# Data and Computational Biology

UniShare

Davide Cozzi
@dlcgold

# Indice

# 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 Biologia Computazionale

**Materiale tratto dalla tesi**.

## 2.1 Accenni di biologia molecolare

### 2.1.1 DNA ed RNA

Prima di iniziare la trattazione più squisitamente computazionale è bene dare un'introduzione, dal punto di vista biologico, di quanto trattato.
Il **DNA**, sigla corrispondente ad **acido desossiribonucleico**, è un acido nucleico contenente le informazioni necessarie al corretto sviluppo di un essere vivente. Dal punto di vista chimico questa particolare macromolecola si presenta nella tipica **struttura a doppia elica**, formata da due lunghe catene di nucleotidi, dette **strand**. Nel dettaglio i singoli nucleotidi sono formati da un **gruppo fosfato**, dal **desossiribosio**, uno **zucchero pentoso,** e da una **base azotata**. Si hanno, inoltre, 4 tipi diversi di basi azotate:

1. **Adenina**, indicata con la lettera $A$
2. **Citosina**, indicata con la lettera $C$
3. **Guanina**, indicata con la lettera $G$
4. **Timina**, indicata con la lettera $T$

Si hanno quindi due **strand**, uno detto **forward strand** (indicato solitamente col simbolo "+") e uno detto **backward strand** (indicato solitamente col simbolo "−") che sono direzionati nel verso opposto (in termini tecnici si ha che il forward strand va da 5' UTR a 3' UTR, mentre il backward strand da 3' UTR a 5' UTR) e sono *appaiati* mediante coppie ben precise di basi azotate. Infatti, secondo il **modello di Watson-Crick**, si ha che:

- l'**Adenina** si appaia con la **Timina** e viceversa
- la **Citosina** si appaia con la **Guanina** e viceversa

Questo accoppiamento permette di poter studiare i due **strand** come uno "complementare" all'altro. Infatti, conoscendo la sequenza di basi azotate di uno **strand**, è possibile ricavare la sequenza dell'altro mediante la tecnica del **Reverse&Complement** dove, preso uno strand, si converte ogni sua base secondo il seguente schema:

- le $A$ diventano $T$
- le $T$ diventano $A$
- le $C$ diventano $G$
- le $G$ diventano $C$

**Esempio 1.** *Vediamo, per completezza, un esempio di* ***Reverse&Complement****.*
*Prendiamo una sequenza genomica* $S =$ *"TAGGCCATATGAC" e definiamo la funzione* $RC(x)$ *come la funzione che, presa in ingresso una stringa* $x$ *costruita sull'alfabeto* $\Sigma = \{A, C, G, T\}$ *(quindi una sequenza genomica), restituisce la* ***Reverse&Complement*** *della stessa. Si ha quindi che:*

$$RC(S) = \text{"}ATCCGGTATACTG\text{"}$$

Per riferirci al **DNA**, contenuto in una data cellula di un essere vivente, usiamo il termine **genoma**, che a sua volta viene organizzato in diversi **cromosomi**. Si definisce **gene** una particolare regione di un **cromosoma** in grado di codificare una proteina.
Ai fini della trattazione del progetto, è necessario introdurre anche l'**RNA**, sigla corrispondente ad **acido ribonucleico** (avendo il **ribosio** come zucchero pentoso), ovvero una molecola, simile al **DNA**, dotata di una singola catena nucleotidica, sempre con 4 tipi di basi azotate (anche se si ha l'**Uracile**, che si indica con la lettera $U$, al posto della **Timina**). Tra i compiti dell'**RNA** si ha quello della codifica e decodifica dei **geni**.

### 2.1.2 Esoni, Introni e Splicing alternativo

Per ottenere una **proteina** da un **gene** si hanno 3 passaggi:

1. La **trascrizione**, fase dove la sequenza del gene è copiata nel **pre-messenger RNA (pre-mRNA)**. Nel dettaglio viene selezionato uno dei due strand del gene e un enzima, chiamato **RNA Polimerasi**, procede alla trascrizione della sequenza selezionata creando il **pre-mRNA**. In questa fase la *Timina* viene sostituita dall'Uracile. È bene introdurre subito che in questo progetto non si terrà mai conto, a fini di semplificazione, del passaggio tra Timina e Uracile in quanto verrà usata sempre la *Timina*.

2. Lo **splicing**, fase dove vengono rimosse le parti non codificanti dalla molecola di **pre-mRNA**, formando il **messenger RNA (mRNA)**, detto anche **trascritto**. Per poter trattare al meglio questa fase bisogna parlare in primis di **esoni** e **introni**. In prima analisi si potrebbe dire, peccando di precisione, che gli **esoni** sono le sezioni codificanti di un gene mentre gli **introni** sono le porzioni non codificanti. Solo gli esoni formano il trascritto. Si ha, inoltre, che le prime due basi di un introne sono dette $5'$, nell'uomo solitamente si ha la coppia $GT$, mentre le ultime due, solitamente $AG$ nell'uomo, sono dette $3'$ e sono meglio identificate come **siti di taglio (splice sites)**. Quindi un esone, in realtà, non coincide esattamente con una regione codificante, detta **CDS**, a causa di queste particolari coppie di basi. Si notifica però che, come spesso accade, i termini vengono usati in sovrapposizione.

3. La **traduzione**, fase dove viene effettivamente codificata la proteina a partire da una sezione dell'**m-RNA**. Bisogna quindi nominare particolari sequenze nucleotidiche di cardinalità 3: i **codoni**. Tali triplette sono tradotte in amminoacidi che, concatenati, formano le proteine. Esistono particolari codoni che sono utili al fine di riconoscere l'inizio e la fine della *sintesi proteica*. In particolare si ha un codone d'inizio, detto **start codon**, che solitamente corrisponde alla tripletta $AUG$, mentre, per il codone di fine, detto **stop codon**, solitamente si ha una tripletta tra $UAA$, $UAG$ e $UGA$.

In realtà, un gene è in grado di sintetizzare più di una proteina mediante il cosiddetto **splicing alternativo**, che consiste in diverse varianti dell'evento

di splicing al fine di ottenere diversi trascritti. Si descrivono le principali modalità di splicing alternativo:

- L'**exon skipping**, ovvero *salto dell'esone*, dove un esone (o anche più esoni) può essere escluso dal trascritto primario oppure dove un nuovo esone (o più nuovi esoni) può essere incluso nello stesso.
- L'**alternative acceptor site**, ovvero *sito di taglio alternativo* $3'$, dove una parte del secondo esone può essere considerata non codificante o, alternativamente, una porzione dell'introne adiacente può essere considerata codificante.
- L'**alternative donor site**, ovvero *sito di taglio alternativo* $5'$, dove una parte del primo esone viene considerata non codificante o, alternativamente, una porzione di introne adiacente può essere considerata codificante.
- I **mutually exclusive exons**, ovvero *esoni mutuamente esclusivi*, dove solo uno di due esoni viene conservato nel trascritto.
- L'**intron retention**, ovvero *introne trattenuto*, dove un certo introne viene incluso nel trascritto primario.

Le varie modalità di splicing alternativo non si escludono a vicenda, rendendo lo studio di tale fenomeno assai complesso.

La **biologia** nasce come una disciplina altamente **descrittiva** mentre altre discipline, come, ad esempio, informatica, matematica o fisica, sono discipline **generaliste**. In biologia infatti si parte dai dati e dagli esperimenti per descrivere un fenomeno ed inferire la teoria su di esso. Questo è un discorso più di **filosofia della scienza**.
I biologi propongono **modelli**, come ad esempio i *pathway*, che sono il diretto risultato di osservazioni sperimentali e interpretazione dei risultati.

**Definizione 1.** *Un* ***pathway (*****percorso*****) biologico*** *è una serie di interazioni tra molecole in una cellula che porta a un determinato prodotto o un cambiamento in una cellula. Tale* percorso *può innescare l'assemblaggio di nuove molecole, come un grasso o una proteina. I* percorsi *possono anche attivare e disattivare i geni o stimolare una cellula a muoversi. I pathway più comuni sono coinvolte nel metabolismo, nella regolazione dell'espressione genica e nella trasmissione dei segnali e svolgono un ruolo chiave negli studi*

*avanzati di genomica.*
*Tra le principali categorie si hanno:*

- *Metabolic pathway*
- *Genetic pathway*
- *Signal transduction pathway*

Un altro aspetto chiave negli ultimi 25 anni è stato quello della mole di dati prodotti, tramite, ad esempio, **Next Generation Sequencing (*NGS*)**, con la produzione di *DNAseq* e *RNAseq* (che rispetto alle *DNAseq* sono più semplici da sequenziare e studiare e servono a vedere cosa sintetizza effettivamente una cellula), o alla cosiddetta **single-cell analysis**, una tecnica più recente, sviluppata negli ultimi 5 anni. I costi di sequenziamento variano a seconda del numero di basi da sequenziare ed è in calo negli anni. Tutte queste tecnologie *high-throughput* usate in biologia computazionale e in bioinformatica richiedono una forte conoscenza algoritmica, matematica e statistica per la gestione di questa enorme quantità di dati (essendo anche nell'ambito **big data**) in ambito biomedico. Saper modellare fenomeni biologici è essenziale anche per poter capire come eventualmente funzionano tecniche di machine learning dedicate, come le reti neurali. Ovviamente le conoscenze, i tempi (ma anche gli spazi), gli strumenti da usare e sviluppare etc... variano al variare del tipo di studio. Ad ogni problema è associato un miglior strumento di modellistica.
Un altro aspetto non trascurabile è la scala di misura di ciò che viene studiato, ad esempio:

- *organismi*, ad esempio per gli organismi multicellulari si passa da $10\mu m$ a $50/85m$
- *tessuti*, ad esempio per i tessuti umani siamo in un range maggiore di $10^4\mu m^3$
- *cellule*, ad esempio per quelle umane si va da $30\mu m^3$ a $10^6\mu m^3$ con:
  - membrane
  - nuclei
  - ribosomi
  - mitocondri e cloroplasti
  - altri organelli e strutture intracellulari

  - proteine
  - materiale genomico (DNA e RNA e affini strutture: ad esempio istoni)
  - ...

Parlando di tipi di organismi distinguiamo in primis:

- **eucarioti**. In questo caso si hanno cellule più complesse, con numerosi organelli e soprattutto il **nucleo**, dove sono contenute le informazioni. Si hanno i **mitocondri**, che si occupano di generare *energia* tramite *glicolisi* e sono studiati in ambito filogenetico, in quanto provengono unicamente dalla madre, permettendo la *filogenesi materna*
- **procarioti**, come i *batteri*. In questo caso si hanno cellule piccole e semplici. Non hanno un nucleo ma solo una regione, detta **nucleoide**, dove sono contenute le informazioni

Si hanno cellule nell'uomo, come quelle del sangue, dove non si ha un nucleo e non si ha riproduzione. D'altro canto si hanno anche cellule, come quelle dell'occhio, che non cambiano mai nel corso della vita.
In aggiunta si hanno anche i cosiddetti **archaea**.

**Tratto da Wikipedia**.
*Gli archèi o archèobatteri, nome scientifico Archaea (dal termine del greco antico che significa antico) o Archaeobacteria che significa "batteri antichi", sono una suddivisione sistematica della vita cellulare. Possono considerarsi regno o dominio a seconda degli schemi classificativi, ma mostrano strutture biochimiche tali da considerarsi un ramo basilare, presto distaccatosi dalle altre forme dei viventi. Nonostante il nome attribuito a questo taxon, gli archaea non sono i procarioti più antichi mai apparsi sulla Terra, ma sono stati preceduti dagli eubatteri. Essendo costituiti da singole cellule mancanti di nucleo, per forma e dimensioni molto simili ai batteri, sono stati in passato classificati come procarioti o monere assieme ad essi. Originariamente furono ritrovati negli ambienti più estremi, ma successivamente sono stati trovati in tutti gli habitat, compreso l'intestino umano, nel caso del Methanobrevibacter smithii.*
*Nonostante non sia del tutto sicura la filogenesi del gruppo, gli archei sono quindi (insieme agli eucarioti e agli eubatteri) uno dei tre fondamentali gruppi*

*degli esseri viventi nella classificazione di Woese. Tesi recenti propongono di considerare Archea ed Eukaryota un unico regno, contrapposto ai Bacteria, in quanto all'origine degli eucarioti vi sarebbe l'endosimbiosi mitocondriale.*

*Per ulteriori informazioni sui tipi di organismi guardare online.*
Parlando di DNA si ha che ogni cellula umana contiene circa 2 metri di DNA e un organismo umano contiene moltissime cellule rendendo lo studio del DNA davvero complesso (anche dal punto di vista computazionale si hanno file di genomi davvero molto pesanti, di centinaia di *MB*). Si hanno migliaia di trilioni di cellule nell'uomo.
Uno dei problemi è "allungare" il DNA che normalmente è incredibilmente avvolto su se stesso (e solo in fase di divisione cellulare si riconosce la forma a "X" dei cromosomi, altrimenti è ancora più avvolto su se stesso).
Dal DNA, nel nucleo, si ottiene l'RNA che esce, verso il citoplasma, dove, nei ribosomi, viene usato per sintetizzare le proteine.
Si hanno alcune specie interessanti dal punto di vista genomico e modellistico:

- **Saccharomyces cerevisiae**, ovvero il lievito da birra, con un piccolo genoma, *12 Mb*
- **Caenorhabditis elegans**, un "verme" di cui si è studiato l'intero sviluppo. Gli esemplari femmina hanno poco meno di mille cellule, 959, mentre i maschi poco di più, 1033. Si ha un genoma di *97 Mb*
- **Drosophila melanogaster** un altro modello molto usato, con un genoma di *180 Mb*
- **Homo sapiens**, con un genoma di *3200 Mb*
- **Mus musculus**, ovvero il topo, che ha un genoma molto simile a quello umano e quindi è molto usato negli studi in laboratorio. Si ha un genoma di *3300 Mb*
- **Arabidopsis thaliana**, ovvero la Veccia, che viene usata come modello base per studiare le piante. Si ha un genoma di *125 Mb*
- **Fritillaria assyriaca**, ovvero la Fritillaria, che ha il più lungo genoma conosciuto, di *120000 Mb*. Le piante normalmente hanno un genoma più lungo a causa dell'evoluzione, in quanto conservano molte informazioni che potrebbero essergli utili in futuro, anche in un futuro molto lontano, dovendo sopravvivere anche al fatto che non possono muoversi

Ad essere interessanti non sono solo le dimensioni di ciò che viene studiato ma anche i vari **tempi**. Vediamo una piccola tabella d'esempio:

| **Proprietà** | **E. coli** | **Uomo** |
|---|---|---|
| diffusione di proteine in una cellula | $0.1s$ | $\sim 100s$ |
| trascrizione di un gene | $\sim 1m$ $(80\frac{bp}{s})$ | $\sim 100s$ |
| generazione di una cellula | da $30m$ a ore | da $20h$ a statico |
| transizione di stato proteico | da $1\mu s$ a $100\mu s$ | da $1\mu s$ a $100\mu s$ |
| rate di mutazione | $\sim \frac{10^{-9}}{\frac{bp}{generazione}}$ | $\sim \frac{10^{-8}}{\frac{bp}{anno}}$ |

Qualche nota:

- i tempi di trascrizione di un gene umano includono i tempi di preprocessamento dell'$mRNA$
- per la generazione di una cellula di E. Coli si hanno 30 minuti in presenza di nutrienti
-

Si studiano quindi i vari **modelli** per la biologia computazionale che possono essere di varie tipologie:

- **continui**, tramite equazioni differenziali ordinarie
- **discreti**
- **stocastici**

Si studiano, in ottica analisi di cancro, anche **grafi mutazionali** e **evoluzioni clonali** (tramite Single-cell analysis).
Un aspetto fondamentale è costituito dall'RNA, che trasposta le informazioni dal DNA (contenuto nel nucleo) al citoplasma della cellula, dove funge da intermediario per il processo di sintesi delle proteine.

