.copbio.2010.08.006

Mentre un criterio per scegliere cosa usare viene fornito da Gillespie stesso²⁶ (con *RRE* si intende *reaction-rate equation*, che sono le simulazioni deterministiche su *sistemi di EDO*):



5.5.1 τ -leaping

Il primo algoritmo che si introduce brevemente è il **τ -leaping**, che è una sorta di ottimizzazione di *SSA* in quanto, nel tempo τ si simulano contemporaneamente più reazioni. Nel dettaglio, rispetto a *SSA*, si procede con intervalli di tempo τ più lunghi, e si approssima il numero di occorrenze di ogni reazione R_μ in $(t, t + \tau)$ con un **processo Poissoniano** $\mathcal{P}(a_\mu(X) \cdot \tau)$, avendo media e varianza uguali a $a_\mu(X) \cdot \tau$. Per far ciò si deve garantire la cosiddetta **condizione di leap**, ovvero: *si richiede che τ sia piccolo abbastanza in modo che il cambiamento di stato in $(t, t + \tau)$ sia così poco significante che nessuna propensity function subisca un forte cambiamento (in termini di valori macroscopici non infinitesimali) nel suo valore.*

Si ottiene quindi un algoritmo più performante di *SSA* ma meno accurato, in quanto si simulano contemporaneamente reazioni che soddisfano la *condizione di leap*. Si hanno anche altri svantaggi:

- le reazioni “veloci” invalidano facilmente la *condizione di leap*
- la generazione di *numeri di Poisson* è computazionalmente costosa
- si possono raggiungere stati non validi e non sensati, addirittura stati con concentrazioni per le molecole negative. Si hanno comunque soluzioni a questo problema

²⁶Gillespie, D. T. (2007). Stochastic simulation of chemical kinetics. Annu. Rev. Phys. Chem., 58, 35-55.

5.5.2 Equazioni Differenziali Stocastiche

Un'altra soluzione è quella di risolvere le **equazioni differenziali stocastiche** (*stochastic differential equation, SDEs*), che sono delle *EDO* estese con termini stocastici legati al *rumore biologico*. Sono ad un livello intermedio tra gli algoritmi di simulazione deterministici e algoritmi di simulazione stocastica, essendo più veloci ma meno accurati di questi ultimi. Adottano comunque step di integrazione fissi, al più di algoritmi adattivi. Le equazioni più comuni sono le **Chemical Langevin Equations (CLEs)**, in cui il numero di esecuzioni di una reazione è approssimato da una distribuzione normale:

$$X(t + \tau) = X(t) + \sum_{\mu=1}^M v_{\mu u} \mathcal{N}(a_\mu(X) \cdot \tau, a_\mu(X) \cdot \tau)$$

Si noti che le *CLE* possono dare risultati sbagliati quando una specie chimica è caratterizzata da poche molecole.

5.5.3 Metodi Ibridi

La soluzione perfettamente intermedia tra il caso deterministico e quello stocastico si ha con i **metodi ibridi**, che sono un modo per affrontare i sistemi biochimici caratterizzati da scale multiple.

L'insieme delle reazioni è suddiviso, *partizionato*, in più *regimi*, in base alle caratteristiche delle reazioni (alte concentrazioni di reagenti, alta velocità cinetica, etc ...) e differenti partizioni sono assegnate a diversi algoritmi di simulazione (ad esempio le reazioni veloci si possono risolvere in modo deterministico mentre quelle lente in modo stocastico).

Anche qui si hanno però dei problemi:

- regimi diversi potrebbero utilizzare diverse unità (come usare le concentrazioni nelle simulazioni deterministiche e il numero di molecole in quelle stocastiche, avere parametri stocastici e parametri cinetici etc...). È necessaria quindi la conversione automatica delle unità
- i modelli deterministicci solitamente usano step fissi, Δt , mentre quelli stocastici step temporali randomici, τ . I due regimi devono quindi essere armonizzati in qualche modo avendo per di più che però Δt , se fissato all'inizio, è conosciuto mentre τ no

La questione del *partizionamento* può essere approfondito, avendo che esso può essere **statico**, quindi deciso a priori, o **dinamico**, quindi deciso ogni

volta durante la simulazione. Si noti che il partizionamento dinamico è computazionalmente costoso ma può gestire sistemi le cui specie chimiche sono caratterizzate da ampi cambiamenti di quantità molecolare, causando reazioni per cambiare regime.

Si segnala il **Partitioned Leaping Algorithm (PLA)**, di Harris e Clancy²⁷ che usa quattro algoritmi per quattro scale diverse, sfruttando un partizionamento dinamico basato sulle *propensity functions*:

$$\begin{cases} a\mu \cdot \tau \lesssim 1 & \text{SSA} \\ a\mu \cdot \tau > 1 \wedge a\mu \cdot \tau \gg 1 & \tau\text{-leaping} \\ a\mu \cdot \tau \gg 1 \wedge \sqrt{a\mu \cdot \tau} \gg 1 & \text{SDE} \\ \sqrt{a\mu \cdot \tau} \gg 1 & \text{EDO} \end{cases}$$

5.5.4 Modelli spaziali

Finora, abbiamo assunto che il sistema fosse *well-stirred*, avendo che le specie chimiche sono uniformemente distribuiti all'interno del volume di reazione. Questo non è sempre vero, infatti, oltre agli compartimenti cellulari, alcune molecole possono essere localizzate nella cellula (recettori di membrana, buffering locale, etc...). Si vuole quindi estendere il modello con le *reazioni di diffusione*. Si ottengono così i **modelli spaziali**, anche 3D, che però sono estremamente costosi dal punto di vista computazionale e quindi vanno usati solo in casi estremamente specifici.

Si hanno tre approcci principali:

1. quelli basati sulle **equazioni differenziali parziali** (*non introdotte in questo corso*)
2. quelli **particles-based**. Tali approcci trattano ogni particella in modo esplicito:
 - ogni particella è considerata come un punto nello spazio euclideo
 - si tiene traccia delle posizioni delle particelle e del *movimento stocastico browniano*, che è il moto disordinato di particelle sufficientemente piccole (aventi diametro dell'ordine del micrometro) da essere sottoposte a una forza di gravità trascurabile, presenti in fluidi o

²⁷Harris, Leonard A., and Paulette Clancy. "A "partitioned leaping" approach for multiscale modeling of chemical reaction dynamics." *The Journal of chemical physics* 125.14 (2006): 144107.

sospensioni fluide o gassose (ad esempio il fumo), e osservabile al microscopio (tale fenomeno fu scoperto agli inizi dell'Ottocento dal botanico scozzese Robert Brown, e modellizzato nel 1905 dal fisico teorico tedesco Albert Einstein). A tal proposito si ricordi la *chemiotassi batterica*

- si ha una reazione quando due particelle collidono
- si ha che il movimento può essere ostruito da corpi solidi per simulare scomparti affollati

Un software per tali simulazioni è **Smoldyn**^{28²⁹}

Come introdotto tali modelli sono onerosi dal punto di vista computazionale in quanto ogni particella è modellata esplicitamente e le collisioni particella-particella devono essere calcolate

3. quelli **subvolumes-based**. Tali approcci dividono il volume di simulazione in sottovolumi minori. Infatti:

- viene tracciato il numero di molecole in ogni sottovolume invece della posizione specifica di ogni molecola
- si assume che i sottovolumi siano abbastanza piccoli da essere *well-stirred*, in modo che ogni sottovolume può essere simulato utilizzando metodi classici (deterministici, stocastici, ibridi etc...)
- si aggiungono *reazioni di diffusione* per spostare le particelle da un sottovolume ai sottovolumi vicini

Un software per tali simulazioni è **Next Subvolume Method (NSM)**³⁰

Si hanno vari problemi però, tra cui il fatto che presunti eventi di reazione/diffusione nei sottovolumi sono ordinati secondo i loro valori τ e quindi evento con il più piccolo τ è selezionato e al massimo due sottovolumi vengono aggiornati durante ogni step, per evitare stati non validi. Inoltre anche questi modelli sono molto onerosi dal punto di vista spaziale.

²⁸Andrews, Steven S., and Dennis Bray. "Stochastic simulation of chemical reactions with spatial resolution and single molecule detail." *Physical biology* 1.3 (2004): 137.

²⁹Steven S Andrews, Smoldyn: particle-based simulation with rule-based modeling, improved molecular interaction and a library interface, *Bioinformatics*, Volume 33, Issue 5, 1 March 2017, Pages 710–717, <https://doi.org/10.1093/bioinformatics/btw700>

³⁰Elf, Johan, and Måns Ehrenberg. "Spontaneous separation of bi-stable biochemical systems into spatial domains of opposite phases." *Systems biology* 1.2 (2004): 230-236.

5.6 Parameter Estimation

L'intera sezione riguarda la lezione tenuta dal professor Nobile.

Come si è visto, parlando soprattutto di modelli meccanicistici, il ruolo dei parametri è fondamentale. Alcuni esempi di parametri dei modelli, come visto, si ritrovano nelle *concentrazioni/numero di molecole* nello stato iniziale o nelle costanti per i *kinetic rate*. In generale il nostro modello è quindi formato, in modo astratto:

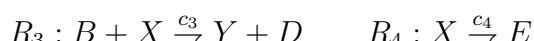
- da un **input**, ovvero, ad esempio, lo *stato iniziale* e un vettore di parametri per il modello, denotato con ϑ
- dal **simulatore del modello**, che possiamo pensare come una *black box*
- dall'**output**, ad esempio il *comportamento simulato del sistema*

Per capire al meglio l'importanza dei parametri si vede un esempio.

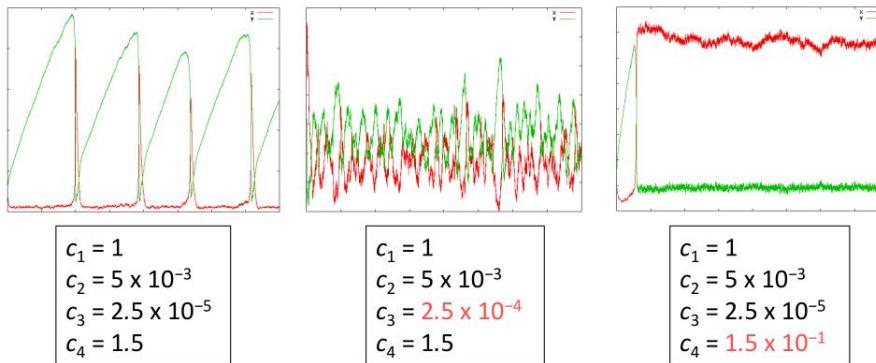
Esempio 15. Si considera il modello del **Brusselator**, che consta di sei specie (più tecnicamente quelle che si ottengono al variare del coefficiente stochiometrico):

1. *A*, con una quantità iniziale di 200 molecole. Tale quantità è fissata
2. *X*, con una quantità iniziale di 200 molecole
3. *B*, con una quantità iniziale di 600 molecole. Tale quantità è fissata
4. *Y*, con una quantità iniziale di 300 molecole
5. *D*, con quantità iniziale nulla
6. *E*, con quantità iniziale nulla

e di 4 reazioni:



Si ottengono quindi vari risultati al variare dei valori delle costanti cinetiche, risultati anche molto diversi tra loro:



Avendo nel primo caso delle ampie oscillazioni, nel secondo oscillazioni con frequenza maggiore e nel terzo addirittura bi-stabilità.

Si nota quindi come scegliere tali parametri sia essenziale ma essi sono parametri che assumono valori in \mathbb{R} quindi, qualsiasi sia il parametro e qualsiasi sia il range di valori assumibili, si hanno infiniti valori possibili (essendo \mathbb{R} **denso**). Risulta quindi chiaro come esplorare tutte le possibili combinazioni di parametri non sia possibile ma servano delle strategie per esplorare lo *spazio delle possibili parametrizzazioni*.

In primis bisogna fare alcune considerazioni:

- normalmente i parametri, come quelli cinetici, non possono essere misurati direttamente in *wet-lab*
- alcuni parametri possono essere calcolati in modo indiretto dai dati sperimentali, ad esempio studiando le *time-series* di quantità molecolari

In questo contesto quindi si definisce il **problema del Parameter Estimation (PE)**, che è appunto la ricerca del *vettore dei parametri ϑ in silico*, di cui si segnala un interessante paper di Moles et al.³¹. Si noti che il *problema PE* è un **problema NP-hard**. Con la *PE* si hanno comunque alcune assunzioni per quanto si andrà a trattare in questa sezione:

- la struttura del modello è conosciuta mentre il problema di stimare sia le reazioni che i loro parametri rientra nel campo del **reverse engineering**

³¹Moles, Carmen G., Pedro Mendes, and Julio R. Banga. "Parameter estimation in biochemical pathways: a comparison of global optimization methods." Genome research 13.11 (2003): 2467-2474.

- lo stato iniziale del sistema, quindi concentrazioni/numero di molecole, è conosciuto anche se volendo anche tali parametri, potenzialmente sconosciuti, sono stimabili ma questo campo non verrà trattato in questa sezione

Si ha quindi, con R insieme delle reazioni, un potenziale algoritmo naïve per la PE , al fine di fare un confronto coi dati ottenuti, che saranno i dati target sperimentali T , in *wet-lab*:

1. si sceglie un vettore di parametri cinetici casuale $\vartheta \in \mathbb{R}^{|R|}$
2. si usa ϑ per effettuare la simulazione, che sia deterministica o stocastica
3. si genera una dinamica S_ϑ dalla simulazione da confrontare con T
4. se S_ϑ differisce da T si migliora ϑ e si ricomincia dallo step 2)
5. si restituisce la parametrizzazione ϑ

Ci sono però vari passi da chiarire:

- come si comparano S_ϑ e T ?
- come si migliora ϑ ?
- come si genera ϑ avendo che farlo in modo completamente causale non è sicuramente ottimale?

5.6.1 Fitness Function

Al fine di poter comparare le dinamiche simulate con quella target attesa viene usata la cosiddetta *fitness function*, definita, dato un vettore di parametri ϑ , come:

$$f(\vartheta) : \mathbb{R}^{|R|} \rightarrow \mathbb{R}^+$$

Tale funzione quindi compara le dinamiche simulate S_ϑ e il target T per quantificare la qualità del fit. Nella sua forma naïve si ha che:

$$f(\vartheta) = \|T - S_\vartheta\|$$

avendo che quindi il risultato ottimale sarebbe $f(\vartheta) = 0$. D'altro canto questa è una definizione assolutamente generale, avendo che le simulazioni e il target possono essere potenzialmente qualsiasi cosa (*time-series, istogrammi per la*

fluorescenza delle cellule, etc...), le simulazioni possono essere stocastiche o deterministiche, ogni misura di distanza può essere usata come fitness etc... La definizione quindi della funzione di fitness è una “questione delicata”. Si hanno quindi varie funzioni, ad esempio si ha la variante naive per il confronto di due time-series:

$$f^{-1}(\vartheta) = \frac{1}{TN} \sum_{t=1}^T \sum_{n=1}^N |T^n(t) - S_\vartheta^n(t)|$$

Avendo però che, all'aumentare della quantità di molecole delle specie, si rischiano grossi errori, che non dovrebbero essere trascurabili.

Oppure funzioni più elaborate come la **Mean Absolute Percentage Error (MAPE)**, che in pratica sfrutta ogni volta l'errore percentuale (anche se non può essere usata se $T^n(t) \rightarrow 0$):

$$f^{-2}(\vartheta) = \frac{1}{TN} \sum_{t=1}^T \sum_{n=1}^N \left| \frac{T^n(t) - S_\vartheta^n(t)}{T^n(t)} \right|$$

Dove, in entrambe, avendo qui che *valore=numero di specie*:

- $T^n(t)$ denota il valore del target per la specie n al tempo t
- $S_\vartheta^n(t)$ denota il valore della simulazione, con parametri iniziali ϑ , per la specie n al tempo t

Si nota che *MAPE* riduce riduce l'impatto sul valore di fitness finale delle specie con un numero elevato di molecole ma, come anticipato, si rompe quando un bersaglio tende ad avere zero molecole.

Fatte queste premesse si può quindi ridurre il *problema PE* al problema di determinare la soluzione ottima, che viene denotata con o , nello spazio \mathcal{S} di tutte le possibili soluzioni (che per noi sono parametrizzazioni cinetiche). Tale soluzione ottima è quindi quella con il valore di fitness più piccolo possibile, avendo, formalmente, che:

$$f(o) \leq f(\vartheta), \quad \forall \vartheta \in \mathcal{S}$$

Ovvero una soluzione che non può essere ulteriormente migliorata rispetto alla funzione fitness. Ne segue che il *problema PE* viene riformulato come un problema di ottimizzazione, in particolare di minimizzazione. Concettualmente bisogna mettere quindi dei vincoli alle possibili soluzioni. L'obiettivo è minimizzare la distanza tra il target e la dinamica simulata, sfruttando molti dati per l'ottenimento del risultato. Infatti, avendo magari pochi dati, che

idealmente possiamo pensare nel caso più semplice come punti su un piano cartesiano, si rischia di modellare un risultato completamente irrealistico rispetto alla realtà (ad esempio con tre punti penso di avere un comportamento lineare ma già con cinque scopro che è oscillatorio). Un altro problema da considerare è quello dell'**overfitting**, in quanto l'uso di troppi dati rischia di portare anche a modellare il *rumore*, rendendo più complesso lo studio. Purtroppo comunque, parlando del target, solitamente il *wet-lab* produce pochissimi time-points, nell'ordine di cinque o sei, nella maggior parte dei casi. Inoltre spesso sono i momenti iniziali, difficili da misurare con frequenza, ad essere rilevanti in quanto dopo poco si rischia già di avere uno steady state.

5.6.2 Gradient Descent

Ipotizziamo di avere una semplice funzione $f : \mathbb{R} \rightarrow \mathbb{R}$ che plottiamo sul piano cartesiano. In ogni punto, per virtualmente scegliere in che direzione andare per cercare il minimo, possiamo usare le *derivate*, in quanto ci dicono la *pendenza* della funzione in quel punto. Il problema in questo caso diventa quindi trovare il punto in cui si ha derivata nulla, avendo quindi che ci si trova in un minimo (si noti che non si può qui distinguere se sia un *minimo locale* o un *minimo globale*).

Estendendo il discorso a più dimensioni, avendo nel nostro caso un vettore di parametri, abbiamo il concetto di **gradiente**, avendo che il *gradiente* in un punto è un vettore che fornisce le stesse informazioni delle derivata riguardo al pendenza della funzione, ma nel caso multidimensionale. Infatti è un vettore che punta nella direzione della salita più ripida e quindi basta seguire il *gradiente inverso* per cercare il minimo, avendo anche qui che non si può qui distinguere se sia un *minimo locale* o un *minimo globale*. Formalmente il gradiente definisce no *spazio vettoriale* e, ad esempio per due dimensioni, si ha che:

$$\nabla f(x, y) = \frac{\partial f}{\partial x} i + \frac{\partial f}{\partial y} j$$

Si ha quindi l'algoritmo deterministico detto **gradient descent (GD) algorithm**, che parte con la scelta casuale di una soluzione iniziale nello spazio delle soluzioni \mathcal{S} e calcolandone il gradiente. A questo punto si compie un *passo* nella direzione opposta a quella puntata dal gradiente, muovendosi quindi verso un minimo. Si itera fino a quando il gradiente è zero, cioè, non c'è direzione che migliora ulteriormente la funzione fitness. In pratica quindi si sta esplorando la superficie multidimensionale definita dalla *funzione di fitness*, superficie che viene detta **fitness landscape** e quindi *GD* altro non è che un algoritmo deterministico che si muove verso il basso in tale superficie, rischiando però di restare “intrappolato” in un minimo locale.