**Teorema 1** (Dogma principale di Francis Crick)**.** *Si ha quindi il dogma principale della biologia molecolare:*

***il flusso d'informazione è unidirezionale***

*ovvero, in termini più estesi:*

> ... once 'information' has passed into protein it cannot get out again. The transfer of information from nucleic acid to nucleic acid, or from nucleic acid to protein, may be possible, but transfer from protein to protein, or from protein to nucleic acid is impossible. Information means here the precise determination of sequence, either of bases in the nucleic acid or of amino acid residues in the protein.

L'unidirezionalità viene parzialmente infranta in caso di mutazioni del DNA ma questo non accade in fase di replicazione. Questa assunzione è una buona ipotesi dal punto di vista pragmatico.
Vediamo anche il pensiero di Sidney Brenner, biologo molto famoso: geni, proteine e cellule sono il *linguaggio macchina* della vita quindi per una corretta simulazione servono questi elementi, altrimenti il programma è una mera imitazione:

> *... his must not simply be another way of describing the behaviour. For example it is quite easy to write a computer program that will produce a good copy of worms wriggling on a computer screen. But the program, when we examine it, is found to be full of trigonometrical calculations and has nothing in it about neurons or muscles. The program is an imitation; it manipulates the image of a worm rather than the worm object itself. A proper simulation must be couched in the machine language of the object, in genes, proteins and cells.*
> *... The reader may complain that I have said nothing more than 'carry on with conventional biochemistry and physiology'. I have said precisely that, but I want the new information embedded into biochemistry and physiology in a theoretical framework, where the properties at one level can be produced by computation from the level below.*

Veniamo quindi alla distinzione delle due branche di studio. **Bioinformatica** e **Biologia (del Sistema) Computazionale** sono due aspetti sovrapposti del modo in cui usiamo l'approccio computazionale alla Biologia e alla Medicina, manipolando oggetti simili ma con enfasi diversa e diverse scale spazio-temporali. In entrambe si usano ontologie, formalismi descrittive ma anche, lato più pratico, database. Nel dettaglio:

- la **Bioinformatica** si occupa in primis dell'**analisi di sequenze** ovvero, tra le altre cose, di studio del genoma, RNA folding, folding di proteine e studio dei database necessari a questi studi. Si usano algoritmi di pattern matching e altri metodi di analisi delle stringhe
- la **Biologia (del Sistema) Computazionale** studia, tra le altre cose:

  - modelli e inferenze sulle proprietà dei sistemi, studiando simulazioni e nuove proprietà
  - ricostruzione di reti metaboliche e regolatorie e di modelli di progressione

  Si usano, ad esempio, metodi di machine learning per l'analisi dei dati prodotti e si simulano modelli biologici in modo sia deterministico che stocastico (tramite ad esempio Gillespie e Monte Carlo) e si fa analisi di raggiungibilità

D'altro canto, tecniche come la **Polymerase chain reaction (*PCR*)** ed altre sono appannaggio di biologi e biotecnologi. L'interesse per un biologo computazionale e per un bioinformatico è quello di aiutare altri ricercatori a svolgere le proprie attività. Ad esempio i biologi traggono vantaggio in ottica di acquisire conoscenze di base o anche al ricevere strumenti atti al progettare e pianificare esperimenti. Gli esperimenti biologici sono costosi sia dal punto di vista dei materiali che di persone e tempo.
In biologia computazionale si è quindi interessati a comprendere, anche in termini computazionali, l'interazione complessiva di:

- processi intracellulari (regolatori e metabolici)
- cellule singole
- popolazioni cellulari

Un altro compito dei biologi computazionali è quello di capire cosa succede quando si ha la possibilità di perturbare un sistema e vedere quali sono gli effetti della perturbazione, in particolare vedere cosa succede usando un dato farmaco piuttosto che un altro per intervenire su una certa patologia, parlando, in questo caso, del cosiddetto **momento traslazionale** della **medicina traslazionale**. Con "momento" ci si riferisce al trasferimento di conoscenze delle attività di pura ricerca alle **attività di produzione**, ovvero all'*attività clinica*, con il passaggio alla "vita vera". È interessante studiare il comportamento di una popolazione di cellule anche in presenza di una evoluzione tumorale.

# Capitolo 3

# Esempio del Repressilator

Introduciamo un esempio che rientra nell'ambito della *synthetic biology*, di M. B. Elowitz e S. Leibler[1], che sarà rivisto sotto diversi aspetti durante il corso. Questo è un esempio di un sistema biologico "ingegnerizzato", uno dei primi esempi di sistema biologico, di **biologia sintetica**.

## 3.1 Il Modello Biologico

In questo sistema si hanno tre geni, che per praticità chiamiamo *gene A*, *gene B* e *gene C*, ognuno dei quali, dopo essere trascritti e tradotti producono il rispettivo *mRNA* e poi, nel citoplasma, tali *mRNA* vengono usati per sintetizzare le tre rispettive *proteine*.
Quello che succede è che la trascrizione dei 3 geni può partire solo se non c'è proteina attaccata ad una sezione, detta *promotrice del processo di trascrizione*. Tale proteina è detta anche *promotore* o *inibitore*. Diciamo quindi che:

- per il *gene A* non deve esserci la *proteina C* attaccata per avere la trascrizione del gene stesso
- per il *gene B* non deve esserci la *proteina A* attaccata per avere la trascrizione del gene stesso
- per il *gene C* non deve esserci la *proteina B* attaccata per avere la trascrizione del gene stesso

È quindi un processo ciclico, che sarebbe discreto ma viene approssimato nel continuo. Nel dettaglio del Repressilator le proteine (prodotte dai rispettivi

[1]M. B. Elowitz, S. Leibler, A synthetic oscillatory network of transcriptional regulators, Nature 403(20), January 2000

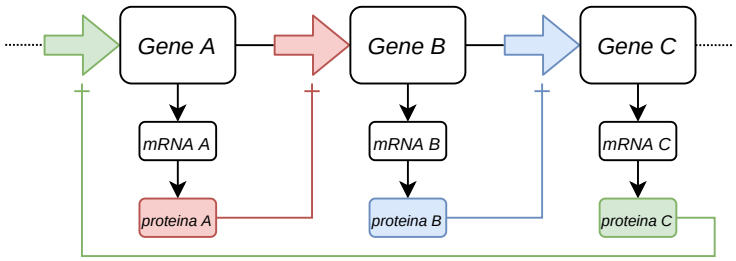



Figura 3.1: Schema di base del Repressilator, con le frecce bidimensionali che rappresentano l'azione di inibizione delle proteine.

geni che si indicano con la prima lettera minuscola) sono, in ordine (*A, B, C*):

- $TetR$ prodotta dal gene $tetR$
- $\Lambda cI$ prodotta dal gene $\lambda cI$
- $LacI$ prodotta dal gene $lacI$

Il punto fondamentale, come visibile in figura 3.1, è capire che se sto producendo una grande quantità di una certa proteina allora sicuramente non avrò produzione di quella di cui tale proteina inibisce la trascrizione del gene e così via. Nel nostro caso se si produce tanta *proteina A* non avremo produzione di *proteina B* e di conseguenza avremo produzione della *proteina C*, ma nel momento in cui questa terza viene prodotta cala la produzione della *proteina A* comportando la produzione della *proteina B* etc.... Ho, in pratica, un sistema oscillatorio, con 3 proteine che si reprimono l'una con l'altra.
La rappresentazione "su carta" di questo comportamento è abbastanza semplice, come vedremo, modellandola tramite un insieme di equazioni differenziali. Il problema è passare dalla teoria alla pratica. Questo sistema "ingegnerizzato", di equazioni differenziali, è in grado di confermare quanto visualizzabile poi tramite esperimenti.
Vediamo quindi come viene effettivamente costruito il sistema sperimentale usando delle colonie di E. Coli, sfruttando la loro biologia. Nei batteri il DNA non è, come detto, racchiuso nel nucleo ma "circola" in una regione, detta *nucleoide*, abbastanza accessibile all'interno del citoplasma. Nei batteri il DNA circola in forme dette **plasmidi** quindi potenzialmente si può sintetizzare un particolare plasmide e inserirlo in un batterio, il quale lo userà per sintetizzare proteine. Prima è stato comunque pensato il modello matematico e poi stato effettivamente costruito l'esperimento (al contrario dell'ordine con cui si stanno ora spiegando quindi).

I due ricercatori hanno costruito due plasmidi (di cui per ora non approfondiamo i dettagli):

- un plasmide che codifica il *Repressilator*, ovvero che contiene i 3 geni che codificano le 3 proteine. Prima di ogni gene si ha attaccata una *zona di induzione*
- un plasmide che codifica un *Reporter*, che codifica una particolare proteina, detta **green fluorescent protein (*Gfp*)**. La *Gfp* è una proteina usata spesso in quanto fa si che un certo sistema diventi fluorescente, di colore verde, una volta che viene illuminato con una certa luce (un laser ad una determinata frequenza). Questo plasmide fa si che, quando $TetR$ è presente in abbondanza la trascrizione del gene *gfp* viene bloccata e quindi diminuisce la quantità di *Gfp*. Quindi, come $TetR$ oscilla per il sistema di *mutua repressione*, si vedrà al microscopio un'oscillazione della fluorescenza della colonia di batteri.

Ricordiamo che la fluorescenza è in realtà abbastanza comune in natura.
Si ha un ulteriore "trucco". Se si lascia una colonia di E. Coli senza alcun controllo si avrebbe che ogni batterio inizierebbe il ciclo per conto suo, in modo non sincrono, impedendo una corretta visualizzazione della fluorescenza. Questo trucco è quello di inibire la produzione di $LacI$, interferendo con la sua espressione, usando un'ulteriore induttore, detto *IPTG* (*isopropyl* $\beta$-*D*-1-*thiogalactopyranoside*), e ottenendo così la sincronia delle cellule dopo questo impulso iniziale di *IPTG* (che poi decade velocemente lasciando tutti gli E. Coli nello stesso stato iniziale).

## 3.2 Il Modello Matematico

Facciamo quindi un passo indietro e vediamo il modello matematico del Repressilator. A partire dal modello matematico si scelgono le proteine da usare e il comportamento da ottenere.
Per prevedere il comportamento complessivo del sistema ingegnerizzato, si è quindi scritto un modello matematico che rappresenta la variazione dell'RNA e delle proteine espresse.
Per farlo indichiamo (**questo indice va sistemato**):

- $\alpha_0$, numero di copie di proteine per cellula prodotte da un certo promotore in presenza del repressore
- $\alpha$, numero di copie di proteine per cellula prodotte da un certo promotore in assenza del repressore (sarebbe $\alpha + alpha_0$)

- $\beta$, rapporto tra la velocità di decadimento dell'*mRNA* e quella della proteina
- $n$, *coefficiente di cooperatività di Hill* (nel caso del Repressilator si ha $n = 2$)
- $m_i$, i-esimo *mRNA*
- $p_i$, i-esima proteina che funge da repressore

L'intero sistema viene modellato con *coppie di equazioni differenziali*. Si hanno quindi:

- un'equazione che ci rappresenta la velocità di variazione dell'i-esimo mRNA:
$$\frac{\mathrm{d}m_i}{\mathrm{d}t} = -m_i + \frac{\alpha}{1 + p_j^n} + \alpha_0$$
Tale velocità dipende dalla quantità che già si ha di mRNA, dalla presenza della proteina che lo reprime (essendo sotto nella frazione al crescere il termine tende a zero, mentre al diminuire tende a 1)
- un'equazione che ci rappresenta la velocità di variazione dell'i-esima proteina che funge da repressore:
$$\frac{\mathrm{d}p_i}{\mathrm{d}t} = \beta(m_i - p_i)$$
Tale velocità dipende da quanto mRNA si ha a disposizione meno la quantità di proteina che si ha a disposizione in quel dato momento. Maggiore è la quantità di mRNA e maggiore è la produzione fino a che la proteina stessa non supera un certo livello di quantità, avendo che "satura"

In ordine si hanno, per i geni:

| Indice | 1 | 2 | 3 |
|---|---|---|---|
| $i$ | $lacI$ | $tetR$ | $\lambda cI$ |
| $j$ | $\lambda cI$ | $lacI$ | $tetR$ |

Con "velocità di variazione" si intende in pratica un tasso di cambio di concentrazione delle due *specie molecolari*, ovvero un'entità che osserviamo nel modello (in questo caso mRNA o proteina).
Le concentrazioni si esprimono con l'unità di misura $K_M$, ovvero il numero di

repressori necessari per dimezzare la repressione di un promotore, e il tempo in $\tau_{mRNA}$, ovvero la velocità di trascrizione dell'mRNA, detto **mRNA lifetime**. Integrando numericamente le due equazioni differenziali otteniamo un comportamento periodico.
L'esperimento è stato fatto poi osservando come tutto questo diventa osservabile in una colonia di E. Coli, opportunamente trattata, usando delle foto (dove si è osservato anche un drift verso l'alto nel grafico oscillatorio a causa del fatto che la colonia si espande).
La conoscenza di tipo matematico deve però essere trasferita in un esperimento reale che funzioni (e i ricercatori devono essere in grado di manipolare entrambi gli aspetti, si quello della modellazione matematica che quello più biologico e chimico). In questo caso per ottenere oscillazioni stabili servono determinati prerequisiti:

- usare inibitori artificiali piccoli, con la cosiddetta *low leakiness*. Promotori più corti sono più facili da manipolare e sono più "veloci"
- la velocità di decadimento di proteine e mRNA doveva essere simile, per ottenere l'oscillazione, una meglio: una buona oscillazione. Questo si ottiene attaccando *ssrA* ad ogni repressore
- servono curve di repressione piuttosto "ripide". Per questo si è usato un promotore con multipli *binding sites* (arrivando alla scelta di quelle date proteine), usando repressori cooperativi (questo è rappresentato con il parametro $n$)
- usare un *Reporter* non stabile, attaccando una variante di *ssrA* a *Gfp*, altrimenti si avrebbe una fluorescenza costante

**Listing 1** Semplice implementazione del sistema in Python dove l'unico parametro che varia è $n$ mentre gli anni sono stati precedentemente fissati

```
def repr(var, time, n):
  mRNA = var[:3]
  prot = var[3:]
  dmRNA0 = - mRNA[0] + alpha/(1 + prot[2]**n) + alpha0
  dmRNA1 = - mRNA[1] + alpha/(1 + prot[0]**n) + alpha0
  dmRNA2 = - mRNA[2] + alpha/(1 + prot[1]**n) + alpha0
  dprot0 = - beta*(prot[0] - mRNA[0])
  dprot1 = - beta*(prot[1] - mRNA[1])
  dprot2 = - beta*(prot[2] - mRNA[2])
  return [dmRNA0, dmRNA1, dmRNA2, dprot0, dprot1, dprot2]
```

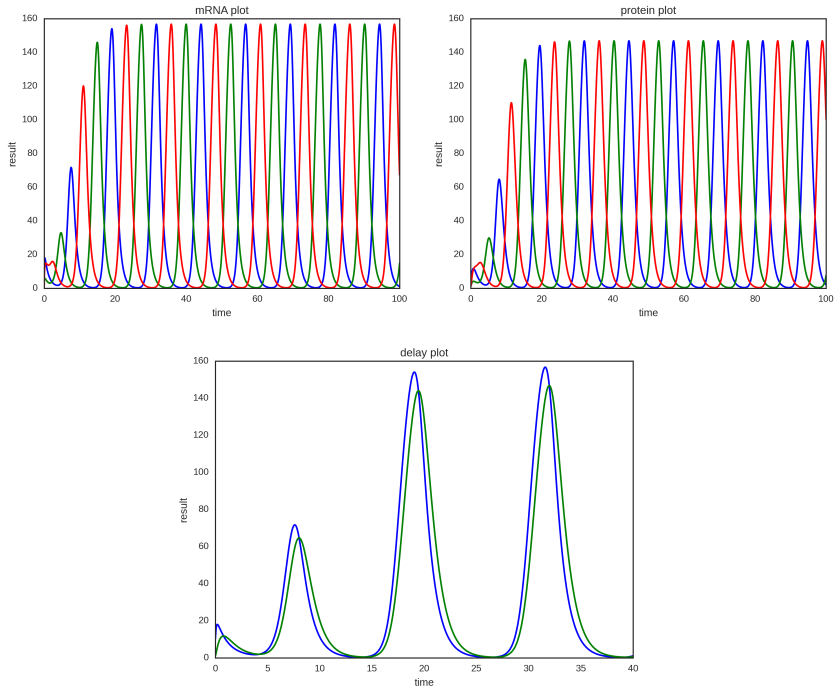



Figura 3.2: Grafici relativi al modello del Repressilator ottenuti tramite Python e Matplotlib, con $n = 2$, $\alpha_0 = 0.005$, $\alpha = 220$ e $\beta = 2$. In primis, a destra quella di mRNA mentre a sinistra la quantità di repressore/proteina rispetto al tempo. In basso le quantità di mRNA (nel caso di $tetR$) rispetto al repressore/proteina (in questo caso ovviamente $TetR$) associata rispetto al tempo. Si nota un piccolo delay, che rappresenta il tempo di traduzione.

# Capitolo 4

# Studio di Sistemi Biologici

Cerchiamo ora di capire come classificare i problemi, come analizzarli e comprenderli (anche tramite machine learning) e avere coscienza delle risorse online disponibili per la tematica.
Buona parte della ricerca in biologia computazionale ha come obiettivo quello di ottenere il passaggio dai risultati di laboratorio alle applicazioni cliniche (ed è qualcosa di molto complesso). Per quanto ci sia interesse verso tutte le patologie la più interessante e più studiata (soprattutto in questo corso) è il **cancro** (ma si avrà anche un approfondimento di situazioni pandemiche come quella del **Covid**). Un esempio di un sistema particolare dove i tumori si sviluppano è quello delle cosiddette **cripte coloniche (*colonic crypts*)**, avendo che questo sistema è relativamente semplice da studiare dal punto di vista computazionale.
Le *cripte coloniche* si trovano nell'intestino e sono delle sorta di "pozzetti", morfologicamente divisibili in varie aree. Alla base delle cripte ci sono delle **cellule staminali epiteliali**, che sono quelle che poi danno luogo ai tessuti dell'epitelio. Nella parte più esterna, ovvero nella superficie intestinale, si hanno le strutture per l'assunzione dei nutrienti.
*Dal punto di vista matematico tutti gli essere viventi sono di topologia isomorfa a dei tubi.*
Tornando al discorso delle cellule staminali si ha che essere si suddividono e, man mano che si suddividono tendono a spingere verso l'alto le cellule che si trovano "al di sopra" di loro, spingendosi verso la superficie dell'intestino. Man mano che tali cellule vengono spinte anch'esse tendono a dividersi spingendo le altre cellule verso il *lumen della cripta*. In questo processo di suddivisione queste cellule si differenziano e le cellule staminali danno luogo ad una progenie che possiamo, dal punto di vista in primis computazionale, rappresentare come un *albero*. Si hanno le cellule di tipo diverso, più o meno differenziate che continuano a salire verso la superficie dell'epitelio e

poi tendono a salire su quelli che sono detti i *villi intestinali*. Nel salire si possono produrre situazioni di sovra-riproduzione, provocando la produzione dei cosiddetti **polipi**. Questo è un interessante processo che può essere simulato, tra i vari modi, in modo tale che si simuli cosa accade quando le varie differenziazioni non funzionano perché, ad esempio, si ha una cellula che ha acquisito una mutazione, mutazioni che danno luogo ad una crescita non corretta, ad una *displasia*, che è la fase iniziale da cui poi si sviluppano i *tumori del colon*. Si vuole quindi fare queste simulazioni e farle in modo il più fedele possibile. Il modello delle cripte coloniche è comodo in quanto richiede poche cellule per la sua simulazione.
Per capire se una cellula si sta comportando in modo corretto o meno dobbiamo misurarne il comportamento. In primis vogliamo misurare due cose, tra le tante:

1. **gene expression**
2. **gene alterations**, ovvero le varie mutazioni del genome, le cosiddette le *copy number variations* etc...

La tecnologia a disposizione per queste tematiche si è molto evoluta ma tra le tante tecnologie si segnalano:

- *microarrays* per l'espressione genica, usati però molti anni fa essendo una delle prime tecnologie per misurare, in modo indiretto ma parallelo, l'espressione dei geni
- *Next Generation Sequencing (NGS)* per praticamente qualsiasi cosa, anche per l'espressione genica, in modo diretto tramite particolari esperimenti (**nella rec non ho capito il nome di tali esperimenti**). NGS ha avuto molta fama da circa il 2006 in poi, con il monopolio poi di Illumina, anche se di recente si hanno nuove tecnologie che stanno rivoluzionando il settore (che producono read più lunghe)

## 4.1 Microarrays

Parliamo un secondo dei **microarrays**.
Questa è una tecnologia non più utilizzata, essendo di inizio anni duemila, che però è utile per spiegare come si procede a fare un certo tipo di misure, con una tecnologia che è stata poi ripresa da Illumina.
Questo strumento si basa su una griglia a cui sono attaccate delle "sonde"

lunghe circa 25 nucleotidi e venivano usati per caratterizzare i geni. Si producono infatti segnali luminosi di diversa intensità e diversa lunghezza d'onda in una griglia, da cui si può ricavare una griglia numerica che dà informazioni in merito alla luce di ogni punto.
I Microarrays sono prodotti da Affymetrix e hanno circa $10^5$ sonde, che caratterizzano tutti i geni che interessano e l'attacco alle sonde avviene tramite basi complementari.
Si ottiene quindi un'immagine contiene una griglia, dove in ogni punto si produce un segnale luminoso di diversa intensità e lunghezza d'onda dalla quale si ricava, misurando i segnali luminosi, una **matrice di espressione**, dove:

- le righe sono i geni/trascritti
- le colonne sono misure numeriche

e si ha, per ogni sonda, quanto e come è luminoso il tal punto nella griglia.
Si prende quindi del DNA, lo si "denaturalizza", ovvero lo si sgroviglia, e lo si versa direttamente sulla griglia. Il DNA (ma potrebbe essere anche essere RNA) viene versato sulla griglia e si "attacca", grazie alle sue proprietà chimiche, alle sonde (in pratica le parti di DNA/RNA si attaccano alle sonde a loro complementari). Il trucco è quello di "colorare" i pezzi di DNA e RNA e questo si fa usando, come nel caso del Repressilator, delle proteine fluorescenti, verdi e rosse, usando quindi processi biochimici per attaccare ai pezzi di DNA/RNA queste proteine, che emetteranno fluorescenza una volta colpite da un laser. Si può quindi vedere, in ogni punto della griglia, se si ha un segnale rosso o uno verde, misurandone l'intensità, ottenendo una misura di quanto materiale genico si sia attaccato in ogni punto della griglia.
Vediamo quindi come si utilizza questo tipo di tecnologia per fare delle misure di *geni differenzialmente espressi in diverse condizioni*.
Si hanno delle cellule in una certa condizione e altre in un'altra condizione (magari, per esempio, una delle due condizione è una crescita in ambiente con pochi nutrienti o in un ambiente con temperature estreme, sia alte che basse con associati shock termici per le cellule). La prima condizione è normalmente uan *condizione standard*, detta *condizione wild-type*, mentre la seconda è la condizione che si vuole studiare.
Si hanno due fasi per l'esperimento (anche se tendenzialmente non sono esperimenti molto semplici):

1. si estrae dalle cellule nelle due condizioni l'RNA, che descrive ciò che le cellule stanno in quel momento esprimendo, quali proteine stanno sintetizzando, etc... Dall'RNA, che nel dettaglio è *mRNA*,

estraggo il *cDNA*, al quale poi attacco le proteine per la fluorescenza. Si procede quindi con la cosiddetta *ibridazione*, ovvero si prende il materiale genetico con fluorescenza e si immerge il microarrays in questa soluzione, procedendo poi alla scansione con il laser

2. nella griglia si ottiene quindi che del materiale genico delle cellule nella prima condizione si attaccano ad alcune sonde mentre quella delle seconda condizione ad altre. In ogni punto della griglia o non si attacca niente (non avendo che le cellule esprimono quanto necessario per quel punto) o si attacca solo l'RNA di una delle due condizioni o si attaccano entrambi. Usando poi i laser per le due fluorescenze si ottiene l'immagine, avendo punti senza luce (nero), alcuni con luce verde, alcuni con luce rossa, a seconda della prevalenza del materiale che viene da una delle due condizioni (se simili si ha una luce tendente al giallo). Una volta prodotta l'immagine si produce l'output numerico delle intensità.

L'esperimento può essere ripetuto più volte, ottenendo una serie di matrici numeriche che possono unite in vari modi, ottenendo la **gene expression data matrix** finale, coi vari **gene expression levels**, i livelli di espressione di ogni gene, ricordando che ogni gene è codificato da più sonde. Per ogni gene ho la **differenza di espressione** tra le due condizioni.

**Definizione 2.** *Si definiscono due geni come* ***differenzialmente espressi*** *se sono due geni che risultano rossi o verdi (???).*

Se tale matrice finale è ottenuta variando solo i tempi e mantenendo fisse le altre condizioni sperimentali si ha che essa rappresenta il *time-course of genes expression*.
Sui risultati si può fare **data mining**, usando tecniche di machine learning. Si vuole fare clustering di geni o sonde che esibiscono un comportamento simile dato un insieme di condizioni sperimentali o ambientali. Per farlo si hanno vari tool (molti dati disponibili sulla repository NCBI, soprattutto nella sotto-repository GEO) ma molti studi richiedono una sistemazione finale non banale in merito a "rumori" e variazioni di protocollo nei laboratori.
Ad esempio, in un esperimento di espressione genica si hanno vari step:

1. dopo la "pulizia" della matrice (tramite controllo qualità) si usano alcune analisi standard, ragionando magari su vari *time points* discreti:

    - *clustering*, tramite K-Means, per ogni punto, ottenendo dei vettori che rappresentano il comportamento di

un gene in un certo tempo. Si ottengono cluster di traiettorie. Si raggruppano geni con simile profilo di espressione

- *enrichment*, che altro non è l'operazione in cui si prendono i dati e gli si associano informazioni, tramite *Gene Ontology (GO) Terms*. La GO è un elenco di nomi con ID unico e oggi come oggi i geni noti sono stati già etichettati coi termini dalla GO. Vengono annotati i termini sovrarappresentati in un cluster. L'etichettatura fa si che quando ho gruppi di geni posso usare tecniche statistiche, come il **test esatto di Fisher**, per estrarre i termini più rappresentativi, quelli più presenti e descrittivi di un gruppo. In questo modo, un cluster può essere associato ad alcuni termini "rappresentativi", che possono indicare una certa caratterizzazione funzionale e ipotesi di associazioni tra geni e un certo comportamento (se questo non fosse già annotato). Questa tecnica è detta **associazione a delinquere**, in quanto si "accusano" geni di essere associati ad altri, comportandosi in modo simile

**Su slide, parte 2 a pagina 13, grafici di un esperimento e annessi termini da GO**.
Vediamo nel dettaglio GO[1] che è appunto un *vocabolario controllato/ontologia* che è diventato la chiave per condividere le conoscenze biomolecolari, in particolare per i geni e i prodotti genici. Questa ontologia è nata studiando la *fruit fly*. È nata negli anni novanta a Berkeley mettendo insieme una serie eterogenea di conoscenze proveniente da vari ambienti. È nata cercando una nomenclatura standard per la genetica della Drosophila. È stata ottenuta con lo sforzo di informatici, biologi, filosofi etc... usando, in primis, l'IA simbolica (usata per le ontologie, ovvero modi di descrivere in modo simbolico una serie di concetti).
Ogni temrine in GO ha un codice numerico univoco.
Si hanno tre sotto-ontologie, ognuna con una struttura gerarchica a DAG (**su slide immagine di struttura**):

1. **MF** (*Molecular Function*), per le attività biochimiche il tipo molecolare
2. **BP** (*Biological Process*)

[1] `www.geneontology.org`

3. **CC** (*Cellular Component*)

Lato tecnico si ha, sotto GO, un linguaggio logico (stile *Prolog*), con un insieme di relazioni, termini e costanti di un linguaggio.
GO non è l'unica ontologia a disposizione, anzi se ne hanno centinaia ma meno importanti. GO offre delle API e si hanno tool come *AmiGO* o *PANTHER* per recuperare informazioni.