Una variante adattiva, che gestisce in modo adattivo la lunghezza dei *passi* è l'**algoritmo di Levenberg-Marquardt (LM)**³² ma anche in questo caso non si evita il problema dei minimi locali.

Tale problema può essere mitigato usando un *approccio multi-start*, ovvero banalmente ripetendo l'algoritmo con una scelta causale iniziale diversa, questo per molte volte. La probabilità di convergere alla stessa soluzione, che magari è un minimo locale, viene quindi ridotta ma non si sta davvero risolvendo il problema. Il punto cruciale è che *GD* (ma anche *LM*) è un algoritmo strettamente deterministico, funzionando bene solo con *fitness landscape unimodali*, quindi con un solo punto di minimo, che nel dettaglio è quello globale non avendo minimi locali.

Si vedrà a breve quindi come la soluzione sia quella di “far sbagliare” volontariamente l'algoritmo.

5.6.3 Simulated Annealing

Una soluzione è quella di usare una **meta-euristica stocastica per ottimizzazione globale**, come l'algoritmo detto **Simulated Annealing (SA)**³³, stimando quindi l'ottimo globale evitando quelli locali.

Questo metodo prende spunto dal mondo metallurgico, dove per eliminare difetti causati da ordini dei reticolati cristallini che vengono a mancare si procede con la loro parziale rifusione, l'*annealing* appunto, seguita da una procedura di raffreddamento molto lenta. Continuando l'analogia della metallurgica si ha:

- l'ottimo globale corrisponde al metallo con struttura cristallina senza difetti
- gli ottimi locali corrispondono al metallo con struttura cristallina con presenza di difetti reticolari
- la procedura algoritmica per trovare il minimo/massimo globale, sfuggendo ai minimi/massimi locali locali, corrisponde al processo di lento raffreddamento ed eventuale riscaldamento

Dal punto di vista algoritmico abbiamo una soluzione simile a *GD* ma con la possibilità di prendere direzioni errate.

Il procedimento dell'algoritmo è il seguente:

³²Marquardt, Donald W. "An algorithm for least-squares estimation of nonlinear parameters." Journal of the society for Industrial and Applied Mathematics 11.2 (1963): 431-441.

³³Kirkpatrick, Scott, C. Daniel Gelatt Jr, and Mario P. Vecchi. "Optimization by simulated annealing." science 220.4598 (1983): 671-680.

1. si genera una soluzione iniziale ϑ
2. si inizializza un parametro, che per analogia viene detto *temperatura*, $T \in \mathbb{R}^+$
3. si genera una soluzione vicina a ϑ , chiamata ϑ_{new}
4. se, con *f funzione di fitting*, $f(\vartheta_{new})$ è miglior e di $f(\vartheta)$ allora $\vartheta \leftarrow \vartheta_{new}$. In caso contrario posso comunque accettare il nuovo candidato con una certa probabilità:

$$P(\text{accept } \vartheta_{new} | f(\vartheta) \text{ is better than } f(\vartheta_{new})) = e^{-\frac{|f(\vartheta) - f(\vartheta_{new})|}{T}}$$

5. si aggiorna T e si itera dallo step 3) fino a che i criteri di stop non sono soddisfatti
6. si ritorna ϑ come soluzione ottima

5.6.4 Meta-Euristiche Population-Based

Si è visto come *GD* e *SA*, nel caso semplice, si basino sul miglioramento iterativo di una singola scelta iniziale di soluzione per ϑ .

In generale esistono due classi di algoritmi ispirati alla biologia basate su popolazioni di soluzioni candidate, largamente impiegate per la stima dei parametri:

1. **evolutionay computation**³⁴, che sono algoritmi basati su processi darwiniani (selezione, crossover, mutazioni etc...), tra cui:
 - algoritmi genetici
 - strategie d'evoluzione
 - evoluzione differenziale
2. **swarm intelligence**³⁵, con algoritmi che sfruttano i comportamenti emergenti di gruppi di agenti (come pesci, uccelli, api etc...), tra cui:
 - ant colony optimization

³⁴De Jong, K. (2016, July). Evolutionary computation: a unified approach. In Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion (pp. 185-199).

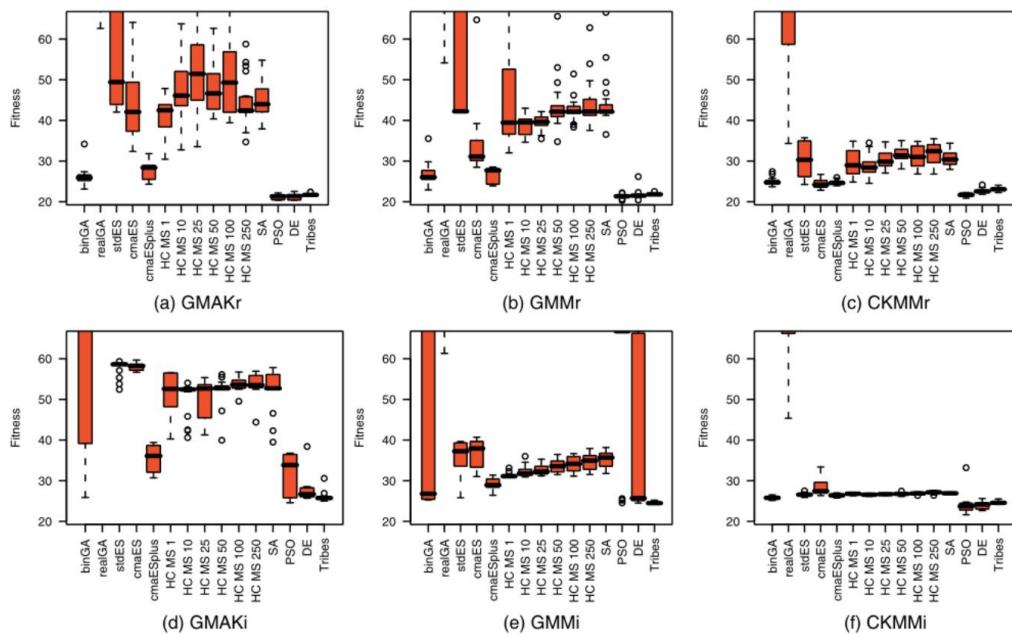
³⁵Eberhart, R. C., Shi, Y., & Kennedy, J. (2001). Swarm intelligence. Elsevier.

- bat algorithm
- particle swarm optimization (*PSO*)

Interessante è un paper di Dräger³⁶, dove si confrontano vari algoritmi:

- Gradient Descent (GD)
- Hill Climbing (HC)
- Simulated Annealing (SA)
- Genetic Algorithms (GA)
- Differential Evolution (DE)
- Evolution Strategy (ES)
- Covariance Matrix Adaptation Evolution
- Strategy (CMA-ES)
- Particle Swarm Optimization (PSO)
- Tribes

Ottenendo i seguenti plot per la *miglior fitness media*:



Notando come *PSO* sia miglior algoritmo, sul comportamento medio, con anche la minor varianza

³⁶Dräger, A., Kronfeld, M., Ziller, M. J., Supper, J., Planatscher, H., Magnus, J. B., ... & Zell, A. (2009). Modeling metabolic networks in *C. glutamicum*: a comparison of rate laws in combination with various parameter optimization strategies. BMC Systems Biology, 3(1), 1-24.

5.6.5 Particle Swarm Optimization

Particle Swarm Optimization (PSO)³⁷ è una meta-euristica stocastica bio-ispirata basata sul comportamento emergente di pesci e uccelli.

In questo metodo le *particelle* rappresentano le soluzioni candidate, quindi i *vettori coi parametri cinetici*. In pratica le *particelle* rappresentano l'intera parametrizzazione del modello. Si ha quindi che le particelle si muovono all'interno di uno spazio con D dimensioni, che rappresenta lo spazio di ricerca, e cooperano al fine di identificare e convergere alla soluzione ottima. Inoltre la particella i -esima è caratterizzata da tre vettori:

1. $\mathbf{x}_i(t) \in \mathbb{R}^D$ che rappresenta la *posizione* della particella i -esima nello spazio di ricerca all'iterazione t
2. $\mathbf{v}_i(t) \in \mathbb{R}^D$ che rappresenta la *velocità* della particella i -esima all'iterazione t
3. $\mathbf{b}_i(t) \in \mathbb{R}^D$ che è la *miglior posizione*, stimata tramite la *funzione di fitness*, visitata dalla particella i -esima fino all'iterazione T

Nel dettaglio la velocità è usata per aggiornare la posizione delle particelle:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t)$$

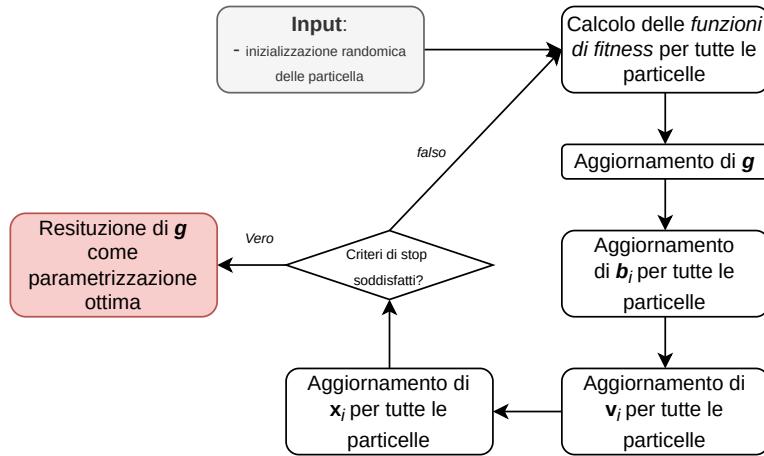
Inoltre si hanno due valori per cui possono variare le velocità:

1. la **social attraction**, denotato con \mathbf{g} , ovvero l'attrazione verso la miglior posizione trovata dallo *swarm/sciamo* di particelle. Tale valore viene modulato tramite il cosiddetto **fattore sociale**, denotato con c_{soc} . In pratica modula l'esplorazione locale (???)
2. la **cognitive attraction**, denotato con \mathbf{b}_i , ovvero l'attrazione verso la miglior posizione trovata dalla particella i -esima. Tale valore viene modulato tramite il cosiddetto **fattore cognitivo**, denotato con c_{cog} . In pratica modula l'esplorazione globale (???)

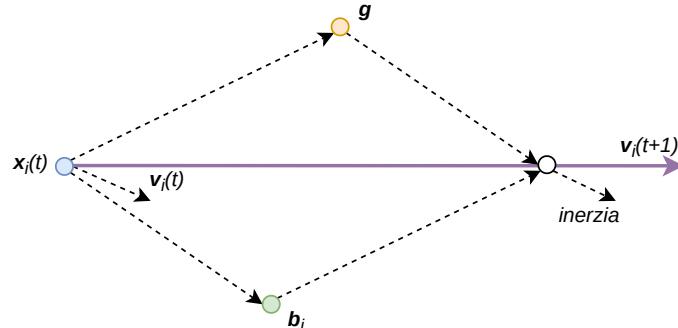
Inoltre il movimento subisce anche l'effetto di un fattore d'**inerzia**, pesato da un fattore w , avendo:

$$inertia = w \cdot \mathbf{v}_i(t)$$

³⁷Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks (Vol. 4, pp. 1942-1948). IEEE.

Figura 5.13: Flowchart del funzionamento di *PSO*.

Approssimativamente si ottiene, per il calcolo di $\mathbf{v}_i(t+1)$, che solitamente è in un intervallo di bound $[\mathbf{v}_{min}, \mathbf{v}_{max}]$:



Non appena tutte le posizioni vengono aggiornate, i valori di fitness di tutte le particelle vengono aggiornati e queste informazioni vengono utilizzate per aggiornare (se necessario) entrambi i \mathbf{b}_i e \mathbf{g} . Si ottiene quindi il flowchart visibile in figura 5.13. *PSO* inoltre richiede da parte dell'utente l'input di diversi *settings*, per il *tuning* dell'algoritmo stesso, i cosiddetti **iper-parametri**:

- il già citato *social factor* $c_{soc} \in \mathbb{R}^+$
- il già citato *cognitive factor* $c_{cog} \in \mathbb{R}^+$
- il già citato peso dell'inerzia $w \in \mathbb{R}^+$ che bilancia le impostazioni di cui sopra. L'inerzia bassa aiuta la ricerca locale, l'inerzia alta aiuta la ricerca globale

- la grandezza dello sciame N
- l'intervallo della velocità $[\mathbf{v}_{min}, \mathbf{v}_{max}]$
- una strategia per la gestione delle particelle che escono dallo spazio di ricerca³⁸
- un algoritmo per la distribuzione iniziale delle particelle nello spazio di ricerca³⁹

È quindi ovvio come tali parametri siano difficili da scegliere e abbiano un fortissimo impatto al punto di vista sia dei risultati che dell'ottimizzazione. Tale lavoro non può essere svolto da chiunque. Basti pensare comunque che già solo $N = 50$ con 1000 iterazioni comporti davvero tante simulazioni ad ogni passo.

Dal punto di vista degli *iper-parametri* si segnala quindi il paper di Nobile et al.⁴⁰ dove si propone un algoritmo di *self-tuning* basato su *logica fuzzy*, detto **Fuzzy Self-Tuning PSO (FST-PSO)**. In pratica con l'uso della logica fuzzy, tramite 15 semplici regole, si dà alle particelle “un po' di intelligenza in più”, calcolando con *FST-PSO* l'inerzia, i due fattori e il range di velocità n *real-time*. Ogni particella adatta i suoi iper-parametri basandosi su due informazioni:

1. il *miglioramento normalizzato di fitness* rispetto all'iterazione precedente, denotato con Φ
2. la distanza dalla soluzione globale migliore, denotata con δ

Tale algoritmo risulta essere la miglior soluzione in ottica *PE*.

Un'applicazione pratica di *FST-PSO* è stata fatta, in collaborazione con l'*Istituto di Oncologia Europeo (IEO)*, in merito alla **leucemia mieloide acuta/Acute Myeloid Leukemia (AML)**, tramite un framework di si-

³⁸Xu, F., Zhang, Y., Hong, W., Wu, K., & Cui, T. J. (2003). Finite-difference frequency-domain algorithm for modeling guided-wave properties of substrate integrated waveguide. *IEEE Transactions on Microwave Theory and Techniques*, 51(11), 2221-2227.

³⁹Cazzaniga, P., Nobile, M. S., & Besozzi, D. (2015, August). The impact of particles initialization in PSO: parameter estimation as a case in point. In 2015 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB) (pp. 1-8). IEEE.

⁴⁰Nobile, M. S., Cazzaniga, P., Besozzi, D., Colombo, R., Mauri, G., & Pasi, G. (2018). Fuzzy Self-Tuning PSO: A settings-free algorithm for global optimization. *Swarm and evolutionary computation*, 39, 70-85.

mulazione chiamato **ProCell**⁴¹. Lo scopo dello studio erano le *ricadute* dell'*AML*, studiando cellule umane di *AML* xenotriplantate nei topi e contrassegnate con proteine fluorescenti verdi per mezzo di lentivirus. Infine per raccogliere istogrammi di fluorescenza si è usata la citometria a flusso, a $t = 0, t = 10$ e $t = 21$ giorni (appunto come anticipato pochi time-points). Si è quindi studiata la *self-proliferation* studiando la distribuzione delle fluorescenze ottenute sperimentalmente (si noti che per ottenere i dati sperimentali bisogna uccidere il topo). Si hanno alcune assunzioni:

- *GFP* si lega all'istone *H2B*: il livello di fluorescenza si dimezza ad ogni divisione cellulare
- i topi sono immunocompromessi: nessuna risposta immunitaria da linfociti, cellule NK, etc...
- livello di fluorescenza basale: se la fluorescenza scende al di sotto di una soglia diventa *GFP-negativa*

Per questa analisi si è quindi creato un nuovo framework di modellazione stocastica per studiare la proliferazione cellulare chiamato appunto **ProCell**. La simulazione richiede diversi parametri:

- il *numero delle sotto-popolazioni* e il *rappporto* (rispetto al totale ???)
- la *media* e la *deviazione standard* della divisione cellulare per ogni sotto-popolazione
- *soglia di GFP-positivity*
- *tempo di simulazione*, in ore

Si ha quindi una simulazione stocastica dove:

- si mantiene uno *stack* di cellule, avendo che le nuove cellule generate da eventi di divisione randomici vengono aggiunte allo *stack*
- le cellule *GFP-negative* sono rimosse dallo *stack*
- l'algoritmo restituisce lo *stack* di cellule fluorescenti a $t = t_{max}$

⁴¹Nobile, M. S., Vlachou, T., Spolaor, S., Cazzaniga, P., Mauri, G., Pelicci, P. G., & Besozzi, D. (2019, July). ProCell: Investigating cell proliferation with swarm intelligence. In 2019 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB) (pp. 1-8). IEEE.

Nel design di questo modello si sono dovute fronteggiare varie problematiche, tra cui l'assenza della maggior parte delle informazioni biologiche, tra cui i tempi di divisione stessi. In pratica nessuno dei parametri era misurabile con esperimenti in *wet-lab*. In output si hanno infine degli istogrammi che dipendono fortemente dalla parametrizzazione del modello, avendo quindi necessità di un buon metodo per la *PE*. Nel dettaglio si è quindi usato *FST-PSO* per lo sviluppo di un sistema di *PE* completamente automatizzato, usando come *fitness function* la **distanza di Hellinger** tra l'istogramma simulato \hat{H}_j^ϑ , ottenuto tramite divisione in sotto-intervalli (i cosiddetti *bin*), normalizzato e creato con la parametrizzazione ϑ , e l'istogramma sperimentale target \hat{T}_j . Formalmente si aveva quindi, con N numero delle sotto-popolazioni (???:)

$$f(\vartheta) = \frac{1}{\sqrt{2}} \sqrt{\sum_{j=1}^N \left(\sqrt{\hat{H}_j^\vartheta} - \sqrt{\hat{T}_j} \right)^2}$$

Questo è per di più un esempio pratico per vedere come non siano solo i parametri cinetici quelli che si possono stimare con un algoritmo di *PE*. Sono stati quindi testati quattro modelli per lo studio della *AML*:

1. la *proliferazione semplice*
2. la *proliferazione semplice* e le *celle quiescenti*, che sono quelle che non vengono uccise dalla chemioterapia. Questo era il sospetto dello *IEO*
3. la *proliferazione lenta*, la *proliferazione veloce* e le *celle quiescenti*
4. la *proliferazione lenta* e la *proliferazione veloce*, avendo avuto il sospetto che fossero le prime quelle a non sopravvivere alla chemioterapia e non le *cellule quiescenti*

I risultati delle simulazioni hanno mostrato come il terzo modello spieghi effettivamente le osservazioni sperimentali dei dati ottenuti tramite le fluorescenze, avendo la minor *distanza di Hellinger* tra i modelli testati e avendo che anche una simulazione con validazione a $t = 21$ giorni abbia un buon fit coi dati sperimentali.