## 4.2 Next Generation Sequencing

Dopo aver parlato di *microarrays* parliamo di **Next Generation Sequencing (*NGS*)**.
Vediamo quindi le nuove tecnologie di sequenziamento. Diciamo "nuove" perché le prime tecnologie di sequenziamento sono datate anni cinquanta con il metodo Sanger per sequenziare proteine. Più avanti, nei primi anni settanta, si sono sviluppati i primi progetti per sequenziare DNA e RNA, studiando i virus (in quanto molto piccoli). Nel 1995 poi si è riuscito a sequenziare interamente un batterio, l'H. Influenza.
Nel 1990 si svilupparono vari metodi per il sequenziamento high-throughput, progetti che permisero di lanciare lo *Human Genome Project*, che fu completato nel 2000 quando pubblicarono in estate la prima bozza di genoma umano.
Le prime macchine semiautomatiche per il sequenziamento furono le *Biosystems ABI 370* ma oggi si usano i macchinari *Illumina*. Un macchinario *Illumina HiSeq 2000* corrisponde, in termini di prestazioni, a 23648 *Biosystems ABI 3730*, degli anni novanta.
Si hanno due tipi di attività parlando di NGS:

1. **Wet-Lab Activity**, ovvero le attività di raccolta dati/misure del materiale biologico, ovvero del vero e proprio sequenziamento tramite tecniche biochimiche. Si ha quindi la frammentazione e l'estrazione dei frammenti di DNA e RNA, il sequenziamento dei frammenti e la generazione delle read (con le 4 basi e caratteri extra per le ambiguità o i dati mancanti)

2. **Dry-Lab Activity**, ovvero le attività di assemblaggio. Si ha il salvataggio delle read, che sono tantissime (con conseguenti problemi di storage), e l'assemblaggio delle read (che sono *short read*) in *contings*, che sono read più lunghe. Dai contings si passa poi alla sequenza più ampia che stiamo sequenziando (anche un intero cromosoma o un intero genoma). Quest'ultimo è un problema prettamente algoritmico

Attualmente le tecnologie NGS producono read di lunghezza limitata (Illumina produce read da 70/150 basi circa) e il costo è proporzionale al numero delle read prodotte. Il parametro più importante è il parametro di **coverage**, ovvero il *depth of sequencing*, ovvero quante sequenze si hanno che coprono la medesima zona di DNA. Avere un alto coverage riduce il rischio di errore di sequenziamento ma un alto coverage implica alti costi e quindi è un parametro che va "bilanciato". Sono limiti tecnologici.
Il costo di sequenziamento, dal 2006, è sceso di molto e siamo ora intorno ai 1000 dollari per genoma (mentre nel 2000 eravamo intorno ai 100000 dollari). Nel 2006/2007 sono state infatti introdotte le tecnologie Illumina, molto più economiche. Anche nel 2015 si ha avuto un abbassamento e ora siamo in un plateau sui 1000 dollari. Inoltre, rispetto alla **legge di Moore**, il costo per genoma è sceso molto rispetto alla legge stessa.
Si hanno anche nuovi macchinari di sequenziamento, con una diversa tecnologia di base rispetto ad Illumina:

- **Single Molecule Real Time (*SMRT*) sequencing**, che sequenzia una molecola di DNA o RNA per volta
- **Nano Sensing sequencing**, che permette di avere un sequenziatore piccolissimo collegabile via USB al proprio computer. Si hanno problemi relative al software che ricostruisce le sequenze, avendo percentuali di errore veramente molto (con errori di natura diversa da quelli di Illumina, che sono comunque percentualmente molto minori)

Tra i tipi di sequenziamento abbiamo:

- **Whole-Genome Sequencing**, per interi genomi, anche *de-novo* (ovvero senza un *reference* preesistente)
- **Exome Sequencing**, per solo le parti di genoma codificanti (infatti solo alcune parti, poche, del genoma codificano le proteine mentre il resto del genoma non si sa bene a cosa serva)
- **Target (re)sequencing**, per zone specifiche del genoma, spesso sono misure secondarie dopo un Whole-Genome Sequencing per zone "dubbie" o che servono in quantità maggiore (magari perché legate a certe proteine)
- **RNA-seq**, sequenziando RNA, usando le tecnologie NGS per determinare l'attività dell'espressione genica (studiando che proteine sta generando una cellula etc. . .), caratterizzando i trascrittomi.

Si evitano i vari passaggi che si facevano con i microarrays, che davano una misura indiretta per di più (l'intensità della luce etc...). Qui basta sequenziare e poi contare le read di un particolare RNA

- si hanno ora molte altre ***-seq** in letteratura. Ad esempio **ATAC-seq (*Assay for Transposase-Accessible Chromatin using sequencing*)**, che è legato allo studio della conformazione tridimensionale del DNA, delle *aperture/chiusure della cromatina*, studiando cosa è trascrivibile in un dato istante oppure no

### 4.2.1 Dal Sequenziamento alle Analisi

Dopo il sequenziamento vogliamo vedere come tutte queste informazioni possono essere usate per fare analisi su come si comportano alcuni processi biologici, in particolare il **comportamento dei tumori**.
Una delle cose che si possono fare è prendere campioni di tumori da più pazienti e ricostruire le parti comuni dei tumori stessi, per ottenere i vari sottotipi del tumore. Questo studio è legato a certi tipi di tumore (si hanno circa 40 tipi di tumore in totale con i relativi sottotipi).
Un'altra cosa che si può fare è analizzare il tumore di un individuo che si è poi suddiviso in *primario* e *metastatico*, costruendo una **filogenia tumorale**. Per farlo si prendono campioni del tumore e si fa una cosiddetta **bulk analysis**, ovvero un'analisi aggregata prendendo un tessuto ed estraendo il DNA dal tessuto, ottenendo materiale genico da diverse cellule (perdendo l'individualità di ogni cellula ottenendo una misura "media"). Si hanno poi vari algoritmi per ottenere la filogenia, più o meno complessi, ricostruendo **l'albero della filogenia tumorale**, che parte da un tumore iniziale e poi presenta le varie differenziazioni che si sono sviluppate di quel tumore.
Ora si sta sviluppando anche la **special trascriptomic**, dove si sequenziano *slice* di tumori tenendo anche in considerazione la posizione del sequenziamento.

## 4.3 Single-Cell Analysis

In merito all'ultimo aspetto della sezione precedente, più di recente, si sono sviluppate tecnologie più sofisticate dal punto di vista chimico e fisico, per isolare singole celle prese da un campione. In questo modo si ha una rappresentazione più precisa di come sono fatte le popolazioni di cellule in un campione. Si usano poi algoritmi di filogenia per ricostruire le *evoluzioni clonali*. Questa tecniche sono dette appunto**tecniche di Single-Cell Analysis**. Si parte quindi sempre da un sequenziamento ma associato a singole

cellule.
La Single-Cell Analysis è cruciale in questo periodo e può essere usata per tantissimi progetti. In Bicocca si hanno progetti di **Metagenomica**, dove si isolano organismi da popolazione di organismi, sequenziando il singolo organismo (ma sequenziandone tanti). Viene fatto per studiare le popolazioni microbiche nelle falde acquifere o negli acquedotti. Si isolano organismi noti da organismi non noti, per riuscire poi a distinguerli e catalogarli, etichettandoli con il rispettivo materiale genico (il *corredo genomico*). Questo non era possibile con questa facilità prima dell'uso della Single-Cell Analysis.
Attualmente è comunque una tecnica molto costosa (contando che in un esperimento si sequenziano migliaia di singole cellule).

## 4.4 Risorse Online

Vediamo quindi una breve carrellata di risorse online importanti:

- **NCBI (*National Center for Biotechnology Informationy*)**, dove si hanno tutte le varie risorse più usate, ad esempio *PubMed* (per la ricerca di paper), *Blast* (uno dei più famosi allineatori di sequenze, nonché uno dei software informatici più usati al mondo), *Gene* (un importante database) etc... Si hanno inoltre modalità per trasmettere i risultati di ricerche, scaricare dati, informazioni su come interfacciarsi senza usare l'interfaccia web (per fare programmaticamente analisi più ampie tramite API), varie risorse per imparare le tecnologie, tutorial etc...
  Dal formato *SBML*, uno standard inspirato *XML*, con cui si rappresentano in modo standard i modelli poi si generano gli altri formati, tra cui i formati per *MATLAB*

- **BioModels**, dove si trovano modelli di sistemi biologici di varia natura (come vari modelli per il Repressilator). Tali modelli sono disponibili in vari formati (ad esempio per MATLAB etc...) e sono simulabili

- **BioCyc**, un database storico che contiene una rappresentazione di tutte le reazioni metaboliche di un organismo. Era nato originariamente per il metabolismo di E. Coli ed è stato poi generalizzato a vari organismi

- **KEGG (*Kyoto Encyclopedia of Genes and Genomes*)**, un portale giapponese che fornisce un insieme di database relativi

a vari dati di carattere biologico. Fornisce delle API, di recente riscritte per usare la terminologia REST, e altri tool

- **Pathway Commons**, un database per pathway metaboliche o regolatorie pubbliche
- **Firehose e Firebrowse**, un'interfaccia semplificata ad un database complesso chiamato **TCGA (*The Cancer Genome Atlas*)**. TCGA è un database, ora parte di NCBI, che raccoglie i dati di esperimenti che hanno misurato variazioni nel genoma relativi a tumori, e permette di scaricare in modo semplificato i vari dati relativi a tali tumori. Si hanno a disposizione vari tipi di studio tra cui, ad esempio, la *CopyNumber Analysis*. NOn tutti i tipi di tumori permettono di scegliere tutti i tipi di studio, non ancora perlomeno. Si nota che il cancro ai polmoni è quello più studiato

Ovviamente questa lista è solo introduttiva.

# Capitolo 5

# Introduzione ai Prerequisiti

Prima di proseguire è bene fare una breve digressione sui modelli delle reazioni chimiche al fine di poterne fare simulazioni tramite modelli matematici. Verrà quindi fatta una brevissima introduzione di **biologia molecolare** e di **biochimica**, con la rappresentazione di reazioni chimiche e la loro modellazione. Per farlo verrà ripreso l'esempio del Repressilator.
In primis conviene riprendere il concetto di **cooperatività** visto per il Repressilator, ovvero il valore rappresentato dal **coefficiente di Hill** $n$, da cui dipende il dominio dell'oscillazione. Si può quindi ricavare il coefficiente anche dall'analisi matematica.. Per capire cosa sia la *cooperatività* abbiamo bisogno di alcune nozioni di biochimica e di come le reazioni biochimiche siano state rappresentate nel mondo della computer science e della bioinformatica. Per cultura personale si elencano alcuni di questi sistemi:

- **BioNetGen**, un framework di modellazione *rule-based* ed esempio di linguaggio standard per modellare sistemi biologici
- **VCell**, un'altra piattaforma di modellazione
- **COPASI**, un software per la simulazione e l'analisi di reti biochimiche e della loro dinamica, nato per modelli stocastici ma poi passato anche ad altre tipologie
- **SBML**, un *linguaggio di markup* per modellare processi biologici
- **PySB**, una libreria in *Python* per la modellazione di sistemi biologici e biochimici mediante modelli matematici

## 5.1 Biochimica

La **materia** è studiata in varie forme, tra cui, in **biochimica**, quella di *miscele*. Le miscele possono essere:

- **omogenee**
- **soluzioni** con un *solvente* e un *soluto*. Ci sono vari metodi per separare il solvente dal soluto, specialmente metodi fisici/meccanici come usare una centrifuga

Alcune sostanze non sono però separabili usando semplici tecniche fisiche. Tali sostanze sono principalmente di due tipi:

- **sostanze pure**, come ad esempio acqua, sale etc...
- **composti**, come moltissime sostanze in natura

Le *sostanze pure* non possono essere separate ulteriormente tramite tecniche fisiche ma possono essere modificate da reazioni biochimiche. Una **reazione** coinvolge un certo numero $n$ di **reagenti** che portano ad un certo numero $m$ di **prodotti**. Le proprietà chimico-fisiche dei reagenti possono essere modificate e i prodotti della reazione possono essere composti con caratteristiche molto diverse da quelle dei reagenti. Come formalismo potremmo avere, indicando con $R_i$ i reagenti e $P_j$ i prodotti:

$$R_1 \oplus R_2 \oplus \cdots R_n \rightarrow P_1 \oplus P_2 \oplus \cdots P_m$$

Ovviamente i prodotti possono essere separabili o diventare a loro volta reagenti.
Ci sono inoltre composti che non possono essere modificati a livello chimico e questi sono gli **elementi**. Gli atomi sono l'elemento minimo da considerare per parlare di reazioni biochimiche a livello cellulare e sono composti, come si sa, da:

- il **nucleo**, con **protoni** e **neutroni**
- gli **elettroni**, che si trovano in un'orbitale quantizzato. Ogni orbitale contiene fino a 8 elettroni (tranne il primo che ne contiene massimo solo 2) e quindi si parla di *octet rule*. L'orbitale più esterno è *completo* solo nei cosiddetti **gas nobili** mentre negli altri è *incompleto*. Il numero di elettroni nell'ultimo orbitale rappresenta la **valenza dell'atomo**

La configurazione dell'orbitale più esterno permette agli stessi di legarsi in composti. Le **molecole** sono i composti più piccoli e, se divise, cambiano le loro proprietà chimiche. La struttura delle molecole dipende dagll'organizzazione degli elettroni dell'ultimo orbitale condivisi dagli atomi. Gli atomi con valenza fino a 4, detti **donors**, tendono a donare elettroni agli atomi con valenza da 5 a 7, detti **receptors**, che si dice hanno una **tendenza elettronegativa**.
Uno strumento essenziale in tale ambito è la **tavola periodica**. La tavola periodica ci fornisce informazioni su ogni elemento conosciuto in natura, nonché i nuovi elementi sintetizzabili in esperimenti nucleari.
Si hanno vari modi in cui gli atomi *legano* tra loro:

- **legame ionico**, tra atomi con una valenza molto diversa (ad esempio $NaCl$, dove $Na$ ha valenza 1 e $Cl$ ha valenza 7)
- **legame covalente**, tra atomi con una valenza simile (questo succede spesso con molecole di atomi dello stesso tipo, come $Cl_2$)
- **legami doppi**, possibili in altre configurazioni (ad esempio Carbonio di valenza 4 e due atomi di Ossigeno, che, a loro volta, in coppia comportano valenza 4, condividono due coppie di elettroni per formare la $CO_2$)

Un altro concetto importante è quello di **polarità**. Le molecole hanno una polarità, a seconda dell'elettronegatività di ciascun atomo partecipante e della loro configurazione spaziale. Ad esempio:

- le molecole risultanti dai legami tra $O$ e $H$ tendono ad essere *polari*, poiché l'elettronegatività di $O$ e $H$ è abbastanza diversa
- le molecole risultanti dai legami tra $C$ e $H$ tendono invece ad essere *non-polari*, poiché l'elettronegatività di $C$ e $H$ è simile

Molecole polari tendono ad *attrarsi* mentre quelle non-polari a sono *neutre* (ad esempio l'acqua è polare mentre olio è non-polare). Nel dettaglio, le molecole che non si mischiano bene con l'acqua sono dette **idrofobiche**.
Le forze che creano i legami tra gli atomi sono anche responsabili dell'*attrazione* tra atomi e molecole. Tra esse si ha la **forza elettrostatica** che agisce tra atomi e molecole. Un'altra forza è la **forza di Van der Waals ($vdW$)** che, a causa di effetti quantistici, attrae le molecole a "lunghe" distanze e allontana quelle a "corte" distanze.
Una forza di attrazione intermedia è quella risultante dal cosiddetto **legame a idrogeno**. Legami di questo tipo sono essenziali in biologia (basti vedere

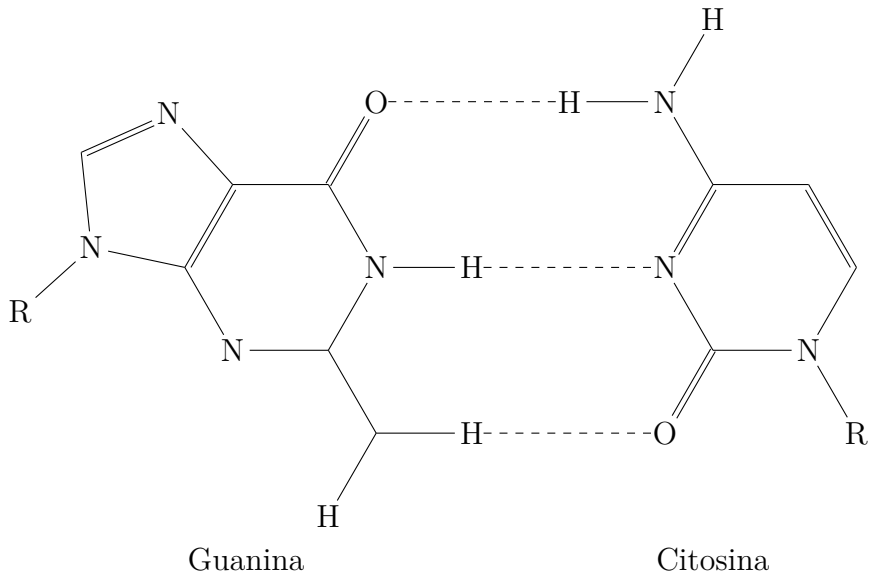



Figura 5.1: I tre legami a idrogeno tra Guanina e Citosina

il DNA e il legame tra le basi azotate, come in figura 5.1 e 5.2) in quanto il Carbonio è abbastanza elettronegativo da essere un *donor* per il *legame a idrogeno*. Tale legame è quello che viene "rotto" con la **polimerasi**.
I legami ionici, covalenti e doppi sono **legami forti** mentre il legame a idrogeno, la forza di Van der Waals e la forza elettrostatica sono **legami deboli**.
*Ogni atomo di carbonio forma 4 legami con altri atomi, l'Azoto 3, l'ossigeno 2 e l'Idrogeno 1. Nelle figure delle molecole dove non si ha nulla indicato si ha un Carbonio.*

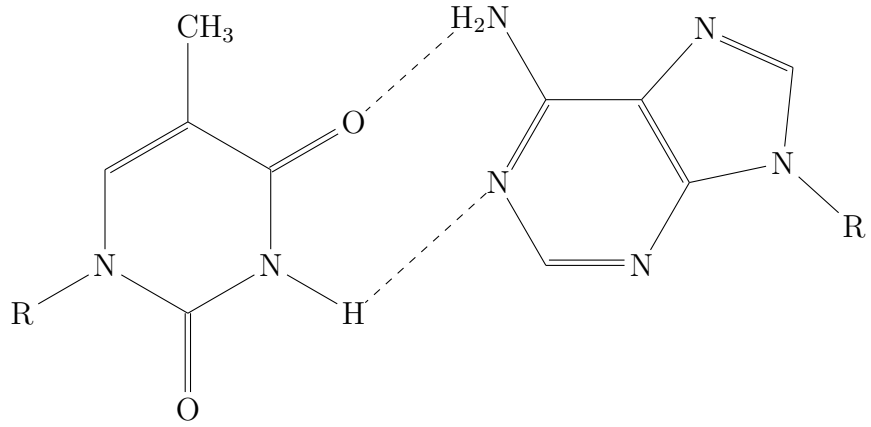



Figura 5.2: I due legami a idrogeno tra Timina e Adenina

### 5.1.1 Biochimica e Metabolismo

Abbiamo visto come le reazioni biochimiche modificano le proprietà di vari composti.
Una delle reazioni più semplici è quella detta **dissociazione**. Uno degli esempi tipici è il sale, $NaCl$, che dissocia in $Na^+$ e $Cl^-$ a causa della polarità delle molecole d'acqua. Si noti che questa reazione è **reversibile**.
Un'altra reazione semplice, stavolta **irreversibile**, che coinvolge più composti è la combustione dell'Idrogeno, che potremmo scrivere come:

$$2H_2 + O_2 \rightarrow 2H_2O$$

*Il rapporto quantitativo tra i composti in una reazione è chiamato* ***stechiometria della reazione*** e l'**energia** è la chiave di questi ragionamenti .
Per passare dal rapporto espresso con la stechiometria e le unità di misura della fisica (come i grammi) e viceversa si introduce una nuova unità di misura. Un esempio famoso dice che:

*Un grammo di Mercurio, $Hg$, contiene un numero diverso di molecole rispetto ad un grammo di Potassio, $K$.*

I chimici quindi, usando la teoria della *fisica statistica*, hanno introdotto il concetto di **mole (*mol*)**, definita come la misura della *quantità di sostanza*. La mole è definita come la quantità di sostanza che contiene esattamente $6,02214076 \times 10^{23}$ entità fondamentali, essendo questo il valore numerico della costante di Avogadro quando espressa in $mol^{-1}$. In altri termini è la quantità di sostanza che pesa esattamente il suo peso molecolare (ad esempio una mole di Ossigeno, $O_2$, pesa circa $32g$ mentre una di Idrogeno, $H_2$, circa

$2g$).
Tra le caratteristiche principali di una reazione abbiamo il **reaction rate**, ovvero la *velocità* con cui avviene la reazione stessa. In chimica la **cinetica** è lo studio dei vari fattori che influenzano i *reaction rate*. Tra questi fattori abbiamo:

- temperatura
- concentrazione di reagenti (nelle reazioni biochimiche con importanti effetti biologici spesso la concentrazione di un dato reagente è molto piccola)
- ...

Data una reazione, quando la concentrazione di un reagente è molto bassa, o il *reaction rate* è molto lento, allora si dice che la reazione è **cineticamente alterata** (sebbene questo termine sia solo evocativo).
Un tipo molto importante di reazione è quello che implica il trasferimento di elettroni da una molecola all'altra. Queste reazioni sono dette **oxy-reduction** o anche **redox**. Il composto che cede l'elettrone si dice che viene **ossidato** (il nome deriva dal fatto che l'Ossigeno è l'agente ossidante per eccellenza ma non è l'unico) mentre di quello che lo riceve viene detto che si **riduce**, ovvero diventa "più negativo".

### Termodinamica

Si introduce anche qualche concetto di base di **termodinamica**.
Le reazioni possono avven ire se è presente energia e le reazioni in sé corrispondono ad un cambio di energia nel sistema in analisi. Si ha che l'energia **si conserva** e in biochimica anche la quantità complessiva di materia si conserva. Le reazioni che necessitano energia/calore per avvenire sono dette **endotermiche** mentre quelle che generano energia **esotermiche**.
La quantità di energia interna che un sistema termodinamico può scambiare con l'ambiente è detta **entalpia**. Essa non può essere direttamente misurata ma possiamo misurare la sua variazione $\Delta H$, avendo:

$$H = U + p \cdot V$$

con:

- $U$ energia interna
- $p$ pressione

- $V$ volume

Non tutta l'energia è disponibile per il lavoro infatti si ha che una parte di essa viene dispersa e non è quindi utilizzabile. Questa nozione di dispersione è formalizzata in termodinamica come l'**entropia** del sistema. Il cambiamento di energia complessiva del sistema lo possiamo calcolare come:

$$\Delta U = T \cdot \Delta S - w$$

con:

- $T$ temperatura iniziale del sistema
- $S$ entropia
- $w$ lavoro

L'energia che è effettivamente disponibile per compiere il lavoro è detta **energia libera di Gibbs**, che si calcola come:

$$\Delta G = \Delta H - T \cdot \Delta S$$

Le reazioni biochimiche devono essere fattibili dal punto di vista termodinamico, avendo quindi, per esempio, che l'energia libera di Gibbs disponibile deve essere sufficiente per iniziare la reazione. La fattibilità termodinamica della reazione comunque non implica che essa avverrà spontaneamente (ad esempio la combustione del Metano è esotermica ma Metano e Ossigeno si mischiano senza far partire la reazione a temperatura ambiente). Si ha quindi che una reazione può avvenire solo se si ha $\Delta G > 0$ e $\Delta G$ è sufficiente a superare la **barriera di attivazione** della reazione. L'energia necessaria per superare tale barriera è detta **energia di attivazione** e ogni reazione ha una propria barriera/energia di attivazione.
Durante una reazione l'energia complessiva del sistema cambia. Gran parte dell'energia di un sistema biochimico è contenuta nei legami tra i vari composti, avendo quindi la cosiddetta **energia di legame**. Gli organismi devono costruire e distruggere questi legami per vivere/riprodursi e possono farlo anche eseguendo reazioni non termodinamicamente fattibili, quindi utilizzando energia per superare le barriere di attivazione. Quando si libera energia per degradare molecole complesse in molecole più semplici si parla di **catabolismo** (come nel caso della *glicolisi*) mentre se si consuma energia per sintetizzare molecole complesse da molecole più semplici si parla di **anabolismo** (come nel caso della *gluconeogenesi*). Gli organismi, di conseguenza, hanno bisogno di effettuare delle reazioni per generare energia e uno dei modi

più comuni è quello di rompere un legame fosfato nell'**ATP (*adenosina-trifosfato*)**, ottenendo/liberando **ADP (*adenosina-difosfato*)**. L'energia contenuta nei legami fosfato dell'ATP è sufficiente per attivare molte reazione biochimiche anche se non è sufficiente per molti degli altri tipi di legame presenti in un organismo. Per acquisire nell'organismo l'energia presente nell'ATP si hanno vari modi, tra cui nutrirsi o fare la fotosintesi. Anche il *grasso* è un modo per conservare energia atta alle azioni base: riprodursi, eventualmente muoversi, mangiare e non morire. Interessante è notare che le cellule tumorali si rifiutano di fare **apoptosi**, che è una procedura di morte controllata utile negli esseri viventi.

### Metabolismo

Ciò che l'organismo continua a fare per sopravvivere e riprodursi è accumulare energia per consumare e sintetizzare complessi biochimici e questa attività, che per lo più avviene nel citoplasma delle cellule, è detta **metabolismo** e si hanno due tipologie:

1. **catabolsimo**, ovvero reazioni di decomposizione di vari complessi, per lo più acquisiti dall'ambiente
2. **anabolismo**, ovvero la sintesi di complessi

Questa divisione è presente anche nella *GeneOntology*.
*Con "complessi" qui si intende "molecole complesse".*
I vari organismi condividono il funzionamento di molte reazioni di base, parlando quindi di **metabolismo centrale (*core metabolism*)**, mentre i meccanismi specializzati prendono il nome di **metabolismo secondario (*secondary metabolism*)**.
Un processo molto importante è quello che permette ad un organismo di "caricare" molecole di *ADP* con un gruppo fosfato, generando così *ATP* e per farlo si ha una catena di reazioni (molte delle quali con alta energia di attivazione o basso reaction rate), ovvero i **metabolic pathways**. Per "accelerare" queste reazione gli organismi usano il meccanismo della **catalisi**, in quanto un **catalizzatore** accelera una reazione o ne riduce l'energia di attivazione senza essere "consumato" durante la reazione stessa. Si ha quindi che il catalizzatore è sia un reagente che un prodotto della reazione complessiva. I catalizzatori sono chiamati **enzimi**, agendo su materiali/sostanze dette **substrati**.
Tra i pathway metabolici principali abbiamo (**su slide immagini dei pathways** e della **Metabolic Map**):

- **glicolisi**

- **ciclo di Krebs**

## 5.2 Modellazione Matematica

Bisogna capire come modellare matematicamente le reazioni biochimiche. Si useranno:

- la **legge di azione di massa**
- le **equazioni di Michaelis-Menten**
- la **cooperatività**, mediante l'**equazione di Hill**

Uno dei punti chiave della biologia computazionale è modellare **reaction network (*reti di reazioni*)** che si possono anche suddividere in:

- **metabolic network**, con lo studio di reazioni che riguardano molecole, proteine etc...
- **regulatory network**, con lo studio di interazioni di geni e proteine, studiando, promozione della trascrizione, inibizione, etc...