Si segnalano i paper dello studio, con i dettagli degli algoritmi e dei risultati:

- Nobile, M. S., Nisoli, E., Vlachou, T., Spolaor, S., Cazzaniga, P., Mauri, G., ... & Besozzi, D. (2020). cuProCell: GPU-accelerated analysis of cell proliferation with flow cytometry data.

IEEE Journal of Biomedical and Health Informatics, 24(11), 3173-3181.

- Nobile, M. S., Vlachou, T., Spolaor, S., Cazzaniga, P., Mauri, G., Pelicci, P. G., & Besozzi, D. (2019, July). *ProCell: Investigating cell proliferation with swarm intelligence*. In *2019 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)* (pp. 1-8). IEEE. (premiato con il “best paper award”)
- Nobile, M. S., Vlachou, T., Spolaor, S., Bossi, D., Cazzaniga, P., Lanfrancone, L., ... & Besozzi, D. (2019). *Modeling cell proliferation in human acute myeloid leukemia xenografts*. *Bioinformatics*, 35(18), 3378-3386.

5.6.6 Dilation Function

Si ricorda che, in determinati casi, l’uso della *distribuzione uniforme logaritmica* possa fornire risultati migliori qualora si necessiti di un dato campionato in un intervallo prossimo allo 0 (tipico se si parla di costanti cinetiche), avendo che usando una normale *distribuzione uniforme* non si ottiene realmente l’intero range di valori in modo uniforme (avendo che si campiona meno avvicinandosi a 0).

Il team formato dalla professoressa Besozzi, dal prof Nobile e dai loro colleghi hanno quindi cercato un metodo per ottimizzare al meglio la scelta dei parametri cinetici, avendo che essi sono spesso compattati in un intervallo iniziale sulla funzione ottenuta avendo sulle x i valori dei parametri e sulle y i valori del *fitness value* ottenibile con quella parametrizzazione, che poi tende ad andare in uno steady-state. L’idea è quindi quella di “stretchare” i valori sull’asse delle x cambiandone letteralmente la semantica. Infatti si ha che i parametri cinetici ottimali sono spesso molto piccoli e nascosti negli ordini di grandezza più bassi. Con le **dilation functions**, proposte dal prof Nobile et al.⁴² si ha quindi un mapping dei valori cinetici durante la fase di ottimizzazione, letteralmente *dilatando* alcune regioni dello spazio di ricerca, avendo come obiettivo di selezionare quelle regioni che contengono le soluzioni ottimali e di semplificare la fase di ottimizzazione. Si ottiene quindi un

⁴²Nobile, M. S., Cazzaniga, P., & Ashlock, D. A. (2019, June). Dilation functions in global optimization. In *2019 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2300-2307). IEEE.

dilated fitness landscape. Si è permesso quindi di trovare anche quei valori, molto piccoli, che solitamente sono difficili da trovare usando un’euristica di ottimizzazione. Dilatando gli ordini di grandezza più bassi e comprimendo il valori cinetici elevati, si può infatti rivelare la regione ottimale. Grazie a questo metodo si sono migliorate le performance di praticamente tutti gli algoritmi di *PE*.

5.6.7 surF

Il *dilated fitness landscape* può comunque essere molto “ruvido” a causa dell’uso di algoritmi di simulazione stocastici che permettono appunto di simulare anche il *rumore*. Nel contesto della *PE* tale rumore però non sempre risulta essere un vantaggio. Nel 2020 Manzoni, la professoressa Besozzi, il prof Nobile et al⁴³ proposero **Surrogate Fourier modeling of fitness landscapes (surF)**. Si ha che *surF* esegue un campionamento casuale del *fitness landscape* e quindi sfrutta la **trasformata di Fourier discreta (DFT)** per calcolarne gli spettri multidimensionali. Filtrando i componenti ad alta frequenza ed eseguendo la *DFT inversa*, *surF* produce una versione *liscia* del *fitness landscape*, senza appunto la rappresentazione *ruvida* data dal *rumore*. Infatti la *DFT* converte una sequenza finita di campioni equi-distanziati di una funzione, che in questo caso è la *fitness function*, in una sequenza di campioni equi-distanti della *trasformata di Fourier a tempo discreto*, che è una funzione di frequenza a valori complessi. Si ha quindi il passaggio dai campioni alle frequenze. Inoltre la *DFT* è invertibile, permettendo quindi anche il viceversa. Rimuovendo le componenti ad alta frequenza dello *spettro di Fourier* si può quindi rimuovere il rumore dal segnale e utilizzare la *DFT inversa* per ottenere una versione più *liscia* della funzione originale.

La *DFT* può essere generalizzata a N dimensioni ma necessita di campioni equi-distanti, avendo che il numero di campioni necessari esplode in modo esponenziale. Questo problema può comunque essere mitigato mediante campionamento e interpolazione casuali. Si ha del resto che *surF* è limitato in termini dimensionali (al più circa $N = 30$, *dato da verifica, la professoressa non era certa*) anche se si stanno svolgendo studi in merito. Nonostante ciò *surF* permette di semplificare notevolmente i metodi di *PE*.

⁴³Manzoni, L., Papetti, D. M., Cazzaniga, P., Spolaor, S., Mauri, G., Besozzi, D., & Nobile, M. S. (2020). Surfing on fitness landscapes: A boost on optimization by Fourier surrogate modeling. Entropy, 22(3), 285.

Capitolo 6

Rule-Based Modelling

Questo capitolo è relativo al seminario tenuto dalla professor Simone Spolaor.
Si presenta ora il **rule-based modelling** che è una tecnica modellistica a tratti simile al **reaction-based modelling**, anche se molto più verboso, con la differenza che vengono usati in presenza di realtà biologiche che prevedono potenzialmente un numero molto grande (o addirittura non finito) di reazioni, come *signaling*, *trasduction cascade*, *risposta immune*, *polimerizzazioni etc...*

Tali modelli sono quindi *modelli meccanicistici*.

Un altro dei motivi chiave per cui si usa questo tipo di modellazione, al posto di quello meccanicistico basato su reazioni, è per gestire la cosiddetta **modularità**. Infatti le *biomolecole*, specialmente le *proteine*, possono avere multiple funzioni, come di *chinasì* o *binding*. Le *proteine* sono spesso organizzate in modo “modulare”, avendo i cosiddetti *protein domains*. Quindi la *modularità* (che è presente anche in altri fenomeni biologici come la *sugar chain*) è una funzionalità chiave per comprendere il funzionamento delle proteine, in quanto sfruttano la modularità per “localizzare” alcune funzioni all’interno della loro struttura 3D. Tra i *protein domains* abbiamo:

- enzimi
- proteine adattatrici (*adapters*)
- proteine per il docking
- proteine citoscheletliche
- fattori di trascrizione
- fattori di inibizione

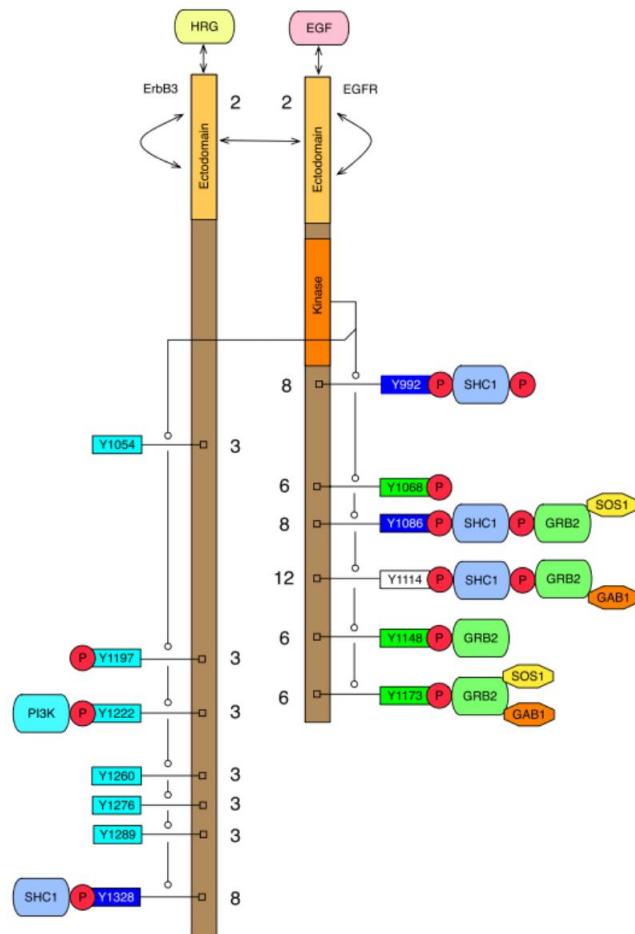
Bisogna inoltre notare come di vari *protein domains* condividano strutture, funzioni, storia evolutiva e sequenze amminoacidiche.

I siti di legame proteico (*protein binding sites*) e le modifiche post-trasduzionali possono essere in stati diversi in due copie della stessa proteina. D’altro canto

descrivere lo stato generale di una proteina potrebbe richiedere l'enumerazione di diversi siti di legame e modifica. Tale numero quindi cresce molto velocemente, in modo proporzionale sia al numero di siti disponibili che del loro stato.

Vediamo un esempio¹.

Esempio 16. Si pensi all'esempio di EGFR, qui rappresentato:



Il recettore EGFR può dimerizzare, ovvero si ha una reazione chimica che coinvolge due entità molecolari identiche (monomeri) per la formazione di una singola entità molecolare, con altre proteine della stessa specie e presenta diversi siti di fosforilazione.

Si ha quindi che:

¹Creamer, Matthew S., et al. "Specification, annotation, visualization and simulation of a large rule-based model for ERBB receptor signaling." BMC systems biology 6.1 (2012): 1-14.

- *EGFR monomero prevede:* $2 \cdot 8 \cdot 6 \cdot 8 \cdot 12 \cdot 6 \cdot 6 = 331776$ stati/specie
- *EGFR Erb83 eterodimerizzazione prevede:* 3.8×10^9 stati/specie
- *EGFR dimerizzazione prevede:* 3.8×10^{12} stati/specie

Già il primo caso è ingestibile dal punto di vista computazionale tramite un modello reaction-based, avendo una specie univoca per ogni stato possibile dal punto di vista dei binding. Si arriverebbe circa a 10^{10} specie, con le quantità iniziali, e con un numero quasi sicuramente maggiore di reazioni, con i vari kinetic rate.

I sistemi *reaction-based* considerano le specie *in massa/bulk*, studiando contemporaneamente più reazioni e non tenendo traccia esplicita dello stato di una singola molecola, quindi è impossibile creare un modello in quanto richiederebbe troppo sforzo umano e tempo macchina. Quindi si ha bisogno di un approccio di modellazione che sia più vicino alle caratteristiche strutturali delle biomolecole avendo quindi i modelli **rule-based** che si pongono ad un livello di astrazione intermedio tra i modelli *reaction-based*, più astratti, e i modelli basati sulla *dinamica molecolare*, meno astratti.

Questo tipo di modellazione è quindi è un formalismo di modellazione che affronta la rappresentazione di *sistemi biologici combinatorialmente complessi*, avendo l'idea di base è quella di specificare le *interazioni proteina-proteina* come **regole** che servono come *generatori di reazioni chimiche e specie*. Le *regole* possono essere viste come una definizione di una *classe di reazioni*, praticamente come una *reazione generalizzata* e specifica le caratteristiche delle proteine che sono richieste per una particolare interazione o che sono influenzate dalla stessa.

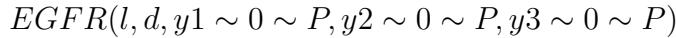
Nel tempo si sono sviluppati vari tool per i modelli *rule-based* tra cui:

- *BioNetGen*, probabilmente il più usato e che verrà introdotto nel capitolo
- *Kappa*, uno dei più complessi in quanto studiato per un uso da parte di matematici più che da biologi/biotecnologi
- *MØD*

6.1 Regole in BioNetGen

Come introdotto nel capitolo si parlerà di **BioNetGen** quindi è bene vedere la sintassi di tale sistema. La prima cosa da vedere è quindi come modellare

una *specie*, che viene modellata come un *oggetto strutturato*, ad esempio:



Dove:

- *EGFR* è il nome della molecola
- *l, d, y1, y2, y3* sono i siti molecolari, che vengono separati da “,”
- $\sim 0 \sim P$ sono i possibili stati di un sito, avendo in questo caso che 0 sta corrisponde al fatto che il sito sia *non fosforilato* mentre *P* che il sito sia *fosforilato*

In questo caso quindi si ha che la molecola *EGFR* ha due siti di binding (*l* e *d*) e 3 siti di fosforilazione (*y1, y2* e *y3*).

Si ha poi che le *regole* definiscono le interazioni tra molecole e comprendono:

- un pattern per riconoscere i reagenti a seconda dei veri siti di binding
- un mapping tra i reagenti e i prodotti
- una costante cinetica per ogni verso del mapping

Si vede un semplice esempio, con una reazione reversibile (quindi con due costanti cinematiche):



Dove:

- *EGR(r) + EGFR(l)* sono i reagenti con specificati i siti di legame necessari (*r* e *l*)
- *EGF(r!1).EGFR(l!1)* è il prodotto, avendo che con . si specifica il complesso. Con la sintassi ! <label> si specifica il *legame* tra le molecole, avendo che si usa la label condivisa, in questo caso 1, per specificare tra quali siti è avvenuto il legame

Nell'esempio si ha quindi che *EGF* si lega al suo recettore per mezzo del suo sito di legame (*l*), con costante cinetica *k1*, e viene rilasciato dal suo recettore con costante cinetica *k2*.

D'altro canto potrei avere non solo la produzioni di complessi ma anche eventi di fosforilazione, come ad esempio:



Dove $EGFR$ è *fosforilato*, in un sito $y1$ disponibile, da una chinasi, con costante cinetica $k3$.

In pratica le *regole* identificano le specie che hanno le caratteristiche necessarie per subire la trasformazione da reagenti a prodotti, facendo quindi una sorta di **pattern matching**. Da una *regola* si ha quindi la generazione di più possibili reazioni, tutte con la stessa costante cinetica, derivanti dalla *regola* stessa. La generazione di reazioni da una regola è informalmente spiegata con il “motto”: *don’t write don’t care!*, avendo che gli elementi non menzionati nella regola possono essere in qualsiasi stato, pertanto, una *regola* può generare reazioni che coinvolgono molte diverse “specie”. Nell’ultima regola, per $EGFR$, ad esempio, si è specificato solo lo stato di $y1$, gli altri 4 stati (se si riprende il primo esempio della sezione) possono essere, per assunzione, in qualsiasi stato. Quindi potrei avere molte specie che potrebbero partecipare alla reazione, ad esempio:

- $EGFR(d!1).EGFR(d!1)$
- $EGFR(y1 \sim 0)$
- $EGFR(y2 \sim P, y3 \sim P)$

In quanto tutte hanno il sito l libero.

Si noti che se modellassi ogni volta esplicitamente lo stato di tutti siti otterrei un modello *reaction-based* quindi si specificano solo gli stati necessari, imponendo di fatto una sorta di “vicolo” alle reazioni generabili, e si lascia che sia il sistema a generare le reazioni.

Per convenzione, le interazioni rappresentate in una regola sono considerate indipendenti dal contesto molecolare (i siti di legame etc...) che non è esplicitamente indicato nella regola stessa. Il fatto che tutte le reazioni generate da una regola condividano la stessa costante cinetica è un’assunzione che riflette l’organizzazione strutturale delle proteine, in cui le funzioni differenti sono isolate in domini differenti. Inoltre la generazione delle reazioni, che avviene a seconda dello stato corrente grazie all’uso del *pattern matching* sulle regole, segue la *cinetica mass-action* anche se i vari sistemi sopra citati, tra cui *BioNetGen*, supportino anche altre leggi cinematiche.

6.2 Simulazioni

Anche in questo caso, prima di effettuare una *simulazione*, bisogna specificare lo stato iniziale, come numero di molecole o concentrazione (a seconda della tipologia di simulazione che si vuole come nei modelli *reaction-based*) di ogni singola molecola (e di ogni possibile combinazione di binding site iniziale

???)

Si può già ora intuire come esista una *relazione 1:1* tra i modelli *rule-based* e *reaction-based* ma si hanno modelli *rule-based* non modellabili come *reaction-based* in quanto esistono regole che generano *reazioni infinite*, come nel caso delle *polimerizzazioni*, in quanto si costruisce una catena di *polimeri* di lunghezza arbitraria (che nei *reaction-based* dovrebbe essere fissata a priori). Infatti ci sono molte reti di reazione biochimica in natura che sono potenzialmente non finite, tra cui:

- allungamenti della catena polimerica
- N-glicosilazione delle proteine
- polimerizzazione di actina e microtubuli
- sintesi delle sequenze nucleotidiche

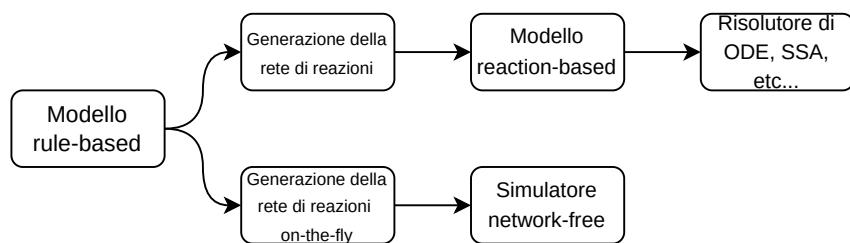
Per le simulazioni si hanno quindi due approcci:

1. l'**approccio network-based**, detto anche **generate-first** o **approccio indiretto**, dove si ha la generazione delle reazioni prima della simulazione. In questo caso, infatti, la rete di specie e reazioni viene generata prima della simulazione, attraverso l'applicazione iterativa delle regole fino a quando non viene soddisfatta un criterio di stop arbitrario, specificato a priori, o si ha che non vengono più generate nuove reazioni. In pratica questo approccio genera un modello *reaction-based*, che può essere simulato con uno qualsiasi dei algoritmi di simulazione già noti, ovvero *algoritmi di integrazione*, *SSA*, *τ -leaping*, etc...
2. l'**approccio network-free**, detto anche **on-the-fly** o **approccio diretto**, dove si ha la generazione delle reazioni durante la simulazione stessa. **Questa strategia è necessaria in presenza di regole che possono generare infinite reazioni**, in quanto l'approccio *network-based* non è applicabile in tali casi. In generale in questo caso la rete di specie e reazioni viene generata durante la simulazione. A partire da una condizione iniziale, la rete viene generata quando nuove specie diventano disponibili e quindi si “perde” il supporto agli algoritmi visti per i modelli *reaction-based*, che si è visto sono da usare per l'*approccio network-based* ma si hanno algoritmi specifici come **NFsim** (**Network-free**

simulator)². *NFsim* è una versione *rule-based* e generalizzata dell’*algoritmo SSA* e ne produce la stessa dinamica. L’algoritmo, che in input prende il numero di molecole per ogni specie essendo una simulazione stocastica, si compone di cinque step:

- (a) calcolare le *propensity function* per ogni regola, dati gli attuali stati molecolari
- (b) campionare il tempo d’attesa per la prossima reazione, come in *SSA*
- (c) selezionare la prossima regola (e non la prossima reazione come in *SSA*)
- (d) selezionare le molecole reagenti come indicato dalla regola scelta
- (e) aggiornare lo stato del sistema e il tempo

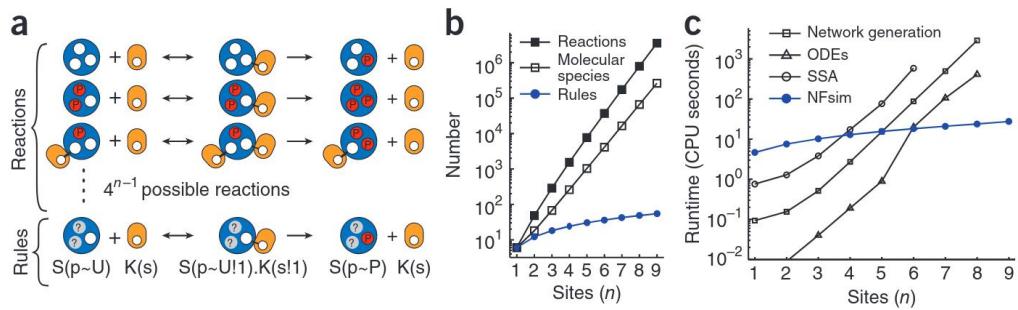
Si ha quindi il seguente schema riassuntivo:



Riassumendo l’*approccio network-free* viene usato per reti grandi e non limitate, per complessi proteici con numerosi siti di modifica, processi di polimerizzazione etc...