*L'interazione spaziale a scala "meso" tra elementi cellulari separati e tra cellule sarà trattata separatamente.*

### 5.2.1 Legge di Azione di Massa

Partiamo dalla **legge di azione di massa (*Law of Mass-Action*)**.
La collisione tra due composti chimici, che sia tra due macromolecole o anche solo tra due ioni, che chiamiamo $A$ e $B$, accade con un certo *reaction rate* $k$, e produce un composto $C$ come risultato. Indichiamo formalmente questo con:

$$A + B \xrightarrow{k} C$$

Il reaction rate $k$ è dovuto a vari aspetti:

- la configurazione geometrica di $A$ e $B$
- la temperatura
- altri parametri ambientali

Inoltre la legge si applica a sistemi che sono **in equilibrio** e non sempre è applicabile, ad esempio quando uno dei composti è presente a concentrazioni molto elevate, infatti in tal caso può essere che gli effetti risultanti non obbediscano alla semplice relazione che deriva dalla formulazione della legge.
Possiamo riscrivere il formalismo della legge in modo da rimuovere $\rightarrow$ e usare la notazione delle derivate, ottenendo un'**equazione differenziale ordinaria (EDO)**, in inglese **ordinary differential equation (*ODE*)**, che sia continua e deterministica. Indichiamo inoltre con $[X]$ la concentrazione del composto $X$. Otteniamo quindi:

$$\frac{\mathrm{d}[C]}{\mathrm{d}t} = k[A][B]$$

A causa della termodinamica possiamo inoltre considerare **reazioni bidirezionali**:

$$A + B \underset{k_-}{\overset{k_+}{\rightleftarrows}} C$$

Una EDO per queste reazione, ad esempio dal punto di vista di $A$ sarebbe:

$$\frac{\mathrm{d}[A]}{\mathrm{d}t} = k_-[C] - k_+[A][B]$$

Come detto prima vogliamo che il sistema sia in equilibrio in quanto, in tal caso, le concertazioni dei composti non cambiano e quindi vale la seguente condizione, data la precedente equazione:

$$\frac{k_-}{k_+} = k_{eq} = \frac{[A]_{eq}[B]_{eq}}{[C]_{eq}}$$

Dove il rapporto $k_{eq}$ è la **costante di equilibrio** della reazione e, qualora $k_{eq}$ sia piccolo, si ha indicazione del fatto che $A$ e $B$ sono state effettivamente "unite" in $C$. Si ha che $k_-$ e $k_+$ sono relativi alla reazione bidirezionale.

### 5.2.2 Equazioni di Michaelis-Menten e Hill

Passiamo ora a formalizzare la **cinetica enzimatica**, tramite le **equazioni di Leonor Michaelis e Maud Mentes**.
Reazioni non elementari, ovvero quelle che non seguono la *legge di massa di azione*, come ad esempio le reazioni enzimatiche, necessitano la seguente rappresentazione (con a destra una semplificazione "empirica" della stessa):

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} C \underset{k_{-2}}{\overset{k_2}{\rightleftarrows}} P + E \qquad S + E \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} C \overset{k_2}{\rightarrow} P + E$$

Dove:

- $S$ è il substrato
- $E$ è l'enzima
- $C$ è il prodotto intermedio
- $P$ è il prodotto della reazione

Quindi da un substrato e l'enzima otteniamo prima un prodotto intermedio con una prima reazione e poi il prodotto finale con ancora l'enzima, tramite una seconda reazione, avendo quindi che l'enzima è come se non fosse modificato.
Normalmente si considera solo la forma semplificata, la seconda, dove si ha solo una reazione bidirezionale mentre la seconda diventa unidirezionale/irreversibile.
*Da queste forme possiamo formulare le equazioni di Michaelis-Menten.*
L'obiettivo principale di Michaelis e Menten era quello di caratterizzare i processi di fermentazione, quindi ciò che cercavano erano misure dell'efficienza di una reazione enzimatica e della sua velocità.
Riprendiamo la formula $A + B \overset{k}{\rightarrow} C$, e la sua versione differenziale $\frac{\mathrm{d}[C]}{\mathrm{d}t} = k[A][B]$ per poter ottenere le due equazioni.

Avendo che $[E]_0$ è la quantità disponibile di enzima e che $[E]+[C] = [E]_0$ possiamo riscrivere lo schema della *legge di massa di azione* otteniamo quattro ODE:

$$\frac{\mathrm{d}[S]}{\mathrm{d}t} = k_{-1}[C] - k_1[S][E]$$
$$\frac{\mathrm{d}[E]}{\mathrm{d}t} = (k_{-1} + k_2)[C] - k_1[S][E]$$
$$\frac{\mathrm{d}[C]}{\mathrm{d}t} = k_1[S][E] - (k_2 + k_{-1})[C]$$
$$\frac{\mathrm{d}[P]}{\mathrm{d}t} = k_2[C]$$

Queste quattro equazioni rappresentano la variazione delle quattro "specie" considerate rispetto alle altre nel tempo. Si nota che sono le reazioni relative alla formulazione semplificata in quanto si nota, nella quarta, come $P$ dipenda solo da $C$ ma non si ha modo di diminuire la velocità di generazione di $P$, non avendo $k_{-2}$ (nella terza equazione, ad esempio, si vede l'effetto della bidirezionalità tramite i coefficienti $k$).
Consideriamo quindi la concentrazione totale dell'enzima $[E]_0$ (in pratica è la concentrazione iniziale dell'enzima) e assumiamo tutti i reaction rate costanti (è un'assunzione non trascurabile ma semplifica molto il problema). Se osserviamo le precedenti equazioni otteniamo che la velocità a cui cresce la concentrazione di $[P]$ è:

$$V = \frac{\mathrm{d}[P]}{\mathrm{d}t} = k_2[C] \approx [E][S]$$

e quindi $V$ è proporzionale alla concentrazione di $[E]$ e $[S]$.
Si ipotizza ora che tutto l'enzima $[E]_0$ sia "esaurito". A quel punto non importa se aumentiamo il substrato, non c'è modo di combinare più enzimi e quindi la velocità della reazione, ovvero la velocità di produzione del prodotto $P$, raggiunge il suo massimo che chiamiamo $V_{max}$. Per dire che una reazione può raggiungere una certa $V_{max}$ si dice che **satura** a $V_{max}$.
Si ha che $\frac{V_{max}}{2}$ la si ottiene ad un certo $K_M$, che è una concentrazione, che verrà a breve approfondito, come visibile in figura 5.3.

Il *sistema di Michaelis-Menten* si risolve analiticamente con una cosiddetta *approssimazione di equilibrio*, con la quale si assume che il substrato $S$ e il complesso intermedio $C$ sono istantaneamente in equilibrio. Questo è comodo perché non sempre si possono ottenere delle soluzioni analitiche (dovendo quindi ricorrere obbligatoriamente a risoluzioni numeriche) ma in questo caso specifico sì.

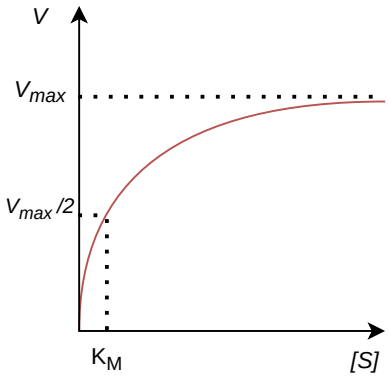



Figura 5.3: Grafico approssimativo rappresentante l'andamento della velocità di reazione.

Si ha quindi che si può inferire:

$$k_1[S][E] = k_{-1}[C] + k_2[C]$$

Sapendo poi che $[E]+[C] = [E]_0$, in quanto la quanittà iniziale di enzima deve essere proporzionale a $[E]+[C]$, avendo che all'inizio, quando ho l'enzima allo stato iniziale $[E]_0$ ho $[C] = 0$. In realtà dovremmo scrivere $[E]+[C] \approx [E]_0$ ma l'uguaglianza è al momento un'approssimazione accettabile quindi, avendo $[E] = [E]_0-[C]$ (che sarebbe in realtà $[E] \approx [E]_0-[C]$), si ha che, sostituendo e raccogliendo:

$$k_1[S]([E]_0 - [C]) = (k_{-1} + k_2)[C]$$

e quindi:

$$[S][E]_0 - [S][C] = [C]\left(\frac{k_{-1} + k_2}{k_1}\right)$$

Introduciamo quindi il già anticipato $K_M$, che è un combinazione vari di reaction rate, che nel caso semplificato, con la seconda reazione irreversibile, è della forma:

$$K_M = \left(\frac{k_{-1} + k_2}{k_1}\right)$$

Facciamo ora un poco di manipolazione delle equazioni già ottenute, ottenendo, introducendo $K_M$, che:

$$[C](K_M + [S]) = [E]_0[S]$$

e quindi:

$$[C] = \frac{[E]_0[S]}{K_M + [S]}$$

Ricordando quindi che la velocità di creazione di $P$, ovvero il reaction rate, è:

$$V = \frac{\mathrm{d}[P]}{\mathrm{d}t} = k_2[C]$$

Sapendo poi che a $V_{max}$ si ha che che tutto l'enzima $E_0$ deve essere legato in $C$, si ottiene:

$$V_{max} = k_2[C] = k_2[E]_0$$

Ma allora, rimettendo insieme le varie equazioni facendo le varie sostituzioni:

$$V = \frac{\mathrm{d}[P]}{\mathrm{d}t} = k_2[C] = k_2\frac{[E]_0[S]}{k_m + [S]} = \frac{V_{max}[S]}{k_M + [S]}$$

e quindi si ottiene, tenendo solo gli estremi, l'**equazione di Michaelis-Menten**:

$$V = \frac{V_{max}[S]}{k_M + [S]}$$

Tale equazione ci dice che, se sappiamo la massima velocità della reazione, siamo in grado di regolare la velocità della reazione stessa (ovvero del substrato che produce il prodotto finale) semplicemente andando a modificare la concentrazione iniziale.
Michaelis e Menten hanno così potuto regolare i processi di fermentazione della birra che stavano studiando.
Studiamo meglio $K_M$, che è detta **costante di Michaelis-Menten**. Uno studio dimensionale sull'equazione di questa costante porta a verificare che è una concentrazione. Facendo vari conti possiamo arrivare ad asserire che:

$$K_M \approx [S]$$

ovvero si ha che $K_M$ è proporzionale alla concentrazione del substrato.
Sostituendo nell'equazione di Michaelis-Menten, si ottiene, come già in parte anticipato, che:

$$V = \frac{1}{2}V_{max}$$

Si arriva quindi ad una definizione.

**Definizione 3.** *Si definisce $K_M$, detta **costante di Michaelis-Menten**, come la concentrazione del substrato $S$, ovvero $[S]$, tale per cui si ha $V = \frac{1}{2}V_{max}$.*
*Questa costante è anche usata per definire la nozione di **efficienza catalitica**, tramite passaggi ulteriori non specificati nel corso.*

Possiamo finalmente discutere il significato di **cooperatività**.
Per molti enzimi la velocità di reazione segue la forma di un **sigmoide**, che è diversa da quella ad **iperboloide** ottenuta da Michaelis e Menten con el loro equazioni. Questo accade, ad esempio, quando un enzima può legarsi a più di un substrato e il primo legame facilita quello successivo. Questo fatto è stato scoperto sperimentalmente.
Ad esempio potremmo avere la seguente situazione, con solo complessi intermedi $C_1$ e $C_2$:

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} C_1 \xrightarrow{k_2} P + E$$

$$S + C_1 \underset{k_{-3}}{\overset{k_3}{\rightleftarrows}} C_2 \xrightarrow{k_4} C_1 + P$$

Nella prima reazione abbiamo la serie di "passaggi standard", come già studiato, mentre nella seconda si nota l'intervento del primo complesso intermedio nella reazione che porta al secondo complesso intermedio, il quale porterà al prodotto finale più ancora il primo complesso intermedio.
Si è modellato quindi un enzima che si può attaccare in più di un modo ad un substrato, in questo caso abbiamo che si attacca in due modi al substrato (avendo nel complesso 6 reazioni). Ovviamente quello che qui si è visto con due equazioni può essere generalizzato a $m$ equazioni. Con altri passaggi matematici potremmo trovare una forma generale per la velocità, ottenendo l'**equazione di Hill**, che in questo caso è:

$$V = \frac{V_{max}[S]^n}{K_M^n + [S]^n}$$

con $n$ numero di siti in cui l'enzima può legarsi al substrato $S$ e con $K_1, K_2, \ldots, K_l$ che sono le $l$ constanti all'equilibrio, avendo che:

$$K_m^l = \prod_{i=0}^{l} K_i$$

Si noti inoltre che con $n = 1$ otterremmo l'**equazione di Michaelis-Menten**.
L'**equazione di Hill** è usata per modellare reazioni che si pensa siano cooperative, ovvero con enzimi che possono legarsi in più di un substrato, ma i cui dettagli non sono completamente noti. Non si sa come l'enzima si attacca al substrato ma si stima che dovrebbe farlo in un certo numero di siti.
A partire dall'**equazione di Hill** si può tornare al Repressilator e fare le giuste considerazioni sul coefficiente di Hill $n$.