Del resto anche se la generazione della rete di reazioni è fattibile e limitata, la generazione di grandi reti (con magari componenti con tanti siti di binding) è computazionalmente pesante, sia in tempo che in memoria, al punto che nel tempo in cui si genera la rete per l’approccio *network-based*, dovendo poi seguire la vera e propria simulazione, si è quasi conclusa la simulazione con l’approccio *network-free*, che, dal punto di vista temporale, cresce in modo lineare all’aumentare dei siti (mentre l’altro approccio cresce in modo esponenziale). Si noti infine che se il numero di specie e reazioni cresce in modo esponenziale all’aumentare dei siti il numero di regole cresce in modo lineare (???). Il tutto è visualizzabile in figura 6.1².

²Sneddon, M., Faeder, J. & Emonet, T. Efficient modeling, simulation and coarse-graining of biological complexity with NFsim. Nat Methods 8, 177–183 (2011). <https://doi.org/10.1038/nmeth.1546>

Figura 6.1: Grafici per confrontare i due approcci per modelli *rule-based*.

6.3 Software

Si sono già introdotti alcuni tool. Approfondendo si ha che *BioNetGen* funziona in combinazione a *RuleBender* e si ha che esiste *PySB* (*Python bindings for BioNetGen*). Inoltre si ha il supporto aggiuntivo di altri software, per stima di parametri, visualizzazione etc..., tra cui:

- *PySB-INDRA*
- *Srsim*
- *BioNetFit*
- *BioUML*
- *SBML Multi*
- *VirtualCell*
- *Parts&Pools*

Come esempio vediamo il modello per il *repressilator* in *RuleBender*, il cui codice è presente alla listing 2 e la visualizzazione alle figure 6.2 e 6.3.

Listing 2 Codice per il repressilator in rulebender

```

begin parameters
    Na      6.022e23      # Avogadro's [mol^-1]
    V       1.4e-15       # Cell volume [L]
    #
    c0      1e9           # M^-1 s^-1
    c1      224           # s^-1
    c2      9              # s^-1
    c3      0.5           # s^-1
    c4      5e-4          # s^-1
    c5      0.167          # s^-1
    c6      ln(2)/120     # s^-1
    c7      ln(2)/600     # s^-1
    #
    tF      1e-4          # telegraph factor
    rF      1000          # rna factor
    pF      1000          # protein factor
end parameters

begin molecule types
    Null()
    gTetR(lac,lac)
    gCI(tet,tet)
    gLaci(cI,cI)
    mTetR()
    mCI()
    mLaci()
    pTetR(cI)
    pCI(lac)
    pLaci(tet)
end molecule types

begin seed species
    Null()
    gTetR(lac!1, lac!2) . pLaci(tet!1) . pLaci(tet!2)      1
    gCI(tet!1, tet!2) . pTetR(cI!1) . pTetR(cI!2)      1
    gLaci(cI!1, cI!2) . pCI(lac!1) . pCI(lac!2)      1
    mTetR()                                         3163
    mCI()                                           6819
    mLaci()                                         129
    pTetR(cI)                                       183453
    pCI(lac)                                         2006198
    pLaci(tet)                                      165670
end seed species

begin observables
    Molecules   pTetR   pTetR(cI)
    Molecules   pCI     pCI(lac)
    Molecules   pLaci   pLaci(tet)
    Molecules   NULL    Null()
end observables

begin reaction rules
    gTetR(lac,lac) + pLaci(tet) <-> gTetR(lac!1, lac) . pLaci(tet!1)      c0/Na/V*tF/pF, c1*tF
    gTetR(lac!+, lac) + pLaci(tet) <-> gTetR(lac!+, lac!1) . pLaci(tet!1)  c0/Na/V*tF/pF, c2*tF
    gTetR(lac, lac) -> gTetR(lac, lac) + mTetR()                         c3*rF
    gTetR(lac!+) -> gTetR(lac!+) + mTetR()                               c4*rF
    mTetR() -> mTetR() + pTetR(cI)                                         c5/rF*pF
    mTetR() + Null() -> Null()                                            c6
    pTetR(cI) + Null() -> Null()                                            c7

    gCI(tet,tet) + pTetR(cI) <-> gCI(tet!1,tet) . pTetR(cI!1)      c0/Na/V*tF/pF, c1*tF
    gCI(tet!+, tet) + pTetR(cI) <-> gCI(tet!+, tet!1) . pTetR(cI!1)  c0/Na/V*tF/pF, c2*tF
    gCI(tet,tet) -> gCI(tet,tet) + mCI()                                c3*rF
    gCI(tet!+) -> gCI(tet!+) + mCI()                                 c4*rF
    mCI() -> mCI() + pCI(lac)                                         c5/rF*pF
    mCI() + Null() -> Null()                                            c6
    pCI(lac) + Null() -> Null()                                            c7

    gLaci(cI,cI) + pCI(lac) <-> gLaci(cI!1,cI) . pCI(lac!1)      c0/Na/V*tF/pF, c1*tF
    gLaci(cI!+,cI) + pCI(lac) <-> gLaci(cI!+,cI!1) . pCI(lac!1)  c0/Na/V*tF/pF, c2*tF
    gLaci(cI,cI) -> gLaci(cI,cI) + mLaci()                           c3*rF
    gLaci(cI!+) -> gLaci(cI!+) + mLaci()                            c4*rF
    mLaci() -> mLaci() + pLaci(tet)                                     c5/rF*pF
    mLaci() + Null() -> Null()                                         c6
    pLaci(tet) + Null() -> Null()                                         c7
end reaction rules

generate_network({overwrite=>1})
simulate_ode({t_end=>4e4,n_steps=>4e2,atol=>1e-12,rtol=>1e-12})

```

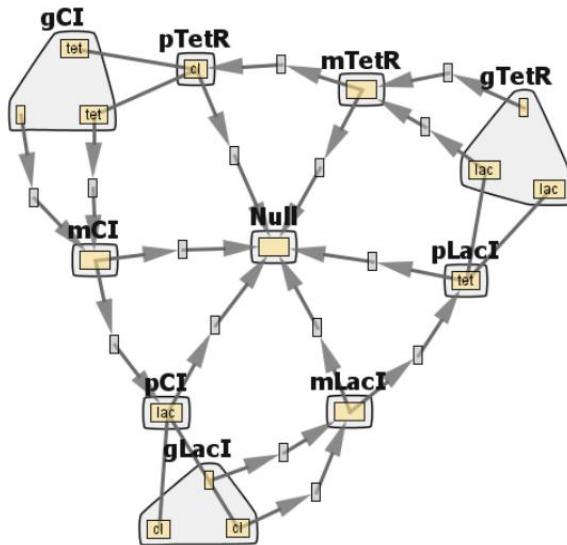


Figura 6.2: *Contact map* del *repressilator* prodotta con *RuleBender*. Si noti la modellazione dei vari siti di binding.

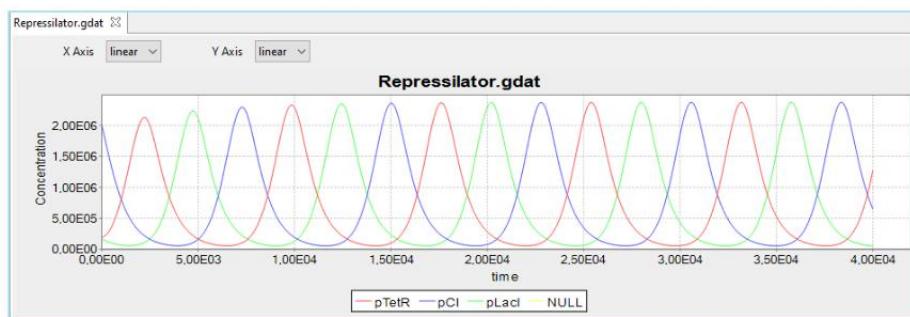


Figura 6.3: Output della simulazione del *repressilator* prodotta con *RuleBender*

Capitolo 7

Constraint-Based Modelling

Questo capitolo è relativo al seminario tenuto dalla professoressa Chiara Damiani.

Si introduce ora l'ultima categoria di modellazione del corso, parlando di **modelli constraint-Based**, che sono modelli che studiano principalmente il metabolismo cellulare, tramite reti metaboliche. Rispetto ai modelli meccanicistici si perde informazione sulle concentrazioni e sul transiente ma si ha solo sullo steady-state (ricordando che lo *stato transiente* è lo stato di transizione che corrisponde al momento della reazione in cui i reagenti stanno rompendo i legami per diventare prodotti e, nello stesso tempo, si stanno formando i nuovi legami che permettono la formazione del risultato della reazione).

Con **metabolismo** si intende l'insieme dei processi che le cellule usano per produrre energia e semplici componenti a partire nutrienti, con i processi *catabolici*, e per produrre macromolecole come le proteine o il DNA a partire da componenti più semplici. Le cellule sane **ossidano** il glucosio, usando-lo quindi insieme all'ossigeno per produrre energia in forma di *ATP*. Invece questo processo è alterato in molte malattie, come nel **cancro**, dove si ha il **Warburg effect**, dove le cellule cancerogene, per soddisfare i propri requisiti metabolici (che sono diversi dal caso di cellule sane), producono molta *biomassa* per proliferare velocemente. In questo caso non ossidano il glucosio ma lo si **fermenta**, convertendo direttamente il glucosio in lattato. Questa soluzione è comunque meno efficiente ma permette alle cellule cancerogene di produrre energia senza bisogno di ossigeno. Studiare il metabolismo permette quindi di comprendere come produrre nuove cure, potendo targettare il metabolismo stesso, provando anche a cambiare il fenotipo delle cellule cancerogene.

Studiare il metabolismo non è importante solo lato “salute” ma anche per fini **bioindustriali**. Ad esempio si possono usare i *batteri* per produrre compo-

Capitolo 7. Constraint-Based Modelling

sti utili a livello industriale, come vitamine, energia da scarti, biocarburanti, farmaci etc... Bisogna quindi ingegnerizzare il metabolismo per ottimizzare il tutto e ottenere una grande produzione di quanto desiderato.

Possiamo pensare al metabolismo come ad una mappa, composta da reazioni che trasformano reagenti in prodotti. Questa classe di modelli è fortemente basata sulla chimica. In questa mappa si ha inoltre il “traffico”, ovvero i **flussi metabolici**, ovvero il *flusso dei metaboliti nella rete metabolica*. Possiamo regolare tali flussi attraverso un modello *constraint-based*.

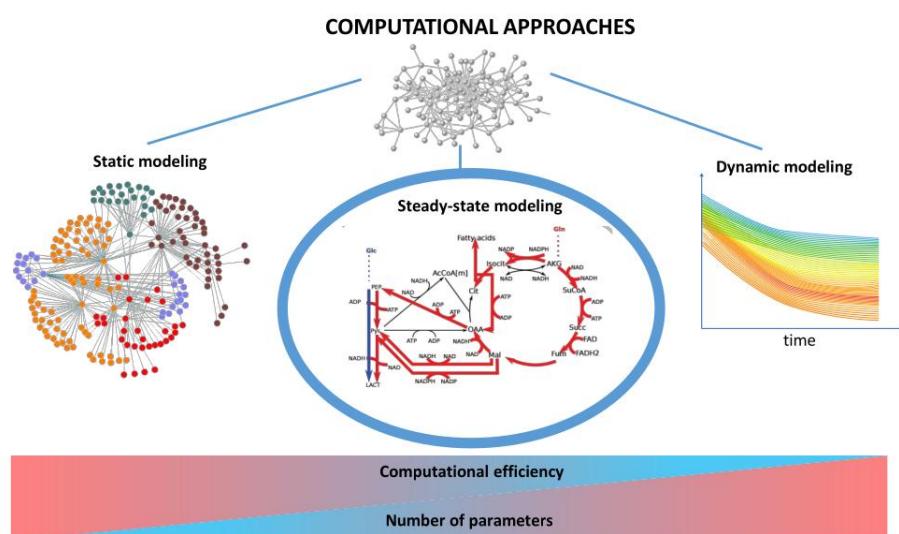
Serve un modello computazionale per studiare il metabolismo sia per predizione che per il fatto che al momento non si ha una tecnica di *wet-lab* per misurare direttamente i *flussi metabolici*, quindi il rateo di una reazione per unità di tempo, in modo efficiente con alta produzione di data. Non si ha ancora una tecnologia *high-throughput* dedicata. In *wet-lab* possono però studiare le concentrazioni all'interno del metabolismo, che è un'informazione “statica”.

Formalmente:

$$flux = rate_{forward\ reaction} - rate_{backward\ reaction}$$

Inoltre è praticamente impossibile calcolare il rateo dei flussi, ovvero il *flusso-soma*, a partire dalla quantità delle componenti del *metaboloma*.

Possiamo riassumere gli approcci computazionali per lo studio del metabolismo in questa immagine¹:



¹Cazzaniga, P.; Damiani, C.; Besozzi, D.; Colombo, R.; Nobile, M.S.; Gaglio, D.; Pescini, D.; Molinari, S.; Mauri, G.; Alberghina, L.; Vanoni, M. Computational Strategies for a System-Level Understanding of Metabolism. *Metabolites* 2014, 4, 1034-1087.

Dove ovviamente in questo capitolo parleremo della soluzione centrale ma, come specificato a destra, si potrebbero usare modelli meccanicistici per studiare i flussi ad ogni istante di tempo ma questo è computazionalmente oneroso, essendo la rete metabolica molto grande, con decine di migliaia di reazioni di cui per di più non si conoscono tutti i *kinetic rates*. A sinistra si ha invece un modello statico tramite rete dove si può studiare la topologia, che nel caso di una rete metabolica è la stessa per ogni essere vivente praticamente, la quale non permette lo studio dei flussi.

Si ha inoltre che variazioni a livello dei metaboliti non si riflettono linearmente in variazioni sui flussi metabolici. Anche per questo servono tali modelli. I modelli *constraint-based* sono anche detti **modelli steady-state** e sono quindi la soluzione migliore per studiare i flussi, anche perché non serve conoscere i parametri cinetici. In pratica quindi sono contemporaneamente modelli dinamici ma che sono studiati in steady-state. La filosofia dietro questi modelli è quella di escludere tutti i fenotipi che sono impossibili e più cose si escludono e più ci si avvicina a qualcosa di “vero”.

La prima cosa che si vuole escludere sono tutte le distribuzioni di flussi metabolici che non sono consistenti con l'assunzione di *steady-state*, che implica che la derivata nel tempo della concentrazione del metabolita sia nulla, ovvero, indicando con p la concentrazione di un certo metabolita:

$$\frac{\partial p}{\partial t} = 0$$

Quindi le concentrazioni dei metaboliti sono costanti, mentre la produzione ed il consumo dei metaboliti all'interno della cellula sono equilibrati. Per giustificare questa sorta di *pseudo steady-state* si sfrutta il fatto che il metabolismo è molto più veloce, avendo un gran numero di molecole, rispetto ad altri processi cellulari (come l'espressione genica) e quindi a lungo termine nessun metabolita può accumularsi o esaurirsi. Quindi eliminiamo tutte le distribuzioni di flusso che sono impossibili date queste premesse. Questa ipotesi ci porta quindi al primo vincolo: **vincolo di mass balance**.

Altri vincoli sono relativi a:

- *vincoli genetici*, ad esempio se un gene viene espresso o se si ha la codifica di un certo enzima (avendo che se non c'è un gene/enzima non avviene una certa reazione)
- *vincoli ambientali*, ad esempio vincolo sui nutrienti (se non ci sono certi nutrienti per una cellula non si hanno certe reazioni o si hanno altri vincoli, tipo se si sa che una cellula cresce in assenza di ossigeno allora si eliminano tutte le distribuzioni di flusso che richiedono ossigeno)

- *vincoli termodinamici*, ad esempio avendo reazioni irreversibili (escludendo tutti i casi dove si ha un flusso nell'ordine inverso a quello possibile)
- *vincoli sui flussi ammessi e sulla loro capacità*, basati su evidenze sperimentali, eliminando i casi che non sono consistenti con le evidenze sperimentali
- *altri vincoli*, dettati dalle varie omiche (*trascrittomica, proteomica e metabolomica*), che si usano anche per ridurre il numero di stati raggiungibili, aggiungendo vincoli

Con questi modelli si risponde quindi a domande del tipo:

- come diverse distribuzioni di flusso sono collegate al tasso di crescita o alla produzione di composti? Un esempio è confrontare *TCA ciclico* con *TCA a rami* e vedere cosa succede al rateo di proliferazione
- in che modo le condizioni al contorno influenzano le distribuzioni del flusso? Ad esempio se su ha poca glutamina nell'ambiente la cellula usa *TCA ciclico* ma se ne aumenta al concentrazione cosa succede? L'ambiente esterno gioca un ruolo fondamentale nel metabolismo

7.1 Flux Balance Analysis

Il metodo più utilizzato (ma non è l'unico anche se ne parla in questo corso) parlando di *constraint-based models* è la **flux balance analysis (FBA)**, usata principalmente per l'*ingegnerizzazione metabolica*, che assume ottimalità rispetto alle funzioni metaboliche. L'idea dietro la *FBA* è quindi quella che, oltre i vincoli, si assume che la cellula lavori in modo ottimale dal punto di vista delle funzioni metaboliche (la cellula vuole massimizzare/minimizzare qualcosa, come il rateo di proliferazione, la produzione di massa, etc.). In questo cointesto è quindi facile escludere tutti gli steady-state che non sono ottimali. La *pipeline* della *FBA* prevede, come alla fine per gli altri modelli in *systems biology*:

1. ricostruzione della rete metabolica
2. simulazione della rete
3. validazione della rete

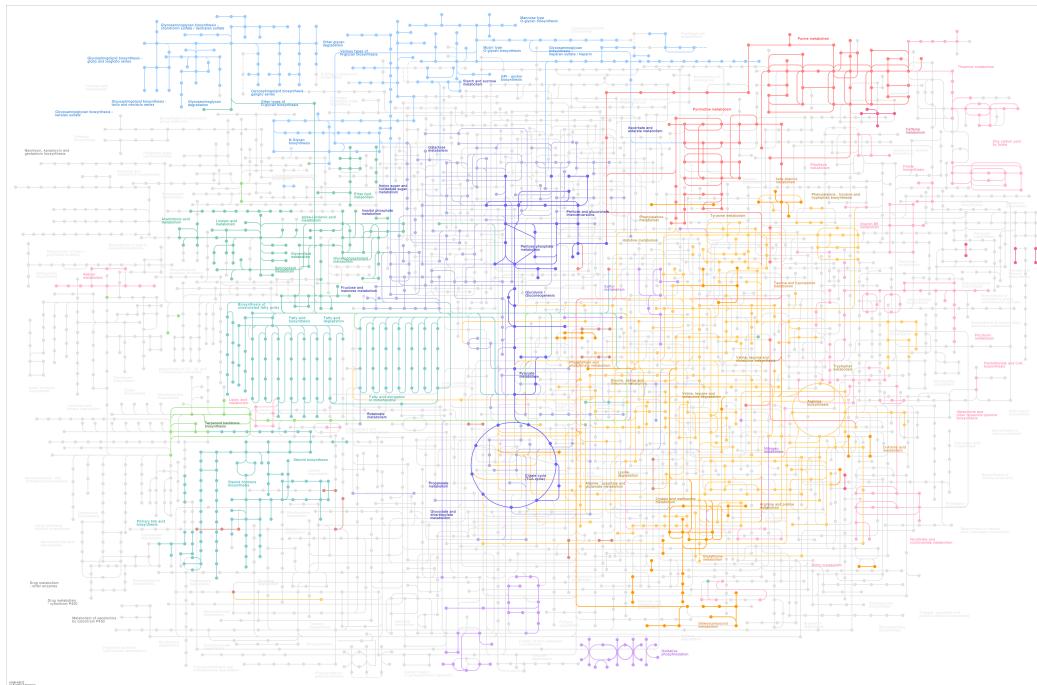


Figura 7.1: Rappresentazione della *human genome wide (GW) metabolic network* disponibile sul database *KEGG*.

4. tuning della rete
5. esperimenti *in silico* dopo la produzione di un modello validato

I tre step centrali sono iterati fino al raggiungimento di risultati validi.

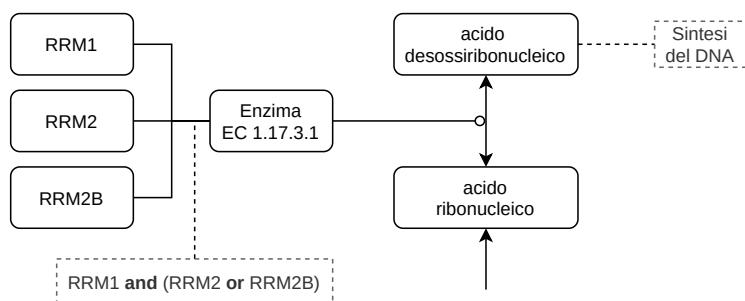
7.1.1 Ricostruzione della Rete Metabolica

Si parte dell'idea che la *human genome wide (GW) metabolic network* consta di:

- ~ 10600 reazioni
- ~ 5835 metaboliti
- ~ 2248 geni

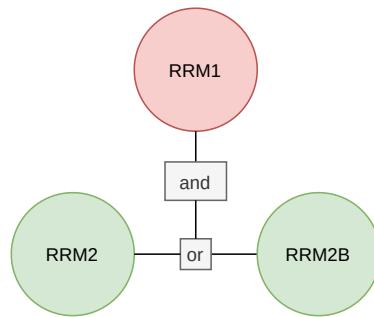
Come visibile in figura 7.1, dove si hanno tutti i metaboliti collegati da tutte le reazioni metaboliche, la situazione è estremamente complessa. Molte organizzazioni comunque hanno ricostruito, previa duro lavoro manuale (in

quanto è un processo difficilmente automatizzabile e parte dalla letteratura in campo biochimico e dalle annotazioni genomiche) durato decenni, tali reti che sono disponibili su vari database, tra cui *KEGG* o *Brenda*. Se si conosce un gene si conosce la proteina codificata (e il suo dominio proteico) da tale gene. A quel punto la si cerca le reazioni legate alla proteina, se ne studia la stechiometria (con la direzionalità, sapendo se la reazione è reversibile o meno). Il tutto si complica col fatto che più geni possono codificare per la stessa proteina, avendo che fortunatamente di solito le relazioni tra tali geni sono codificate, con le *Gene/Protein/Reaction association*. Tali associazioni sono modellate tramite regole logiche. Ad esempio se ho due enzimi che possono alternativamente catalizzare una reazione uso un *or logico* (\vee) tra i corrispondenti geni mentre se mi servono entrambi, magari per ottenere un complesso, un *and logico* (\wedge). Una stessa reazione o uno stesso evento biologico può infatti essere rappresentato in modo più granulare o più astratto, anche se sottostante si ha comunque lo stesso comportamento. Ad esempio potremmo avere la seguente rappresentazione più granulare:



Dove si ha una regola booleana per l'enzima/gene che coinvolge l'enzima **RRM1** (*Ribonucleotide Reductase Large Subunit*), che è una sub-unità catalitica, l'enzima **RRM2** (*Ribonucleoside-diphosphate reductase subunit*) e l'enzima **RRM2B** (*p53-Inducible Ribonucleotide Reductase subunit*), dove le ultime due sono sub-unità regolatorie isoformi. Si nota che l'uso di *funzioni booleane* rende semplice il passaggio dalla modellazione astratta al software.

Tutto questo, che non approfondiamo dal punto di vista biologico, però può essere rappresentato in modo più astratto nel seguente modo, ad esempio:

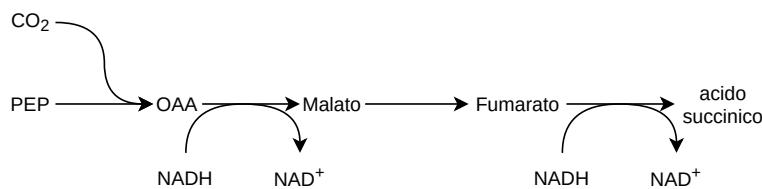


dove si ha astrazione a *livello molecolare*, a cui si potrebbe comunque aggiungere la conformazione delle proteine.

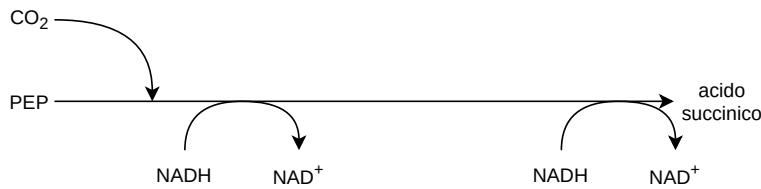
Si procede quindi alla creazione, a partire da tali modelli *genome wide*, dei **Core Reaction Models**. Tali modelli vengono ottenuti semplificando i modelli complessi e “riassumendo” le reazioni, ottenendo quindi modelli che sono:

- curati manualmente
- pronti per la simulazione
- “a grana grossa”, con circa 100 reazioni
- utili per la stima della distribuzione del flusso
- utili per l’identificazione dei principi cardine del sistema
- specifici per un certo aspetto del metabolismo
- potenzialmente contengono meno errori (la *GW network* può contenere errori a causa magari di fretta nelle pubblicazioni etc...). Magari parlando di cancro, ad esempio, ci si concentra sul *carbon metabolism*, dedicato alla produzione di energia e biomassa

In pratica si prende un pezzo di pathway, ad esempio:



e si “ignorano” gli step intermedi (avendo che tra essi il flusso sarà identico), compattando la rappresentazione, semplificando le equazioni e la complessità computazionale, ottenendo:



che presenta quindi, come *steichiometria globale*, solo:

$$-PEP - CO_2 - 2NADH + Succint\ Acid + 2NAD^+ = 0$$

Questo processo è detto **reaction lumping**.

Per scegliere quindi tra i **genome wide models** e i **core models** si utilizza il principio del **rasoio di Occam**, cercando di mediare da un lato per il livello dei dettagli, il costo computazionale, gli errori e gli eventuali cicli termodinamici infattibili e dall'altro mediando lo studio dello spazio di fattibilità, la cura manuale del modello, la “simulabilità” e la visualizzazione del modello stesso. Se il processo della costruzione del **core model** è ben fatto si ottengono con un centinaio di reazioni risultati praticamente analoghi a quelli che si avrebbero con 7000 e passa reazioni.

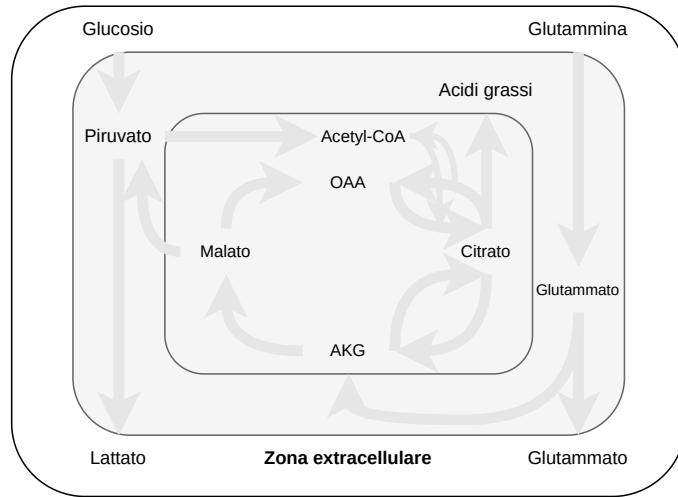
7.1.2 Simulazione della Rete

Si passa quindi alla **simulazione**, sfruttando la *FBA* per predire le distribuzioni dei flussi.

Si hanno 5 step principali parlando di *FBA*:

1. definizione di un *modello metabolico*
2. determinazione di una *matrice stechiometrica S*, che la rappresentazione matematica del modello
3. definizione dei vincoli del modello (*vincoli di sistema, steady-state e capacità massima per i flussi, etc...*)
4. definizione di una *funzione obiettivo Z*
5. ottimizzazione del sistema mediante *programmazione lineare*, che userà i vincoli definiti in 3) sulla funzione definita in 4)

Come esempio giocattolo (dove con le frecce grigie si intende l'assenza di flusso) si userà:



Nell'immagine per quanto poco chiaro si ha che servono sia **Acetyl-CoA** che **OAA** per ottenere il **citrato**, che produce sia **Acetyl-CoA** che **OAA**. Inoltre il rettangolo più interno corrisponde alla zona mitocondriale mentre quello intermedio al *citosol* (???).

Si hanno quindi 14 reazioni, $|R| = n = 14$:

- | | |
|-----------------|-------------------|
| 1. AKG -> Cit | 8. Gln -> Glu |
| 2. AKG -> Mal | 9. Glu -> AKG |
| 3. AcCoA -> Cit | 10. Glu -> GluOut |
| 4. Cit -> AKG | 11. Mal -> Pyr |
| 5. Cit -> FA | 12. Mal -> OAA |
| 6. Cit -> OAA | 13. Pyr -> AcCoA |
| 7. Glc -> Pyr | 14. Pyr -> Lact |

e 12 metaboliti, avendo quindi $|M| = m = 12$:

- | | | | |
|----------|--------|-----------|---------|
| 1. AKG | 4. FA | 7. Glu | 10. Mal |
| 2. AcCoA | 5. Glc | 8. GluOut | 11. OAA |
| 3. Cit | 6. Gln | 9. Lact | 12. Pyr |

A tale sistema corrisponde la seguente matrice S :

M/R	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	-1	-1	0	1	0	0	0	0	1	0	0	0	0	0
2	0	0	-1	0	0	1	0	0	0	0	0	0	1	0
3	1	0	1	-1	-1	-1	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	-1	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	-1	0	0	0	0	0	0
7	0	0	0	0	0	0	0	1	-1	-1	0	0	0	0
8	0	0	0	0	0	0	0	0	0	1	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	1
10	0	1	0	0	0	0	0	0	0	0	-1	-1	0	0
11	0	0	-1	0	0	1	0	0	0	0	0	1	0	0
12	0	0	0	0	0	0	1	0	0	0	1	0	-1	-1

Avendo che S :

- è una matrice $m \times n$
- descrive la relazione tra i metaboliti e i prodotti, descrivendo quali metaboliti a quale quantità sono necessari per creare un prodotto specifico. Ogni valore della matrice è quindi il coefficiente stichio-metrico, avendo il segno negativo per il consumo, avendo quindi che il metabolita è un reagente, e il segno positivo quando è un prodotto. Dal punto di vista dei flussi la secrezione ha segno positivo e ha la freccia in entrata di un metabolita mentre l'uptake funziona all'opposto, segno negativo e freccia in uscita
- è di fatto una *matrice sparsa*

Al momento però si ha un sistema chiuso se non si rappresentano nella matrice anche gli influssi esterni, ad esempio di glutamina, ottenendo una colonna aggiuntiva, per la reazione GLN \leftrightarrow (\leftrightarrow è una convezione per le reazioni di scambio):

M/R	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	-1	-1	0	1	0	0	0	0	1	0	0	0	0	0	0
2	0	0	-1	0	0	1	0	0	0	0	0	0	1	0	0
3	1	0	1	-1	-1	-1	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	-1
7	0	0	0	0	0	0	0	1	-1	-1	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
10	0	1	0	0	0	0	0	0	0	0	-1	-1	0	0	0
11	0	0	-1	0	0	1	0	0	0	0	0	1	0	0	0
12	0	0	0	0	0	0	1	0	0	0	1	0	-1	-1	0

Senza un influsso, in pratica senza i nutrienti, dall'esterno il sistema chiuso ha una soluzione: lo steady-state tutti i flussi nulli. Si noti inoltre che in pratica la *glutamina* “appare dal nulla”. Questo bisogna farlo per tutti i metaboliti che appaiono in reazioni di scambio (quindi anche glucosio, lattato e glutammato, anche se gli ultimi due avranno segno positivo nella matrice). Per formalizzare il problema si ha quindi un vettore di flussi, con un flusso per ogni reazione:

$$\vec{v} = [v_1, \dots, v_{n(i)}, b_1, \dots, b_{n(e)}]$$

dove:

- i v_j sono i flussi del sistema chiuso, ovvero i flussi interni, e sono in totale $n(i)$
- i b_j sono i flussi per le reazioni di scambio e sono in totale $n(e)$

Tale vettore di flussi è quindi l'incognita e sappiamo che allo steady-state le derivate dei vari metaboliti ($A \dots E$) sono nulle.

Si hanno quindi:

- la matrice stoichiometrica S , dove le colonne sono le reazioni (magari nel caso semplice *reazione → prodotto*) e le colonne sono i metaboliti
- lo steady state $S \cdot \vec{v} = [\frac{dA}{dt}, \dots, \frac{dE}{dt}] = 0$. Ogni \vec{v} tale che $S \cdot \vec{v} = 0$ appartiene allo spazio nullo/*kernel* di S . Questo spazio contiene tutte le possibili soluzioni di flussi per la rete metabolica allo stato stazionario. La dimensione dello spazio nullo è determinata dal numero di variabili libere nell'insieme originale di equazioni lineari che è chiamato *rango della matrice stechiometrica*
- i vincoli di flusso:

$$v_{min} \leq v_i \leq v_{max}$$

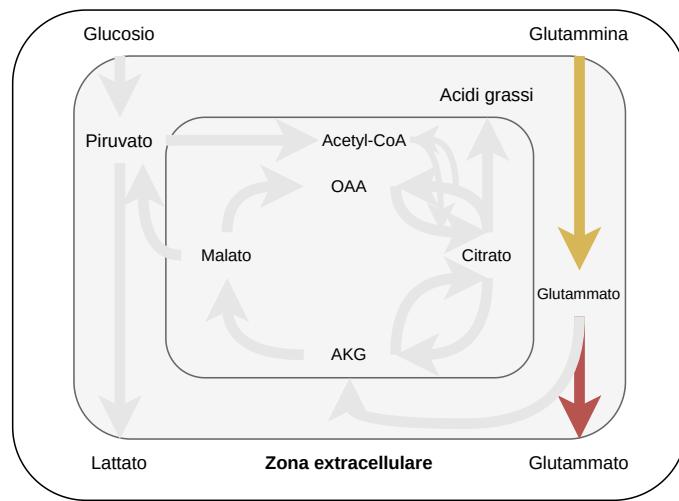
che rappresentano diversi tipi di flusso:

- *flussi in entrata*, tramite vincoli sui nutrienti
- *flussi interni*, tramite vincoli termodinamici e vincoli sul tasso sul *reaction rate*
- *flussi di secrezione*, per rappresentare che il metabolismo è in grado di accumulare, rappresentando quindi la crescita e i suoi vincoli

Per dire che una reazione è irreversibile, valendo solo in segno positivo (reazione forward) avrò che $0 \leq v_i \leq v_{max}$, all'opposto (reazione backward) avrei $v_{min} \leq v_i \leq 0$. I flussi di scambio sono i più importanti da modellare

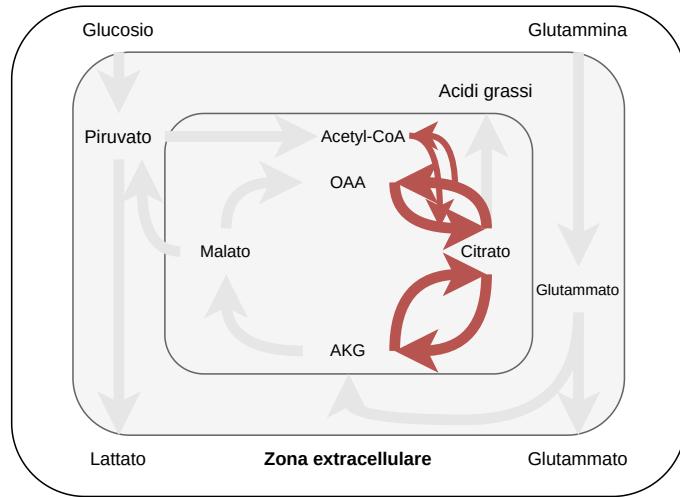
Il flusso è quindi la quantità di sostanza che viene trasformata per unità di tempo.

Una volta fatti i vari conti conti con al programmazione lineare otteniamo un'area dei **possibili fenotipi**, ovvero l'area delimitata dai vincoli stessi. Prendiamo ora un piccolo esempio:



dove eventuali frecce gialle rappresentano il flusso, quindi la quantità di sostanza trasformata per unità di tempo, mentre se rosse rappresentano il massimo flusso possibile (questa notazione può variare, ad esempio usando frecce non piene per il primo caso, avendo che la grandezza della freccia grigia rappresenterebbe il flusso massimo possibile).

Ovviamente, a seconda di dove si hanno i flussi, si possono rappresentare vari comportamenti e, nel dettaglio di questo esempio, si potrebbe anche creare un ciclo:

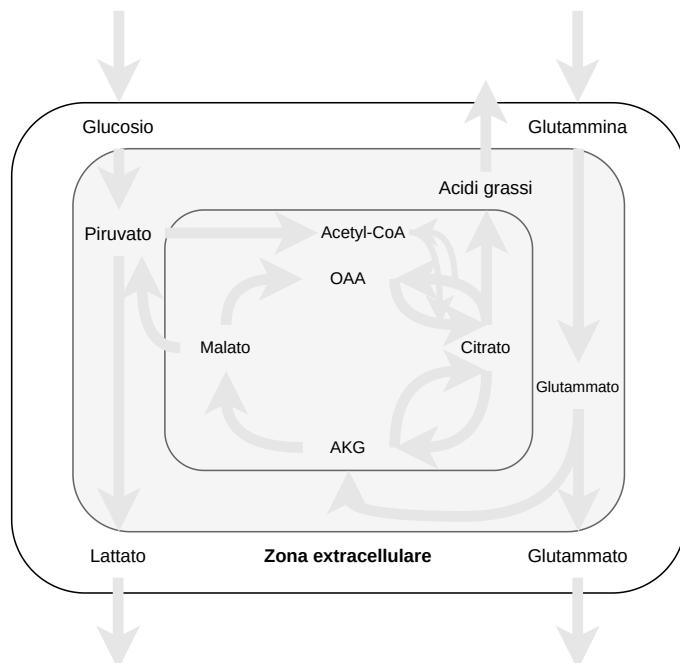


Ottenendo un Ciclo di flusso termodinamicamente irrealizzabile. Una tale situazione deve essere quindi risolta modificando o aggiungendo vincoli al sistema.

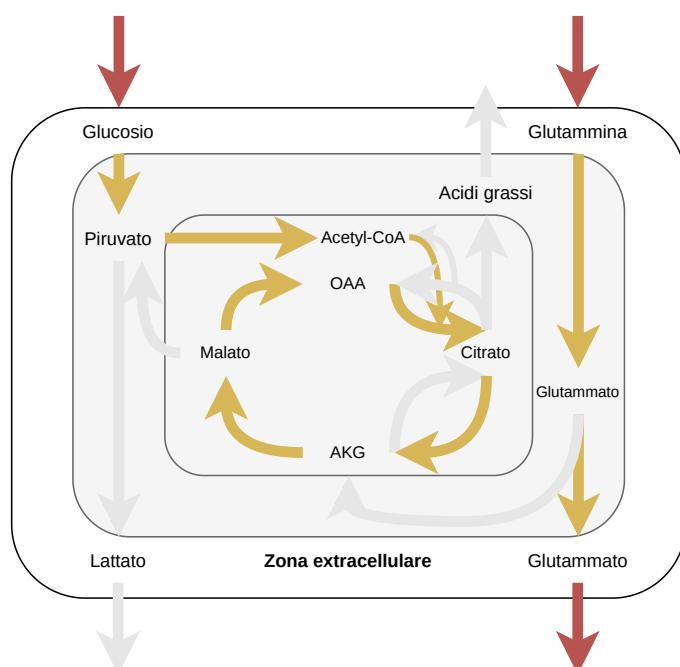
A tale sistema si aggiungono degli scambi necessari. Le reazioni di scambio sono definite tramite due quantità e questo, per la reazione R , si indica con $R \iff$:

1. **uptake (assorbimento)**: la rimozione dall'ambiente extracel-lulare, ovvero il *flusso negativo*. Il valore è compreso tra $-\infty$ e 0
2. **secretion (secrezione)**: l'inserimento nell'ambiente extracel-lulare, ovvero il *flusso positivo*. Il valore è compreso tra 0 e $+\infty$

Si ottiene quindi:



A questo punto possiamo avere molteplici steady state rappresentabili in questo modo. Ad esempio potremmo rappresentare uno (ma non l'unico) steady state per l'assorbimento forzato di Glucosio:



Su slide lezione altri possibili steady state, casi impossibili etc...

Parlando delle *funzioni obiettivo* se ne possono avere, ad esempio:

- per rappresentare obiettivi di design di bioingegneria
- per rappresentare obiettivi fisiologici plausibili
- per testare la capacità della rete di produrre un determinato metabolita
- per esplorare lo spazio delle soluzioni

L'obiettivo può essere sia ciò che vogliamo sia ciò che ci si aspetta ma anche per fare dei test per studiare certi casi (anche limite).

Per vedere qualche caso più specifico:

- minimizzare l'uptake di nutrienti
- minimizzare la produzione di *ROS* (*reactive oxygen species*), che sono specie chimiche chimicamente reattive contenenti ossigeno (sono sostanze tossiche)
- massimizzare la produzione di biomassa, aumentando la crescita
- massimizzare la produzione di un certo metabolita
- massimizzare la produzione di *ATP*

Formalmente la funzione obiettivo non è altro che una sommatoria pesata, con vettore dei pesi \vec{c} che dice quanto ci interessa massimizzare/minimizzare quel dato flusso (quindi una certa reazione), del valore dei flussi:

$$Z = \sum_i c_i v_i = \vec{c} \cdot \vec{v}$$

Quindi, ad esempio, se volessi massimizzare b_5 avrei una cosa del tipo:

$$Z = 0 \cdot v_1 + 0 \cdot v_2 + \cdots + 0 \cdot v_{n(i)} + 0 \cdot b_1 + \cdots + 1 \cdot b_5$$

avendo solo un flusso che contribuisce al valore della funzione obiettivo.

Su slide, pagina 55, esempio per aumento biomassa, che viene rappresentata come un finto metabolita, che non è stato approfondito a lezione.

Bisogna ora capire come trovare la soluzione. La matrice stechiometrica S

e i vincoli di capacità definiscono lo spazio di soluzione fattibile. Attraverso l'ottimizzazione di una funzione obiettivo, la *FBA* può identificare una singola distribuzione ottimale del flusso che si trova sul bordo dello spazio soluzione fattibile. In pratica si usa il **metodo del simplex** per ottenere questa singola distribuzione ottimale, che soddisfa la legge di azione di massa. Ovviamente potrei aver più modi per ottimizzare una data funzione obiettivo. In termini più “bio” si possono avere, ad esempio pensando al toy example di sopra, più modi di ottimizzare la produzione di *acidi grassi*. La *FBA* ne restituisce solo una. (*Per approfondimento sulla programmazione lineare e sul simplex guardare le note di Ricerca Operativa*).

Quello delle soluzioni alternative (avere più modi di ottimizzare Z) è un problema però parlando di metabolismo, infatti le soluzioni ottimali alternative possono differire da un punto di vista qualitativo (vengono sfruttate diverse vie metaboliche) e/o quantitativo (almeno un valore di flusso di una reazione varia tra diverse soluzioni). Si ha quindi la cosiddetta **Flux Variability Analysis (FVA)** che affronta questo problema calcolando la gamma di variazioni di flusso per ciascun flusso all'interno dell'insieme di soluzioni ottimali.

Con la *FBA* ottengo quale sia il massimo per un certo metabolita e poi usare tale valore come vincolo e vedere altre funzioni obiettivo. Ad esempio, avendo calcolato il massimo di acidi grassi, posso usarlo come vincolo per vedere il massimo lattato che è compatibile con la massima produzione di grassi. Questo si fa per ogni flusso, scoprendo i valori minimi/massimi di flusso compatibili con il massimo di un altro certo flusso. Questo è quindi quello che fa la *FVA*, che, senza approfondire i dettagli, si vincola un certo ottimo e si calcolano i massimi/minimi compatibili degli altri flussi. A questo punto ho un range per ogni flusso e posso studiare vari scenari:

- flusso massimo e minimo coincidono e quindi quel flusso deve avere esattamente quel valore per garantire l'ottimo di Z e quindi in tutte le soluzioni alternative quel flusso sarà sempre uguale
- flusso massimo e minimo divergono e quindi il ruolo della reazione varia tra soluzioni ottimali alternative, avendo che il flusso varia
- flusso minimo e massimo sono zero e quindi la reazione non può portare flusso in soluzioni ottimali, quindi quel flusso deve essere “disattivato”, altrimenti penalizzerei l'ottimo

Ovviamente esistono molte implementazioni per il *simplex* e anche questo ha portato a vari software per la *FBA*, tra cui:

- *COBRA Toolbox (Matlab)*
- *COBRAPy (Python)*
- *Optiflux*
- *FAME*
- *EsherFBA (user-friendly)*

Tali software prendono in input la rete, la funzione obiettivo e i vincoli.

7.1.3 Validazione della Rete

Bisogna capire se il modello è valido.

Si hanno:

- **valutazioni qualitative**, ad esempio se so che ad un lievito do certi nutrienti esso cresce allora anche il modello deve essere in grado di ottenere tale risultato con l'input di quei nutrienti, azzerando tutti gli altri input e massimizzando per la crescita. Esempio analogo può essere avere la secrezione attesa di un qualche composto e quindi aspettarsi che anche il modello, massimizzando tale secrezione, funzioni. Sono però valutazioni qualitative: mi aspetto che faccia lo stesso ma non controllo che più nel dettaglio “numerico” il modello si comporti come il corrispondente organismo reale
- **valutazioni quantitative** confrontando direttamente i dati ottenuti con esperimenti in *wet-lab* con quelli *in-silico*. Per farlo un degli standard è usare il **chemostato** in cui si ha un flusso costante in entrata (i nutrienti) e uno in uscita, permettendo di avere a livello biologico uno steady-state. Controllando i nutrienti in input e misurando le concentrazioni dei metaboliti in uscita, verificando quanti metaboliti vengono consumati per unità di tempo. Posso anche verificare la biomassa. A questo punto si pongono vincoli sui flussi di scambio, si trova il massimo della produzione di biomassa con la *FBA* e si confrontano i dati tra il modello e l'esperimento con il *chemostato*. Si controlla quindi se il modello sta predicendo correttamente

7.1.4 Tuning della Rete

Qualora non si sia soddisfatti del modello dopo la validazione si può fare **tuning della rete**, intervenendo sulla rete stessa oltre che sui vincoli.

Gli errori più comuni che vanno sistemati sono:

- si hanno *metaboliti dead-end*, che sono quelli che sono o solo prodotti o solo consumati dalla rete
- si hanno *gap nella rete*, ovvero mancano reazioni avendo proprio dei “buchi” (magari tra *metaboliti dead-end*), impedendo il “passaggio” di un flusso. Solitamente tali *gap* sono dovuti alla mancanza (o meglio “al mancato riconoscimento”) nel genoma dell’enzima che codifica il “collegamento” tra i due metaboliti. Scoprire un *gap* e risolverlo quindi migliora anche l’annotazione del genoma
- si hanno *cicli termodinamicamente infattibili* che accadono spesso in *reti genome-wide*, sempre a causa dell’annotazione genomica. Magari si ha una reazione che sposta un metabolita da un compartimento ad un altro, senza uso di cofattori, e un’altra che fa la stessa cosa usando invece dei cofattori. La seconda magari consuma *ATP* e produce *ADP* mentre la prima fa tutto senza consumare energia (???). Se si crea un ciclo tra queste due reazioni in pratica posso produrre *ATP* dal niente. Quindi una delle due deve essere imposta come irreversibile, per evitare di avere un ciclo termodinamicamente infattibile.
Si hanno nuovi metodi che consentono di rilevare loop termodinamicamente infattibili e di rimuovere soluzioni contenenti cicli di reazione chiusi

7.1.5 Esperimenti in Silico

Dopo la validazione del modello (quindi della rete) si possono vari esperimenti, sia per validare che per simulare/predire.

Uno degli esempi più classici è quello di simulare la *delezione di geni*. Ogni reazione è infatti associata ad un enzima che a sua volta lo è ad uno o più geni. A livello di modello se annullo un gene che era in *or* con un altro che “produceva” lo stesso enzima non succede nulla. Se invece si spengono entrambi (o solo uno che non è in *or* con altri) si ha l’eliminazione di una reazione del modello, ovvero il flusso associato sarà nel range nullo dal punto di vista dei vincoli ($0 \leq v_i \leq 0$). A quel punto si fa nuovamente l’esperimento ma con un flusso mancante. In pratica si sta facendo una *perturbazione*.

Nel nostro toy example, se blocco la reazione che va dal *glutammato* all’*AKG*, la cellula non è più in grado di fare *acidi grassi* (in quanto ho bisogno sia di *OAA* che di *Acetyl-CoA* per fare il *citrato*). A livello pratico faccio la simulazione e confronto i risultati della soluzione *trasformata* con la soluzione *wild-type* (e si può verificare anche in *wet-lab*).

Nel dettaglio, ad esempio, la *gene deletion analysis* si usa per fare debug della rete, studiare perturbazioni, studiare farmaci possibili etc...

7.2 Data-Driven Constraint

Le ricostruzioni metaboliche, come già detto, sono state fatte a mano per un individuo rappresentativo, diciamo un “uomo medio”. Si possono però, sulla base di dati (biopsie, linee cellulari, l’uso delle omiche etc...) specifici per un ceppo o un paziente, ottenere modelli specifici. Ovviamente si deve risolvere i problemi di *data integration* ma con tali modelli si studiano magari gli effetti di un certo farmaco su un paziente specifico. Se so che in un paziente, dalla trascrittomica, so che ho un gene non espresso, lo “spengo” anche nel modello. Discorso simile vale per le proteine/enzimi o dei nutrienti, coi dati della proteomica, per gestire certi vincoli. Il dato di metabolomica è quello più difficile da integrare, non sapendo quanto valgono le concentrazioni dei metaboliti.

Per risolvere i problemi di data integration si hanno varie tecniche. Ad esempio, se pensiamo alla formulazione logica dei geni vista in precedenza essa può essere “convertita” in un formato utile (ovvero una funzione continua) per la programmazione lineare, con l’*and* che diventa *min* e l’*or* che diventa la somma. Ad esempio:

$$(RRM1 \wedge (RRM2 \vee RRM2B)) \implies \min(RMM1, (RRM2 + RRM2B))$$

Con queste funzioni posso utilizzare le mappe metaboliche anche per mappare il dato in trascrittomica, quindi visualizzare su una rete quali sono le reazioni che, a livello di *gene expression*, sono più o meno attive (questo senza fare *FBA* etc...). Il team della professoressa Damiani ha sfruttato la cosa per confrontare la rete per il **cancro al seno** con quella *wild-type*, trovando vari pathway up-regolati, come quella della *glicolisi*, della produzione del *lattato*, della *sintesi dei nucleotidi* etc... Si ottengono informazioni metaboliche da un dato non prettamente metabolico sui flussi (e su altri dati metabolici) ottenendo alla fine vincoli per il nostro modello.

Inoltre non sempre si può conoscere con esattezza la funzione obiettivo, soprattutto pensando a due cellule, quindi si confronta l’intero spazio delle soluzioni, campionandolo, anziché solo l’ottimo, facendo infine un confronto statistico tra i flussi. In pratica non si sta risolvendo una funzione obiettivo ma si usa solo l’area definita dai vincoli. È interessante vedere come, aumentando i vincoli, si ottiene una sorta di **Principal Component Analysis (PCA)** infatti aumentando i vincoli (su nutrienti, omiche, flussi extracellu-

lari etc...) si possono, ad esempio, suddividere i vari sotto-tipi di un certo cancro.

Capitolo 8

Robustezza dei Sistemi Biologici

Dopo aver presentato alcuni dei metodi di modellazione più usati bisogna “mettere insieme” quanto visto, aggiungendo l’ultimo tassello che permette questa analisi. Tale tematica è la **robustezza dei sistemi biologici**, intesa come *robustezza funzionale*.

La robustezza è una proprietà che permette ad un sistema biologico (ma in generale qualsiasi sistema) di mantenere la sua funzionalità nonostante una o più perturbazioni esterne (come ad esempio la temperatura dell’ambiente) e/o interne (come ad esempio le mutazioni). I sistemi biologici, parlando sia di sistemi semplici come per i batteri che di quelli complessi come i mammiferi, devono essere *robusti* per sopravvivere a cambiamenti nell’ambiente e per mantenere una *stabilità fenotipica*, onde evitare conseguenze catastrofiche e la morte dell’organismo. La *robustezza* è un fenomeno fondamentale e onnipresente a livello di sistema che non può essere compreso guardandone i singoli componenti, come ad esempio le proteine. Si guarda quindi l’interazione dei singoli componenti studiando il comportamento complessivo risultante.

Bisogna notare che:

- la *robustezza* è spesso fraintesa per indicare il rimanere invariati da stimoli o mutazioni, in modo che la struttura/ componenti del sistema e la modalità di funzionamento non siano influenzati dagli stessi
- la *robustezza* è il mantenimento delle funzionalità specifiche del sistema contro le perturbazioni, e spesso richiede al sistema di cambiare la sua modalità di funzionamento in modo flessibile per sopravvivere. Questo fenomeno è il cosiddetto **rewiring**

- la *robustezza* consente di modificare la struttura e i componenti del sistema a causa di perturbazioni, mentre le funzioni specifiche sono preservate

Bisogna comunque osservare che non si ha *robustezza* contro ogni perturbazione possibile.

Il concetto di *robustezza* è fortemente legato a quello di *evoluzione*, al punto che non si può pensare all'uno senza pensare all'altro. La *robustezza* è una caratteristica fondamentale e una proprietà onnipresente di sistemi complessi che sono in grado di evolvere, infatti i sistemi biologici complessi devono essere robusti contro le perturbazioni ambientali/genetiche per poter evolvere e l'*evoluzione* (intesa in senso stretto come *evoluzione Darwiniana*) seleziona spesso tratti che potrebbero migliorare la *robustezza* dell'organismo.

Tuttavia, i sistemi robusti spesso affrontano fragilità verso perturbazioni inattese e cali delle prestazioni come *tradeoff*. In questo contesto l'identificazione dell'architettura di base per un sistema robusto e dei tradeoff associati è essenziale per comprendere i loro problemi e contromisure (ad esempio, malattie e terapie). Si studiano infatti sistemi che hanno perso il loro *stato fisiologico* dopo una perturbazione a cui era fragile, perturbazione che ha portato ad uno *stato malfunzionante/patologico*. Per questo si cercano o cure o anche modi per far morire ciò che non sta più funzionando.

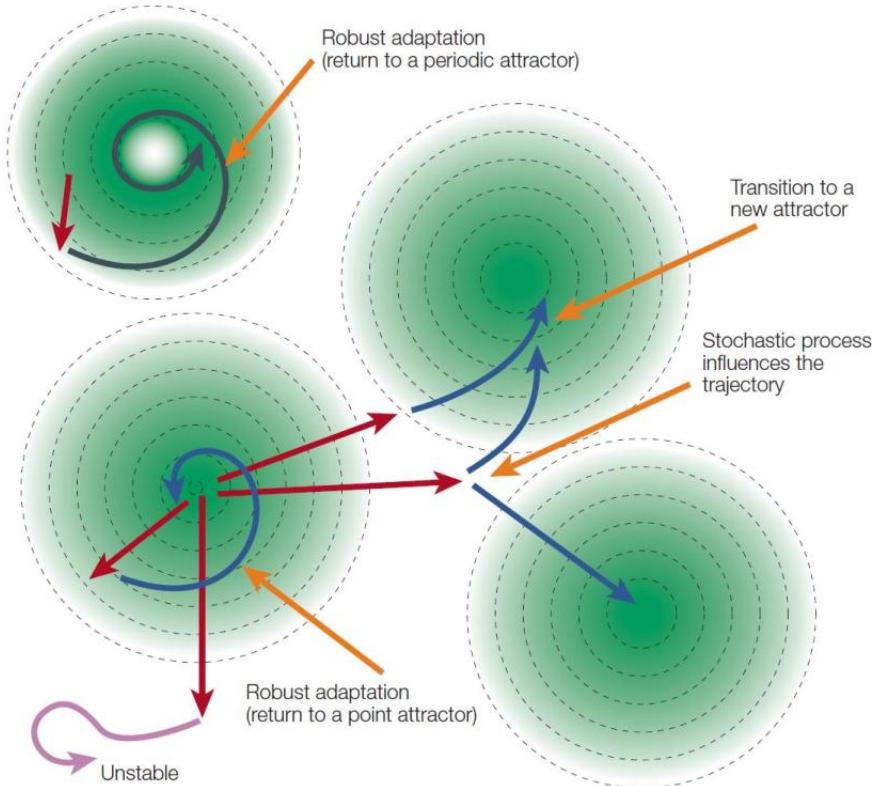
Malattie come il **cancro** o il **diabete** sono manifestazioni di robustezza *coop-tata*, avendo che i meccanismi che di solito proteggono il nostro corpo sono sfruttati per sostenere e promuovere lo *stato patologico*.

Dopo una perturbazione, la robustezza di un sistema può manifestarsi in due modi:

1. il sistema ritorna all'attractor/attrattore corrente, avendo il cosiddetto *robust adaptation/adattamento robusto*
2. il sistema si sposta su un nuovo attrattore, quindi in un nuovo stato, che mantiene inalterate le funzioni del sistema

Si noti che gli attrattori possono essere sia **statici (*point attractors*)** che **oscillatori (*periodic attractors*)**. Si ha quindi un'idea di attractor simile a quella vista coi modelli booleani.

Prendiamo un esempio tratto da un paper¹ di Kitano, uno dei padri della systems biology:



Dove:

- lo stato di un sistema può essere mostrato come un punto nello spazio di stato (qui semplificato in due dimensioni)
- le perturbazioni (le frecce rosse) spostano con forza il punto che rappresenta lo stato del sistema
- lo stato del sistema può tornare al suo attrattore originale adattandosi alle perturbazioni, tramite l'*adattamento robusto*
- il *feedback positivo* può muovere lo stato del sistema lontano dal attrattore corrente, o spingere il sistema verso un nuovo stato

Inoltre:

¹Kitano, Hiroaki. "Biological robustness." *Nature Reviews Genetics* 5.11 (2004): 826-837.

- in alto a sinistra il punto bianco in mezzo rappresenta lo stato fisiologico
- si ha, nell'esempio in cui si possono produrre due stati che mantengono le funzionalità del sistema, che il *rumore biologico* influenza la *robustezza*. Si ha che si arriva a due possibili stati stabili e questo altro non è che il già introdotto concetto di *bistabilità* (anche nell'accezione di avere una simulazione che passa tra uno steady-state e un altro in presenza di perturbazioni in quanto è questo il processo di adattamento). Pensando al discorso invece dei minimi locali e globali si ha che i minimi locali sono appunto altri stati possibili, non ottimi, che mantengono la stessa funzionalità di quello ottimo, che è il minimo locale parlando di ottimizzazione. Sena questi "stati di emergenza" si avrebbero conseguenze catastrofiche e la morte dell'organismo
- la perturbazione in basso a sinistra porta alla perdita dell'attractor e quindi all'instabilità, portando il sistema ad uno *stato patologico*

Bisogna ricordare inoltre come con la *parameter sweep analysis/parameter scan* si possa studiare come la condizione iniziale influenzi, in simulazioni stocastiche visto che quelle deterministiche danno sempre lo stesso risultato, il risultato del modello computazionale, permettendo di studiarne il potenziale bias.

8.1 Garantire la Robustezza

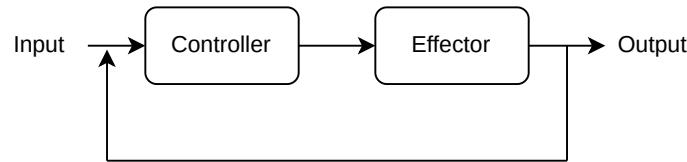
Esistono meccanismi specifici che possono **garantire la robustezza** di un sistema (biologico e non):

- **feedback regulation**
- **alternative/fail-safe mechanisms**
- **modularità**

8.1.1 Feedback Regulation

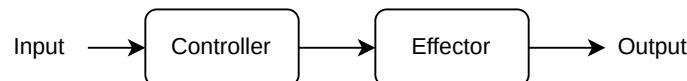
Il **feedback negativo** è la principale modalità di controllo che consente un robusto adattamento alle perturbazioni infatti l'autoregolazione negativa di un fattore di trascrizione può stabilizzare i suoi livelli di espressione allo

steady-state nonostante il rumore intrinseco nella trascrizione, il cosiddetto *rumore di smorzamento*. Si ha quindi un “appiattimento” del rumore a livello di *gene expression* evitando dei *burst* catastrofici. Si ha quindi un comportamento tipo:



Dove si noti che *controller* ed *effect* potrebbero essere una componente unica.

Si ha anche il **feedback positivo (feedforward)** contribuisce alla robustezza amplificando il stimolo e spesso generando bistabilità. Si ha che il livello di attivazione di un pathway downstream (a valle) può essere chiaramente distinto da uno *stato non-stimolato*, in modo che questi stati possano essere mantenuti. Si ha quindi ad esempio una cellula in steady-state che subisce uno stimolo. A questo punto la cellula sarà attivata passando un nuovo steady-state. Si ha quindi un comportamento tipo:



Dove si noti che *controller* ed *effect* potrebbero essere una componente unica.

8.1.2 Alternative/Fail-Safe Mechanism

La robustezza può essere migliorata se ci sono più mezzi per ottenere una funzione specifica, perché in caso di fallimento di una soluzione ci può essere la sostituzione con altre soluzioni.

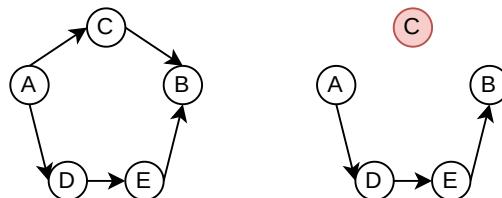
I **meccanismi fail-safe** si basano principalmente su due concetti:

1. la **ridondanza**, dove diversi componenti identici o simili (chiamati anche moduli) possono sostituirsi quando un altro componente fallisce. In realtà, parlando di sistemi biologici, non si hanno davvero componenti identiche ma solo simili. Ad esempio, in alcuni tessuti come quelli della pelle o ossei, le cellule sono circondate da vicini simili (appunto non identiche), in modo che le cellule danneggiate possano essere rapidamente sostituite da altre cellule

2. la **diversità/eterogeneità**, avendo che una funzionalità specifica può essere raggiunta con mezzi diversi, disponibili in una popolazione di componenti eterogenei, avendo cascate di reazioni diverse con lo stesso risultato. Un esempio è quello per cui la produzione di *ATP* può essere effettuata mediante *fosforilazione ossidativa* e *glicolisi*, avendo che la fosforilazione ossidativa richiede un apporto costante di ossigeno, mentre la glicolisi può essere aerobica o anaerobica. Un altro esempio è la *crescita diauxica* nel lievito, basata sul processo di passaggio dalla respirazione anaerobica a quella aerobica, che provoca un drastico cambiamento nelle vie metaboliche, a seconda che il glucosio o l'etanolo siano disponibili per il metabolismo energetico. Infatti entrambe i pathway alla fine producono materiali essenziali per la sopravvivenza e la crescita

Inoltre i meccanismi alternativi sono abbinati a una regolazione del feedback per garantire la robustezza, avendo che il passaggio tra meccanismi alternativi deve essere ben orchestrato per mantenere correttamente il comportamento del sistema. Si ha quindi una forte interconnessione dal punto di vista funzionale.

Se pensiamo alla **robustezza topologica** vista nel caso delle *scale-free network* si ha che c'è una forte connessione con la **robustezza funzionale** che si sta ora trattando. Si pensi a questi esempi:



Dove il nodo *C* sarebbe il percorso ottimale per andare da *A* a *B* ma anche se venisse attaccato si avrebbe comunque una via alternativa passando per *D* ed *E*, anche se magari non è ottimale. Infatti la **robustezza topologica** da informazione in merito al fatto che nonostante alcune perturbazioni che comportano la perdita di nodi (e archi annessi) si mantiene comunque la *connettività*, a meno che ad essere attaccato non sia un *nodo hub* avendo che in quel caso è difficile mantenere la connettività.

8.1.3 Modularità

La **modularità**, o **organizzazione gerarchica**, è un meccanismo efficace per contenere perturbazioni e danni localmente, minimizzando gli effetti

sull'intero sistema. Infatti i danni vengono limitati ad un singolo "livello", prevenendo al più possibile la diffusione degli stessi. Si hanno vari esempi di *modularità*:

- una cellula è un modulo che costituisce sistemi multi-cellulari e interagisce con l'ambiente e altre cellule
- la cellula stessa è composta da organelli, ed è anche una parte di moduli più grandi come un tessuto e un organo
- si hanno moduli funzionali, spaziali e temporali che possono essere riconosciuti come sottosistemi di reti metaboliche, processi di trasduzione del segnale e reti di regolazione dello sviluppo

Esempio dell'Aereo

Un esempio per capire il concetto di *modularità* per ottenere *robustezza* in sistemi che non siano solo biologici è quello del *pilota automatico degli aerei*. tale sistema mantiene la traiettoria di volo (direzione, altitudine e velocità di volo) contro le perturbazioni atmosferiche ed è generalmente composto di 3 moduli ridondanti con la stessa funzione ma ognuno con un "design" eterogeneo. Se fallisce uno dei moduli ci sono gli altri due che funzionano con un design architettonico diverso ma ottenendo il medesimo risultato. In ogni caso gli aerei sono estremamente fragili di fronte a perturbazioni insolite (mancanza di potenza totale etc...) perché il sistema di controllo di volo dipende totalmente dall'elettricità.

8.1.4 Tradeoff della Robustezza

Ovviamente si hanno vari **tradeoff** che sono conseguenze dei tre modi per garantire la robustezza.

L'introduzione di vari cicli di feedback di controllo genera trade-off causando:

- **instabilità** quando si incontrano perturbazioni impreviste, che portano a un fallimento catastrofico
- **riduzione delle prestazioni**, come nel caso della *chemiotassi batterica*, infatti i batteri potrebbero nuotare più velocemente senza feedback negativo, ma questo diminuirebbe la loro precisione nel seguire un gradiente chimico (andrebbero troppo veloci uscendo dalle zone migliori). Alla fine, in questo contesto, un comportamento adattivo risulta la soluzione migliore

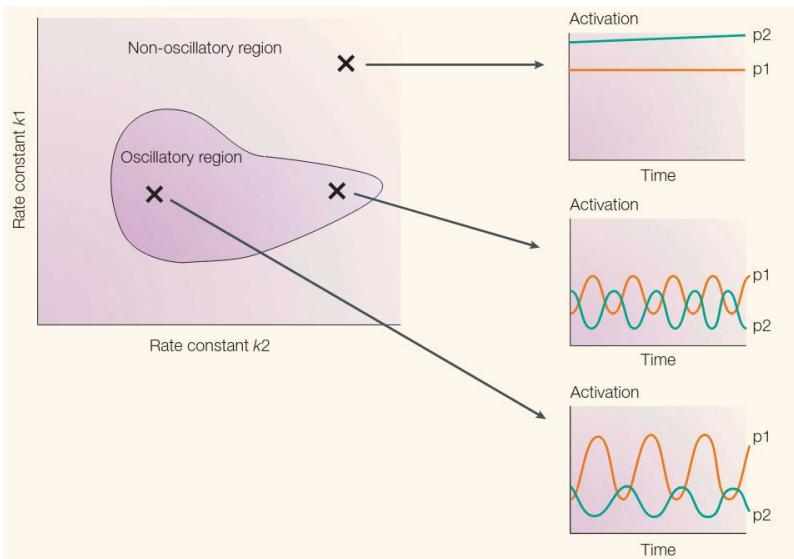
Inoltre i meccanismi alternativi e la modularità migliorano la robustezza, diminuendo la probabilità di guasto, ma al costo di maggiori richieste di risorse, sia in termini di energia che di quantità di cellule etc...

Infine i sistemi robusti sono più vulnerabili quando la fragilità del sistema è esposta. Si ha infatti il **processes hijacking (dirottamento dei processi)**, come lo sfruttamento di meccanismi fisiologici che mantengono le normali funzioni del corpo, che provoca l'insorgenza di molte malattie complesse multi-fattoriali, come il **cancro** che "modifica" il metabolismo per garantire la propria sopravvivenza.

8.2 Robustezza e Parametrizzazione

Molti metodi computazionali possono essere utilizzati per risolvere il problema della stima dei parametri, cioè per inferire la parametrizzazione del modello, in termini di vettori di parametri, che meglio si adatta ad alcuni dati sperimentali. Ogni parametro comunque, per quanto se ne possa calcolare un ottimo, può avere valore in un certo range per permettere al sistema di produrre un certo output.

Si prenda ad esempio questo grafico² dove si identificano due aree distinte a seconda (ma non solo) dei valori di due parametri cinetici, k_1 e k_2 , che sono ciascuno in un proprio range di valori, e tre possibili stati, rappresentati da delle X , con anche il loro comportamento in grafico:



²Kitano, Hiroaki. "Cancer as a robust system: implications for anticancer therapy." Nature Reviews Cancer 4.3 (2004): 227-235.

Si nota come:

- un cambio su k_1 , quindi in verticale cambiando regione, porti ad una variazione qualitativa, avendo il passaggio da un comportamento oscillatorio ad uno non oscillatorio e viceversa
- un cambio su k_2 , quindi in orizzontale ma restando in una delle due regioni, porti ad una variazione quantitativa, avendo solo che varia eventualmente l'oscillazione o la pendenza delle funzioni

In generale le due regioni qui rappresentano, nel caso non oscillatorio, l'interruzione del ciclo cellulare mentre nel caso oscillatorio il corretto funzionamento del ciclo cellulare stesso. Si ha quindi che lavorando sui parametri cinetici si possa interrompere il ciclo cellulare. A seconda della combinazione specifica di parametri, possono essere necessari diversi livelli di perturbazione per commutare il sistema in uno stato diverso e ovviamente, nel caso in esempio, si può lavorare anche su altri parametri per modificare la regione oscillatoria, in modo che le oscillazioni possano essere bloccate da una piccola perturbazione di uno dei due parametri cinetici visualizzati nel grafico, scegliendo anche quello su cui si può fare modifiche con meno fatica (nel grafico è meglio, se prendiamo lo stato più a destra, lavorare su k_1 che su k_2 richiedendo meno fatica per cambiare regione, avendo un range di valori più piccolo ???).

8.2.1 Teoria della Biforcazione

La **teoria della biforcazione** (*che viene solo introdotta in questo corso*) studia il cambiamento nel comportamento qualitativo dei sistemi dinamici, quindi, per esempio, il passaggio da un comportamento oscillatorio ad uno non oscillatorio.

Una biforcazione si verifica quando una piccola variazione di uno o più valori di parametro, i cosiddetti **parametri di biforcazione**, provoca un cambiamento qualitativo nel comportamento del sistema, cioè, la transizione dalle oscillazioni allo stato stazionario, o viceversa. Se prendiamo il grafico della pagina precedente il confine tra le due regioni segna appunto quando si avrà una biforcazione.

Si hanno vari esempi di *biforcazioni locali*, anche se alcuni sono migliori di altri in quanto certe soluzioni permettono cambiamenti al sistema che risultano in conseguenze catastrofiche che portano al collasso l'organismo:

- *Saddle-node bifurcation*
- *Transcritical bifurcation*

- *Pitchfork bifurcation*
- *Hopf bifurcation*

Si noti che tutte queste soluzioni si basano su simulazioni deterministiche e non si hanno al momento implementazioni per il caso stocastico.

I risultati per il *modello del pathway RAS/CAMP/PKA*, nei quali ad esempio sia aveva oscillazione per $150 < Cdc25 < 400$ erano un esempio pratico della *teoria della biforcazione*.

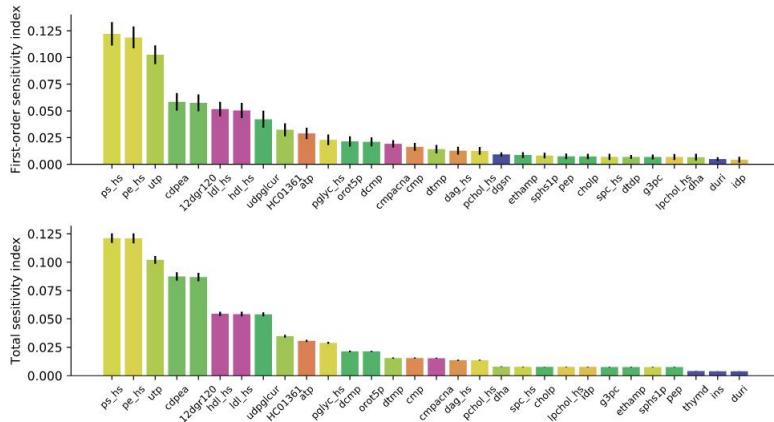
8.3 Sensitive Analysis

I modelli di sistemi biologici sono generalmente definiti in presenza di incertezze di vario tipo (struttura del modello, valori numerici di costanti cinetiche, etc...), avendo quindi lack di dati in input. Determinare la forza della relazione tra un dato input incerto e un dato output è l'obiettivo della cosiddetta **Sensitive Analysis (SA)** che quindi è un metodo computazionale volto a comprendere quale sia l'effetto di cambiare qualcosa nel modello (ad esempio, un parametro) sul comportamento di alcuni output del modello specificato. In pratica si cerca la relazione tra un certo input e un dato output.

SA consente di quantificare e classificare il grado di influenza di uno o più parametri di input (ad esempio, quantità iniziali, costanti cinetiche) su un determinato modello di uscita (ad esempio, la frequenza o l'ampiezza delle oscillazioni) calcolando i **sensitivity coefficients (coefficienti di sensitività)**. Formalmente si ha quindi, dato un parametro di input X_i e un dato parametro di output Y , il *sensitivity coefficient*:

$$S_{X_i,Y} = \left| \frac{\partial Y}{\partial X_i} \right|$$

Un esempio di tale classifica è visibile nel seguente grafico, ottenuto per un modello *constraint-based* per il modello *Recon 3D*³:



Ovviamente per ottenere tutti i valori per la *SA* occorre effettuare molteplici simulazioni e quindi è spesso richiesto un approccio parallelo di computazione.

Con la *SA* si risponde a varie domande:

- *quali sono le conseguenze della modifica di un parametro di input?* Si parla quindi di **model validation**
- *quali parametri richiedono ulteriori ricerche per ridurre l'incertezza di uscita?* Si parla quindi di **model calibration**
- *quali parametri non sono molto significativi e potrebbero essere rimossi dal modello, senza alterarne il comportamento?* Si parla quindi di **model reduction**
- *quanto dipendono le previsioni del modello dai valori dei parametri di input?* Si parla quindi di **model robustness**
- *quali parametri hanno la più alta correlazione con l'output del modello?* Si parla quindi **control analysis**

Si hanno due tipi principali di *SA*:

1. la **local SA**, dove si cambia il valore dei parametri in un range ristretto intorno al valore nominale (o all'ottimo se a disposizione). Viene usata per validare modelli dopo che sono stati raffinati e sono ritenuti “stabili”

³Nobile, Marco S., et al. "Accelerated global sensitivity analysis of genome-wide constraint-based metabolic models." BMC bioinformatics 22.2 (2021): 1-17.

2. la **global SA**, dove si cambia il valore dei parametri in un ampio range di valori, esplorando l'intero spazio dell'input in modo globale. Si sua in fase di definizione del modello e per testing dei modelli ottenuti nelle fasi iniziali, che sono quindi "instabili"

Del resto si hanno poi due strategie per lavorare con la *SA*, che comunque si basano sul calcolo di valori quantitativi:

1. il metodo **OAT (*One Factor at a Time*)** dove si studia un solo parametro per volta, per poi riportarlo al valore nominale e passare allo studio di un altro parametro. Ovviamente posso testare, in step differenti, più valori per un singolo parametro. Questa strategia è molto semplice, consolidata e a moderato costo computazionale ma impedisce di studiare le interazioni tra vari parametri, non esplorando infatti interamente lo spazio di input. Inoltre il coefficiente di sensitività calcolato può dipendere dallo scenario di base in cui si effettuano i test
2. il metodo di **combinazione di fattori**, in cui si sceglie un insieme di valori per ogni variabile per poi considerare tutte le possibili combinazioni creando ogni volta un modello parametrizzato con esse. Dal punto di vista teorico si può così esplorare l'intero spazio di input, considerando variazioni simultanee di parametri, identificando interazioni tra gli stessi e lavorando in modo indipendente dallo scenario sottostante. Del resto, dal punto di vista pratico, è irrealistico testare tutte le combinazioni, avendo una crescita esponenziale sul numero di parametri, e sono necessari metodi sofisticati per l'analisi di un sottoinsieme (basati ad esempio su *low-discrepancy sequences*). Inoltre tali metodi sono ancora poco sviluppati e hanno un altissimo costo computazionale

A livello cronologico quindi, in fase di costruzione del modello:

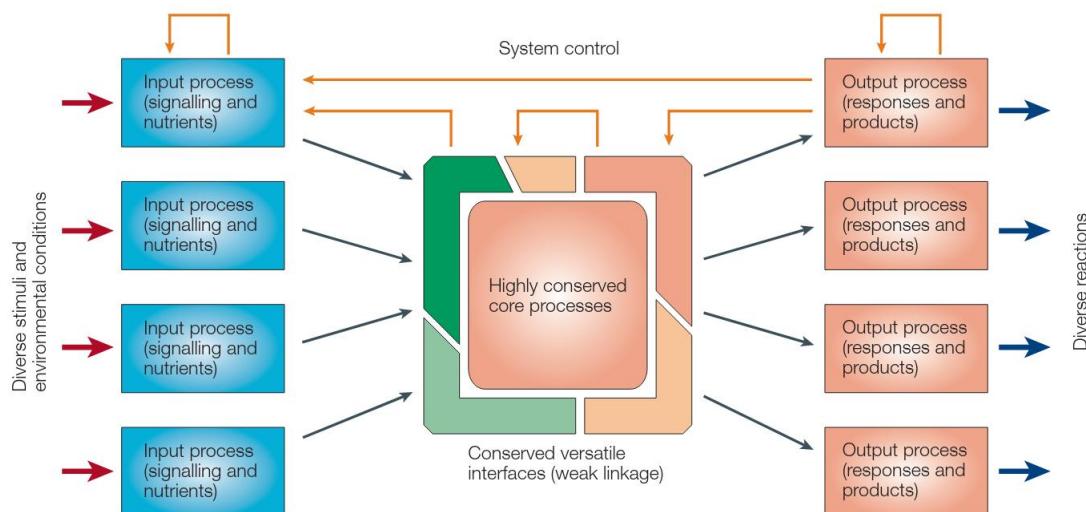
- prima si fa la *parameter estimation*, trovando l'ottimo per poter lavorare sul variare in un suo intorno
- poi si fa la *parameter sweep analysis*, che però non fornisce un output oggettivo come la *SA* ma mi definisce un range su cui basarmi e dei grafici da valutare a vista, e la *global SA*
- infine si fa la *local SA*

8.4 Bow-Tie Architecture

Per concludere il discorso sul legame *robustezza-evoluzione* con una citazione di Kitano:

What are the principles and mechanisms that lead to the emergence of robustness in biological systems? My theory is that robustness is an inherent property of evolving, complex dynamic systems—various mechanisms incurring robustness of organisms actually facilitate evolution, and evolution favours robust traits.

La capacità di evolvere infatti richiede flessibilità nella generazione di diversi fenotipi attraverso mutazioni non letali e l'architettura emergente è composta da processi core altamente conservati che sono co-selezionati con vari altri processi, alcuni dei quali producono tratti fenotipici nuovi. Questa architettura emergente è detta anche **bow-tie architecture** (dove *bow-tie* significa *papillon*) e consiste in un core altamente conservato, una serie di processi che fungono da interfaccia tra questo core e i processi di input e output, che sono a loro volta l'ultima componente di questa architettura. La forma a *papillon* è facilmente visualizzabile in questa figura⁴:



Questa architettura ha molti input e output, che sono collegati attraverso un core altamente conservato e un *legame debole ma versatile* con il controllo del sistema che ne governa le dinamiche. I processi principali e le interfacce versatili si sovrappongono o si fondono in alcuni casi. Il *legame debole*

⁴Kitano, Hiroaki. "Biological robustness." Nature Reviews Genetics 5.11 (2004): 826-837.

consente l'aggiunta di nuovi processi al processo core esistente, utilizzando meccanismi versatili comuni che operano su diversi input e output, come *canali ionici*, *proteine G* e *macchinari di trascrizione*. Infatti, se i macchinari di trascrizione fossero diversi per ogni gene, l'aggiunta di nuovi geni e nuovi regolamenti trascrizionali richiederebbero l'invenzione di un macchinario di trascrizione personalizzato che rende l'evoluzione quasi impossibile, richiedendo anche costi energetici assurdi.

Questo tipo di architettura si ha a vari livelli, coi rispettivi core altamente conservati (*questa slide, causa mancanza di basi biologiche non l'ho capita, mi sono limitato a tradurre e penso pure di averlo fatto molto male*):

- nella **trascrizione** e nella **traduzione**, dove i macchinari comuni sono usati per decodificare una vasta gamma di informazioni genetiche e per produrre le proteine varie, ma i meccanismi versatili, come i vari tipi di *polimerasi*, stessi compongono il core altamente conservato
- nella **trasduzione del segnale**, dove diversi stimoli sono ricevuti dai recettori e diverse isoforme di proteine G vengono attivate, ma convergono principalmente ai *secondi messaggeri* (*calcio, cAMP, IP3*, ovvero l'inositolo trifosfato etc...) che hanno una varietà limitata e comportano un legame debole (la modulazione in secondi messaggeri quindi influenza i processi core per innescare l'espressione di geni diversi)
- nel **metabolismo**, dove le diverse sostanze nutrienti sono trasformate in precursori che i pathway metabolici core convertono in “valute di energia per le cellule” (*ATP* e *NADH*) ed attivano i pathway biosintetici che producono *gli amminoacidi, i nucleotidi, lo zucchero, etc...*

Si noti che nessuno di questi processi è un semplice flusso di informazioni, poiché vengono imposte forti feedback regolatori locali e globali in ogni fase. Come visibile in figura 8.1⁵, è inoltre possibile avere un'architettura *bow-tie* che è il core di un'altra architettura *bow-tie*.

8.5 Cancro

Parlando di sistemi robusti non si può non citare, purtroppo, il **cancro**. Nel cancro infatti le cellule sono in grado di proliferare e metastatizzare

⁵Csete, Marie, and John Doyle. "Bow ties, metabolism and disease." TRENDS in Biotechnology 22.9 (2004): 446-450.

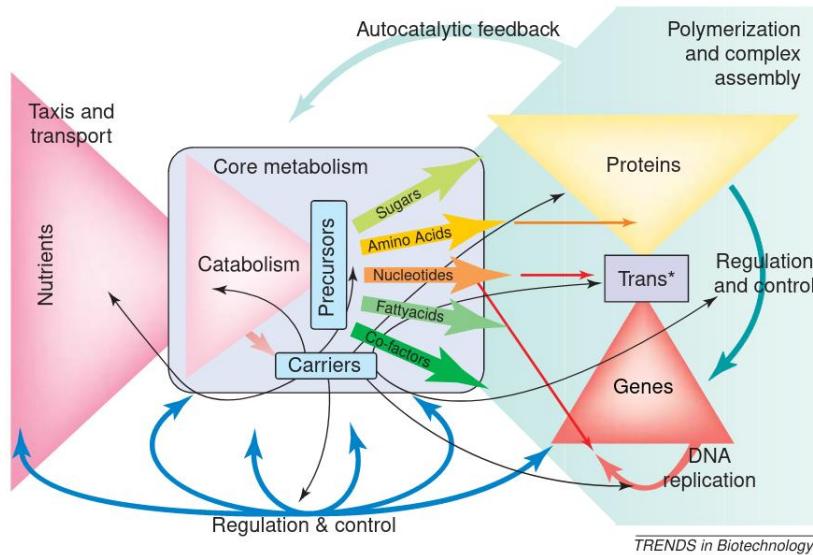
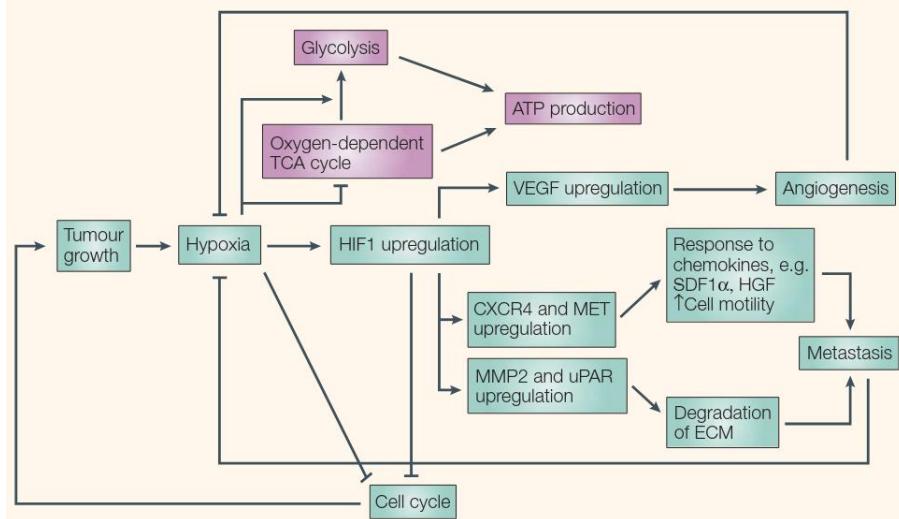


Figura 8.1: Esempio di *bow-tie architecture* innestata dentro un'altra *bow-tie architecture*.

nonostante gli sforzi terapeutici e ogni cellula tumorale è sia un componente autogovernato che parte di un sistema modulare/ridondante. La difficoltà di trattamento del cancro è attribuita alla sua robustezza acquisita, che è dovuta all'alto livello di eterogeneità genetica (causata da instabilità cromosomica) e alla distribuzione spaziale/temporale delle cellule tumorali eterogenee. Tutto ciò sostiene la sopravvivenza e la proliferazione del cancro stesso e consente alle cellule maligne di tollerare o adattarsi alla terapia e di ricostituire il tumore qualora eliminato in modo non totale.

Il cancro infatti sopravvive grazie a:

- molti cicli di feedback regolatori sia cellulare che nell'ambiente in cui si trova
- l'attività di *process hijacking* per migliorare la robustezza del cancro contro la terapia. Infatti:
 - la resistenza ai farmaci è causata dal upregolazione del gene *MDR1* e di altri geni, la cui funzione è quella di estrarre le sostanze chimiche tossiche dalla cellula (tra cui le cure chemioterapiche)
 - l'*ipossia* (basso apporto di ossigeno) durante lo sviluppo del cancro è contrastata da uno spostamento metabolico dal *ciclo TCA* (dipendente dall'ossigeno) alla

Figura 8.2: Schema dei cicli feedback nell'*apossia* tumorale.

glicolisi, nonché dal l'attivazione di un ciclo di feedback che promuove l'angiogenesi, che indirizza i vasi sanguigni in prossimità del tumore per ottenere nutrienti, e la metastasi delle cellule tumorali. In pratica all'inizio produce energia senza ossigeno e nel frattempo fa in modo che il metabolismo gli procuri i nutrienti necessari allo sviluppo. Solo a quel punto si inibisce l'*apossia*. Schematicamente il fenomeno è visibile in figura 8.2⁶

Si noti che la robustezza del cancro come sistema non è uguale alla robustezza delle singole cellule tumorali. Anche se ogni cellula tumorale è più fragile di una cellula normale in risposta a un particolare farmaco chemioterapico, la ridondanza eterogenea può dare luogo a robustezza a livello di sistema attraverso la variabilità genetica nel modello di resistenza ai farmaci. In pratica si fa chemioterapia colpendo le cellule tumorali con un preciso fenotipo e le altre, nel frattempo, creano resistenza farmacologica alla chemioterapia.

Si hanno vari paper sull'argomento:

- *Systems biology of cancer metastasis*⁷, dove si dice:

⁶Kitano, Hiroaki. "Cancer as a robust system: implications for anticancer therapy." *Nature Reviews Cancer* 4.3 (2004): 227-235.

⁷Suhail, Yasir, et al. "Systems biology of cancer metastasis." *Cell systems* 9.2 (2019): 109-127.

Systems biology approaches that result in predictive and testable models of complex phenotypes through integration of expertise from diverse fields including cancer biology, oncology, genetics, mathematics, bioinformatics, imaging, physics, and computer science could provide a more holistic understanding of the complete metastatic process.

- *Hallmarks of Cancer: The Next Generation*⁸ dove si studia come selezionare le cure per i cosiddetti *hellmark* in quanto tutte insieme ucciderebbero il paziente
- *Cancer: a systems biology disease*⁹ dove, come in altre pubblicazioni, si evidenzia la forte complessità del cancro come sistema complesso
- *Cancer: looking for simplicity and finding complexity*¹⁰ dove si studiano anche le scale temporali dell'evoluzione tumorale. Infatti si passa dal livello molecolare, con attività (come l'*apertura dei canali ionici*) in 10^{-6} secondi, a quello cellulare, con attività (come la *mitosi*) in 10^2 o 10^3 secondi, a quello fisiologico, della progressione tumorale, che avviene anche in 10^8 secondi (circa 3 anni)
- *3D multi-cell simulation of tumor growth and angiogenesis*¹¹ dove si studia l'evoluzione tumorale, nel caso di *tumori solidi*, anche dal punto di vista spaziale, con modelli 3D della stessa

Purtroppo non si sa ancora ne quali caratteristiche tumorali sono più significanti per uno studio computazionale ne tantomeno quali siano i migliori modelli da usare.

Poiché la completa eradicazione delle cellule tumorali è spesso irrealizzabile, possibili strategie cliniche contro il cancro consistono nel controllo della robustezza e nello scoprire le sue fragilità, quindi:

⁸Hanahan, Douglas, and Robert A. Weinberg. "Hallmarks of cancer: the next generation." *cell* 144.5 (2011): 646-674.

⁹Hornberg, Jorrit J., et al. "Cancer: a systems biology disease." *Biosystems* 83.2-3 (2006): 81-90.

¹⁰Grizzi, Fabio, and Maurizio Chiriva-Internati. "Cancer: looking for simplicity and finding complexity." *Cancer Cell International* 6.1 (2006): 1-7.

¹¹Shirinifard, Abbas, et al. "3D multi-cell simulation of tumor growth and angiogenesis." *PloS one* 4.10 (2009): e7190.

- la terapia potrebbe essere diretta a indurre la *dormienza*, ovvero lo stop della proliferazione, del cancro inducendo selettivamente l'arresto del ciclo cellulare, in quanto una fonte di robustezza è l'eterogeneità genetica creata attraverso la ricombinazione somatica
- l'instabilità genetica intrinseca e le mutazioni causate dai farmaci antitumorali contribuiscono a mantenere e aumentare l'eterogeneità delle cellule tumorali, che è essenziale per l'evoluzione del cancro
- la riduzione della massa tumorale potrebbe non essere un obiettivo terapeutico appropriato a causa dell'alto rischio di recidiva se il tumore ridotto guadagnasse un maggior livello di eterogeneità

Permangono dubbi su come i meccanismi che controllano la dormienza siano interrotti per consentire il riemergere e il deflusso metastatico anni o decenni più tardi. In ogni caso i cambiamenti cellulari-intrinsici, microambientali e/o sistemicamente relativi ai trattamenti fisiologici normali quali l'invecchiamento o la dormienza di effetto della dieta sono attivamente studiati. Inoltre la selezione di cure con target mirati su *hellmark* con combinazioni basate su studi meccanicistici si tradurrà in terapie più efficaci e durevoli per il cancro umano.

questa slide, causa mancanza di basi biologiche non l'ho capita, mi sono limitato a tradurre e penso pure di averlo fatto molto male.

Infine è bene citare un altro forte ambito di ricerca attivo. Infatti negli ultimi anni, la ricerca biologica e clinica si è concentrata sempre di più sul **tumor microenvironment (TME)**, che regola dinamicamente la progressione del cancro e modula la risposta alle terapie “standard-of-care” (ovvero la resistenza intrinseca e/o acquisita/adattiva). A seconda dello stadio di progressione del cancro e l'organo in cui il tumore si pone, le cellule TME possono essere sia *tumor-suppressive* che *tumor-supporting*. Data questa complessità, le strategie di TME-targeting, clinicamente approvate o attualmente in fase di valutazione negli studi, stanno emergendo come un approccio promettente per il trattamento del cancro tramite:

- *farmaci anti-angiogenici*
- eliminazione delle cellule e microambientali che promuovono il cancro
- “rieducazione” delle cellule microambientali che promuovono il cancro verso i fenotipi immuno-soppressivi/tumorali (ad esempio, con il controllo dei macrofagi associati al tumore, i *TAM*)

Per lo studio del cancro c'è quindi spazio sia dal punto di vista puramente biologico, che da quello della *systems biology* che da quello medico/clinico.