Per fare funzionare comunque un qualsiasi sistema con EDO abbiamo bisogno delle **costanti**. Si hanno sostanzialmente due modi per conoscere le costanti:

1. si ricercano sperimentalmente in laboratorio di biologia provando diverse condizioni in cui avviene una reazione e se ne misura la velocità, misurando le concentrazioni prima e dopo un certo tempo, inferendo poi le costanti. Per sistemi molto grandi è impraticabile

2. si usano metodi computazionali per fare *esplorazione dello spazio dei parametri*, ovvero il **parameter sweeps**. Si hanno diversi metodi di tipo stocastico o anche metodi più "controllati", che danno certezza di aver esplorato buona parte dello spazio dei parametri, usando i metodi detti di *ricerca su griglia*, ovvero i metodi **grid search**. Anche in questo caso è una ricerca sperimentale dei parametri ma dal punto di vista computazionale

Nel dettaglio i parametri $n$, $V_{max}$ e $K_M$ devono essere determinati in uno di questi modi.
Si possono fare alcune considerazioni su $n$:

- per $n > 1$ si hanno legami che cooperano positivamente, avendo che, non appena un *ligando*, ovvero l'enzima, si lega a una molecola, l'*affinità di attrazione* per gli altri ligandi aumenta

- per $n < 1$ si hanno legami che cooperano negativamente, avendo che, non appena un *ligando*, ovvero l'enzima, si lega a una molecola, l'*affinità di attrazione* per gli altri ligandi diminuisce

- per $n = 1$ non si hanno legami che cooperano, avendo che l'*affinità di attrazione* dei ligandi non dipende da quanti di loro erano già attaccati alla molecola o meno

*In base a quanto detto Elowitz and Leibler scelsero sperimentalmente* $n = 2$*, in quanto rappresentava il* comportamento critico *del sistema dal punto di vista teorico. Decisero quindi quale tipo di promotori e inibitori (e sincronizzatori) utilizzare durante la progettazione del proprio esperimento per il Repressilator, portando alla scelta precisa delle 3 proteine usate, grazie alla loro conoscenza di biochimica.*
L'obiettivo di ogni sforzo di modellazione (che mira a modellare, alla fine, le catene di reazioni, modellate in pathway), infine, è osservare comportamenti plausibili con l'obiettivo di poter prevedere quelli imprevisti.
Per la modellazione si ha un elenco molto esteso di EDO, oltre alle due

appena viste (quella di Michaelis-Menten e quella di Hill), tra cui quelle per il **diagramma di Lineweaver-Burk**. Quest'ultima si ottiene a partire dall'equazione di Michaelis-Menten, facendone il reciproco:

$$\frac{1}{V} = \frac{K_M + [S]}{V_{max}[S]} = \frac{K_M}{V_{max}}\frac{1}{[S]} + \frac{1}{V_{max}}$$

È, in pratica, la **linearizzazione** dell'equazione di Michaelis-Menten e infatti si ottiene una formula rappresentante una retta, con $m = \frac{K_M}{V_{max}}$ e $q = \frac{1}{V_{max}}$. Tale retta incontra l'asse $y$, dove si ha $\frac{1}{V}$, in $\frac{1}{V_{max}}$ mentre l'asse $x$, dove si ha $\frac{1}{[S]}$, in $-\frac{1}{K_M}$.

# Capitolo 6

# Simulazioni Deterministiche e Ibride

Si analizza ora come andare a fare simulazioni tramite equazioni differenziali ordinarie, le EDO. Si introducono quindi gli algoritmi numerici, i *risolutori*, per le EDO. Si parla quindi di **modelli di simulazione deterministici**.
Parlando di modellazione si hanno varie teorie in uso:

- equazioni differenziali, per **modelli continui**
- sistemi discreti, come *automi a stati finiti*, *reti di Petri* e *dataflow diagrams* (che non verranno trattati) per **modelli discreti**
- tecniche per **modelli ibridi**, ovvero modelli continui ma con discontinuità e cambi di stato/modalità

Un componente importante in tutti i tipi di modellazione è il **tempo**, inteso come una delle variabili indipendenti del sistema e caratteristica fondamentale del modello e degli strumenti che si usano per le simulazioni.
Una delle cose fondamentali da chiedersi studiano il *tempo* è cosa costituisce il **progresso**. Una volta pensato a come modellare il tempo si ha un modo per capire come procedere a implementare un simulatore etc. . ..
Dal punto di vista computazionale il tempo è **discretizzato**, come del resto qualsiasi altra variabile. In ogni caso si classificano i *framework di modellazione* in base alla loro "visione di base" di come il tempo avanza.

## 6.1 Equazioni Differenziali Ordinarie

Le **equazioni differenziali ordinarie (*EDO*)** sono un modello standard per modelli fisici, biologici, ingegneristici etc. . . e assumono un campo reale

dove c'è una variabile, il tempo, reale (e quindi continua).
La forma generale di una EDO è:

$$\dot{y}(t) = F(y(t), t)$$
$$y(t_0) = k$$

$F$ è una funzione, arbitrariamente complessa, che è sia sul tempo che sulla funzione da calcolare. Si sanno manipolare particolari forme di $F$, tendenzialmente lineari. Si segnala anche la definizione di $y$ al tempo zero, data da una costante $k$. Quest'ultima è la **condizione iniziale** e bisogna averla per forza. Risolvere l'equazioni differenziale corrisponde a trovare la forma di $y$ e si dice essere equivalente a **risolvere un problema iniziale (*initial problem solving*)**.
Riprendiamo, ad esempio, le equazioni del Repressilator:

$$\frac{\mathrm{d}m_i}{\mathrm{d}t} = -m_i + \frac{\alpha}{1 + p_j^n} + \alpha_0$$
$$\frac{\mathrm{d}p_i}{\mathrm{d}t} = -\beta(p_i - m_i)$$

Dove si hanno, si ricorda 3 copie di equazioni differenziali, relativamente semplici, al più del $p_j^n$ a denominatore nella prima equazione.

## 6.2 Modelli Discreti

Nei **modelli discreti** si ha un'idea molto diversa del *tempo*, avendo una nozione derivata dall'osservazione dell'evoluzione del sistema.
Ad esempio con le **finite state automata (*FSA*)** abbiamo una rappresentazione di come l'evoluzione del sistema possa passare attraverso vari stati. Con le **reti di Petri** abbiamo una versione succinta dei FSA con delle eventuali estensioni. Si usano specialmente reti di Petri che sono codifiche che dal punto di vista teorico sono più espressive dei FSA, in quanto si riconosce un insieme di linguaggi che contiene quello riconosciuto dai FSA.
Si ha poi una particolare implementazione di FSA o di reti di Petri, detta **Discrete Event Systems (*DES*)**, ovvero una rappresentazione molto "operativa" di quei modelli, dove si ha una coda di eventi generata da *diversi sorgenti* e processata da vari *componenti*. La generazione degli eventi è normalmente associata ad un tempo, tempo in cui l'evento accadrà, e se si hanno più componenti bisogna ordinare i vari elementi generati.
Il **tempo** è normalmente una *nozione derivata* nei modelli discreti ottenuto da un'osservazione di sequenze di eventi o un'osservazione di un tempo esterno, rappresentabile a sua volta da un FSA o da una serie di eventi, parlando **Wall Clock**.

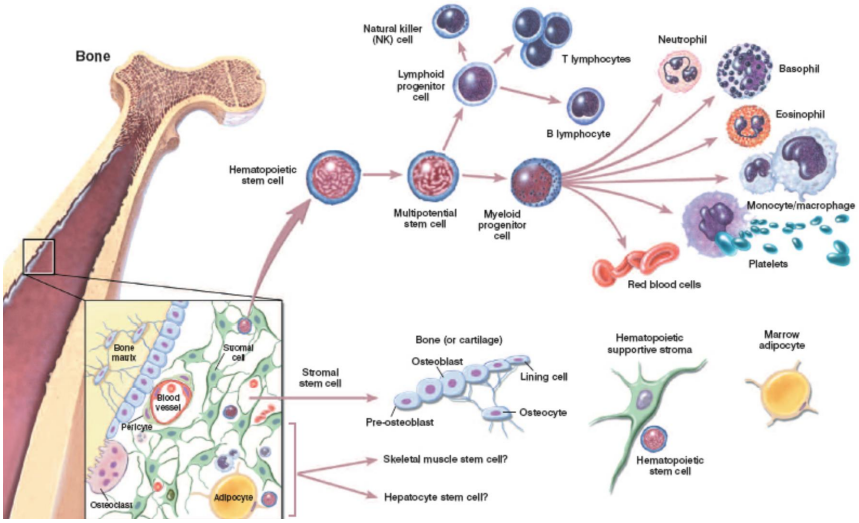



Figura 6.1: Schema completo del sistema ematopoietico con le varie suddivisioni delle cellule.

### 6.2.1 Modellazione di Cellule Staminali

Vediamo quindi l'uso di modelli discreti per la rappresentazione di una **popolazione di cellule staminali**, studiandone *proliferazione* e *differenziazione*. Nel dettaglio si parla di **cellule staminali ematopoietiche**, ovvero quelle che danno origine a tutte le cellule del sangue (si hanno infatti vari tipi di cellule staminali, come quelle neurali, quelle muscolari, quelle epiteliali etc...). Una cellula staminale si differenzia in una serie di possibili **cellule progenitrici** che sono più differenziate e che infine si differenziano totalmente in cellule "finali", nel caso del sangue, in analisi:

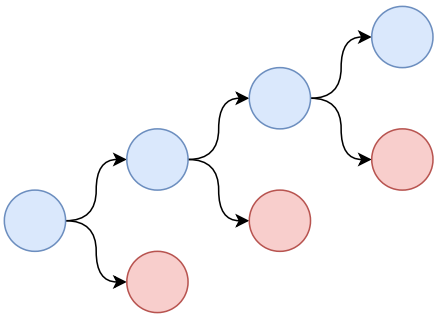
- cellule T
- globuli rossi
- ...

Con **sistema ematopoietico** si intende il sistema che da luogo alle cellule che compongono il nostro sangue. Le cellule ematopoietiche derivano dalle **cellule stromali** contenute nel *midollo osseo*.
Modellazione del sistema in figura 6.1 [1]

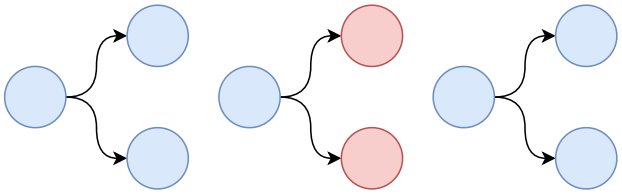
[1] `https://stemcells.nih.gov/info/Regenerative_Medicine/2006Chapter2.html`

Possiamo quindi dividere la divisione e la proliferazione delle cellule staminali in tre tipologie:

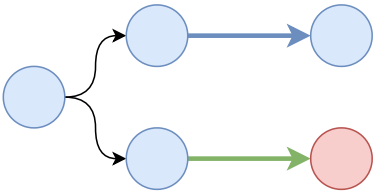
1. **divisione asimmetrica**, che è quello che si nota quando da una cellula staminale si produce una cellula differenziata:



2. **divisione simmetrica**, in cui le cellule staminali si dividono in due cellule staminali o in due cellule completamente differenziate:



3. **divisione ambientalmente asimmetrica**, anche se non è propriamente rappresentato, dove si può avere una divisione simmetrica o asimmetrica a seconda della presenza di un certo microambiente (con la conseguente presenza di determinate proteine) in cui si trova la cellula:

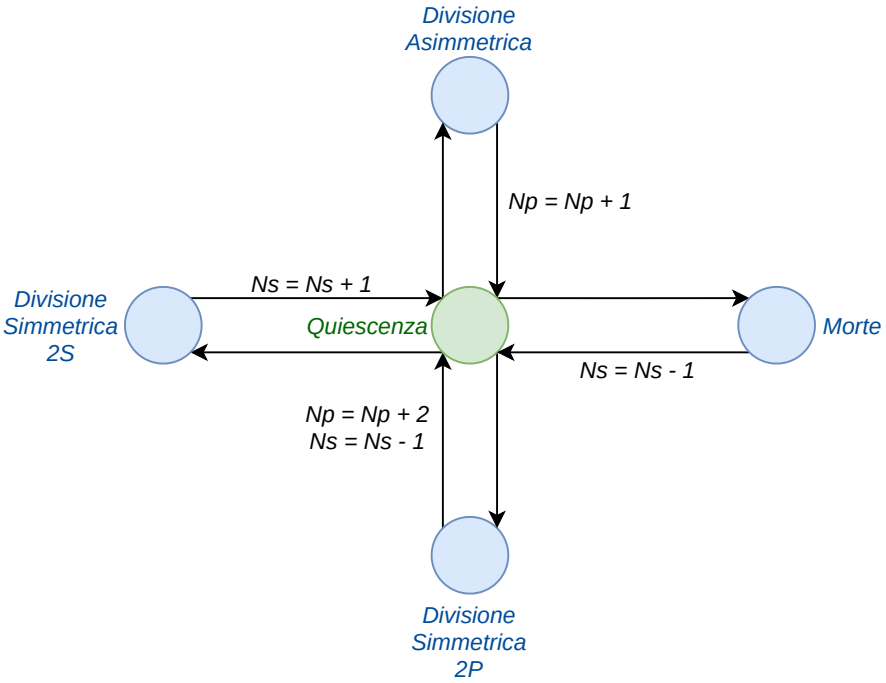


Si ha quindi che l'ambiente non è un aspetto trascurabile dal punto di vista delle simulazioni. Nel ciclo cellulare, a seconda di quali sono gli insiemi di proteine che vanno ad influire sul processo. Le cellule normalmente si trovano in uno stato di **quiescenza**, normalmente indicato con $G_0$, mentre il

**ciclo cellulare** è il processo tramite il quale una cellula si divide. L'ingresso nel ciclo cellulare è normalmente indicato con $G_1$ mentre lo stato di esecuzione del processo con $G_2$, processo dove si ha anche uno stato $S$ (*non viene specificato altro a lezione*). Ci sono poi altri stadi nel ciclo cellulare, tra cui lo stato di **mitosi**, solitamente indicato con $M$, in cui si ha la divisione effettiva, che comporta due cellule che si trovano in partenza nello stato $G_0$. Una cellula in stato $G_0$ può anche *morire in modo programmato*, tramite la cosiddetta **apoptosi**. Tra lo stato $S$ e lo stato $G_2$ si ha che la presenza in ambiente di determinate proteine può provocare le due possibili alternative della *divisione ambientalmente asimmetrica* in fase di *mitosi*, questo in quanto, nel momento in cui il DNA viene duplicato, parti dello stesso vengono *silenziate* mentre altre *attivate* e questa combinazione di eventi comporta la differenziazione delle due cellule figlie.
Per modellare questo sistema usiamo un FSA, che poi potremmo codificare in un qualsiasi linguaggio di programmazione, la semantica resta quella dell'FSA.
Vediamo quindi una versione semplificata di un FSA per questo modello. SI hanno due sole variabili:

1. $Ns$ che è il numero di cellule staminali
2. $Np$ che è il numero di cellule progenitrici

Si ha quindi:



Dove si ha:

- uno stato di *quiescenza* dove si trovano le cellule
- quattro che rappresentano quattro differenti *eventi* possibili, avendo che ogni evento comporta una certa modifica ad una o ad entrambe le variabili:
  - divisione simmetrica di due cellule progenitrici, $2P$
  - divisione simmetrica di due cellule staminali, $2S$
  - divisione asimmetrica
  - morte, nel dettaglio di una cellula staminale (volendo si potrebbe rappresentare anche lo stato di morte per una cellula progenitrice)

In ogni caso quella data è quindi una **semantica**, per quanto semplificata anche troppo (manca in primis una rappresentazione del tempo), di un FSA per rappresentare modelli biologici.
Il **tempo** infatti è normalmente considerato associando un delay esponenziale ad una transizione dell'FSA.
Si possono anche associare alle transizioni differenze probabilità per poter passare ad uno stato partendo da un certo stato, ottenendo in pratica una *Markov chain*, ovvero un **sistema di transizione**.
Vediamo quindi come manipolare questi strumenti modellistici e cosa si ottiene con il loro uso. Quello che viene prodotto è una **traccia**, ovvero una *sequenza ordinata* di vettori di valori, che volendo possono essere *simbolici*.
Ad esempio, per l'FSA appena visto, potremmo avere come traccia, dove ogni volta la coppia è formata dal numero di cellule staminali e dal numero di cellule progenitrici:

$$\langle (100, 10^5), (101, 10^5) \dots (98, 10^5) \rangle$$

La nozione di *traccia* ci permette di ragionare ad un "livello più basso" e ci permette di definire sulla base delle tracce cos'è un **Discrete Event Simulator (*DES*)**. Parlando dei DES si ha che la semantica è più o meno la stessa però l'idea è che studiando come è fatta una *traccia* poi posso costruire un sistema che è di fatto più efficiente. I DES sono molto utili anche per fare simulazioni iniziali di circuiti elettronici.
L'architettura di un simulatore è formata da:

1. una *specifica di sistema* (che può essere, ad esempio, una rappresentazione tramite EDO)

2. un *engine* che, prese in input le specifiche di sistema, produce una *traccia* di un sistema generando una serie di realizzazioni

3. la *traccia di simulazione* che raccoglie tutte le tracce prodotte dall'engine. Si ha una struttura interna

4. una *trace inspection* che manipola la traccia con un certo strumento, ad esempio, costruendo grafici, facendo analisi matematica, producendo una GUI etc...
   In alcuni casi si può anche interagire con la simulazione, interrompendola, ripetendo alcuni step, cambiare parametri "a caldo" etc...

Per implementare un simulatore banalmente si fa un *loop*:

---

**for** $i$ **from** *start* **to** $finish$ **do**
    *evaluate next (i)*

---

dove *evaluate next (i)* determina il tipo di simulazione che si sta facendo.

### 6.2.2 Simulatore di FSA

Per implementare un simulatore di FSA si ha:

- come *specifica* una rappresentazione di un FSA e si hanno quindi abbiamo alcune possibilità:
  - costruire l'FSA e simularlo, tramite una funzione di transizione $\delta$, mettendo in conto che il consumo di memoria cresce rapidamente (anche perché l'eventuale prodotto di un FSA con $N$ stati e di uno con $M$ stati produce in risultato $NM$ stati)
  - tenere l'FSA separato e fare una simulazione più complessa
- come *engine* controllare l'insieme di tutte le transizioni abilitate dallo stato corrente e produrre lo stato successivo

Dal punto di vista "programmativo" si potrebbe usare anche solo una matrice.
La semantica di un DES è quindi quella di un FSA ma si ha una modalità di

esecuzione più efficiente dal punto di vista spaziale a patto di avere una coda di eventi, per questo i DES sono scritti per tenere separati i diversi FSA.
Vediamo quindi come si implementa un DES. Si hanno:

- come *specifiche* si hanno sorgenti, ad esempio unità di computazione, come dei *down samplers* (che riducono il campionamento del segnale) o delle operazioni logiche etc..., che studiano, ad esempio, l'output di generatori di onde quadre
- come *engine* una *coda con priorità* (per controllare quale evento far accadere prima) di coppie $\langle tempo, valore \rangle$, dove il valore può anche essere una struttura dati complessa. La coda contiene gli eventi che devono essere simulati. Si hanno poi i vari *evaluators* delle unità, avendo che ogni unità assume significato sse si hanno tutti i suoi input, potendo così produrre un output

Usando una coda di priorità, implementata tramite un *heap*, abbiamo tempi di inserimento e rimozione logaritmici (con un *heap di Fibonacci* si avrebbe addirittura inserimento in tempo costante).
Un esempio famoso è *Simulink* che è il DES per *MATLAB*.

### 6.2.3 Simulatore di EDO

Un esempio di simulatore che studia equazioni differenziali è il **metodo di Runge-Kutta al quart'ordine**, che appunto è un *integratore* per EDO. È il metodo standard per integrare EDO ed è abbastanza semplice da implementare.
Riprendendo la forma generale di una EDO, o di un sistema di EDO, si ha il cosiddetto **problema del valore iniziale (*initial-value problem*)**, ovvero, avendo $t$ come variabile (che per noi tendenzialmente è il **tempo**) e $c$ come costante:

$$\begin{cases} \frac{\mathrm{d}y(t)}{\mathrm{d}t} = F(y(t), t) \\ y(0) = c \end{cases}$$

Si hanno quindi:

- come *specifica* un insieme di EDO
- come *engine* si calcola in un numero discreto di passi il tempo il valore della funzione e della sua derivata. Come metodi numerici si hanno, ad esempio:
  - il **metodo di Eulero**

- il **metodo di Runge-Kutta**, che può essere a diversi ordini, tipicamente al secondo o al quarto
- altri tipi di integratori, in particolare **metodi impliciti** in grado di trattare *equazioni rigide*, *vincoli algebrici*, etc...

Non si ha quindi tendenzialmente uno studio analitico delle EDO, non essendo in generale fattibile, ma si ha appunto uno studio numerico.

### Il metodo di Eulero

Il **metodo di Eulero** non è più usato in quanto non ha caratteristiche numeriche particolarmente ragionevoli, in primis perché "accumula" errori molto velocemente.
SI usando degli indici per indicare il **passo i-esimo di computazione** e si indica con $h$ il **passo di integrazione**, fisso.
Date queste premesse si supponga di avere l'n-esimo punto della funzione già calcolato, ovvero $y_n$ si ha, per la forma di EDO indicata sopra:

$$y_{n+1} = y_n + h \cdot F(y_n, x_n)$$

In pratica l'idea generale dietro un integratore è quello di calcolare uno step valutando la funzione $F$ allo step precedente e moltiplicando il risultato per il passo di integrazione sommando per lo step precedente, ottenendo in pratica la forma di una retta (tra il punto $x_n$ e il punto $x_{n+h}$).

### Il metodo di Runge-Kutta

Il **metodo di Runge-Kutta al quart'ordine** supera il problema del *metodo di Eulero* di essere troppo lineare, dicendo che per vincolare meglio il computo dei valori successivi della funzione $y$ è bene considerare non solo il punto $x_{n+h}$ ma anche $x_{n+\frac{h}{2}}$. In particolare "costringiamo" la funzione a passare in un certo "corridoio".
Il *metodo di Runge-Kutta al prim'ordine* è in pratica il *metodo di Eulero* e si arriva a quanti ordini si vuole.
La formulazione completa di *Runge-Kutta al quart'ordine* (quart'ordine in quanto si calcolano quattro punti intermedi), detto anche **metodo di Runge-**

**Kutta 4** :

$$k_1 = h \cdot F(x_n, y_n)$$
$$k_2 = h \cdot F\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$$
$$k_3 = h \cdot F\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right)$$
$$k_4 = h \cdot F(x_n + h, y_n + k_3)$$

Dove si nota che ogni $k_i$ è calcolato a partire da $F$ e da voli di $x_n$ e $y_n$ che si conoscono e dai valori di $k_{i-1}$.
Si ha infine la formula per calcolare lo step successivo:

$$y_{n+1} = y_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + \mathcal{O}(h^5)$$

Introducendo quindi un errore è un $\mathcal{O}(h^5)$ (per questo viene anche chiamato **metodo di Runge-Kutta 45**, indicando sia l'ordine, 4, che l'ordine errore, 5), che è un valore molto piccolo, fornendo un ottimo compromesso tra l'errore e la velocità di computazione, ovvero la **velocità di convergenza** del metodo.
A livello computazionale serve quindi la funzione (passandola tendenzialmente come *funzione lambda*), il range, i valori iniziali etc...
Si ha che lo step $h$ può essere fisso o meno, avendo un **algoritmo adattivo**.

### 6.2.4 Sistemi Ibridi

Ci si chiede cosa fare quando si ha una visione del sistema "ad alto livello" e una visione "a basso livello", avendo sistemi complessi che in uno stato si comportano in un modo e in un altro stato in un altro, in uno stato magari si usa un modello e in un altro un modello differente. Si mischiano quindi sistemi con FSA, discreti, e sistemi con EDO, continui, creando dei **sistemi ibridi**. Un sistema ibrido può anche essere un sistema che in due stati presenta due diversi insiemi di EDO, da scegliere in base a certe condizioni, come ad esempio in figura 6.2.

In un sistema ibrido con EDO cambia leggermente la forma del *loop* principale dell'integratore, avendo (con uno step fisso):

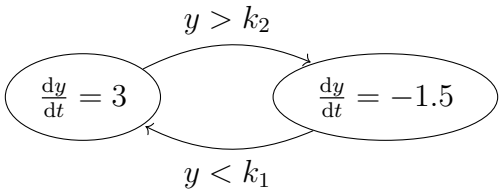



Figura 6.2: Esempio di sistema ibrido con due stati e le corrispondenti EDO, avendo che nel sistema vale $k_1 > k_2$.

---

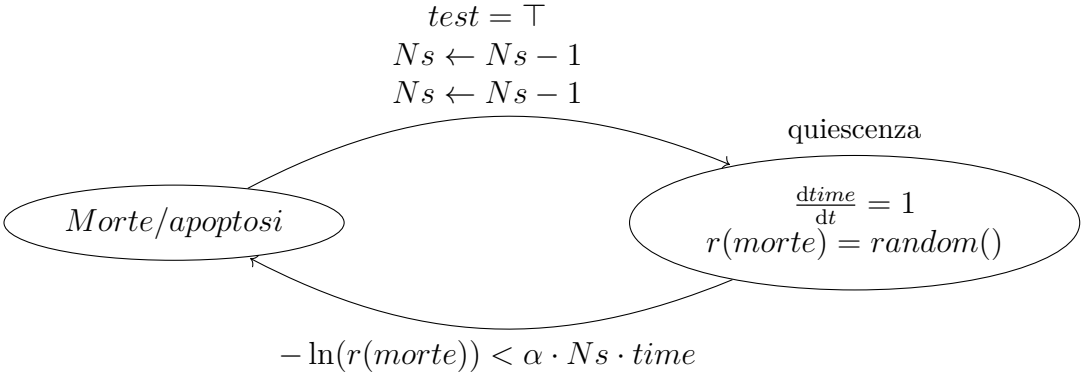
*$h \leftarrow \langle$un certo valore$\rangle$*
**for** *$i$* **from** *start* **to** *$finish$* **do**
    *studia le EDO attuali ad uno step $h$*
    **if** *si abilita una qualsiasi transizione* **then**
        *cambia insieme di EDO*

---

Introducendo anche gli FSA in questo discorso potrei modellare, tramite un *sistema ibrido*, una popolazione di cellule staminali (e il conteggio visto in figura 6.2.1) in modo più realistico, più vicino alla natura discreta e stocastica del modello. Si inserisce quindi un'equazione differenziale che modella il **tempo**, che si evolve in modo costante, integrando:

$$\frac{\mathrm{d}time}{\mathrm{d}t} = 1$$

Si associa quindi ad ogni transizione un ritardo esponenziale che rappresenta il tasso di un evento del mio modello/popolazione (come la *divisione asimmetrica* o l'*apoptosi*).
Si ha quindi il seguente FSA:



Dove ad ogni integrazione in pratica "tiro dei dadi", avendo:

- la transizione etichettata con un certo *test*, nel nostro caso:

  $$-\ln(r(morte)) < \alpha \cdot Ns \cdot time$$

  scatta in base al "lancio dei dadi" fatto con $random()$

- la transizione che torna nello stato di quiescenza merita un approfondimento. Avendo che se sono nello stato $morte/quiescenza$ test è sempre $\top$ e tornando nello stato $quiescenza$ si "riazzera" il tempo e si conteggia la "morte" di una cellula staminale

Ci sono dei problemi con gli algoritmi per la simulazione di *sistemi ibridi*, specialmente *problemi numerici*, ad esempio il problema del **guard crossing**. Per questo problema si ha che, avendo ad esempio un test del tipo $y(t) < 0$, non seguendo la funzione esattamente l'andamento atteso, a livello numerico avendo l'integratore che integra solo in certi punti, si rischia di "perdere" il test, non facendo il cambio di stato.
Si ha quindi un probelma di fedeltà rispetto a problemi che non si conosce bene.

# Capitolo 7

# Simulazionioni Stocastiche

Si introudcono ora le **simulazioni stocastiche**, ovvero "l'altra faccia della medaglia" rispetto alle simulazioni discrete, sempre per reazioni biochimiche. Si vedranno quindi varie rappresentazioni usabili per simulazioni stocastiche e si vedrà un **metodo MonteCarlo**, ovvero il **metodo di Gillespie**, pensato appositamente per studiare reazioni chimiche e biochimiche.
Il problema chiave quando si ragiona sulla simulazione di sistemi è che ci sono differenze di evoluzione degli insiemi di reazioni biochimiche e biologiche che può dipendere da un numero "limitato" di tipi di una certa molecola, ovvero al dalla numerosità di una specie molecolare nel sistema. Dovendo considerare questi effetti non si è più in grado di usare direttamente le EDO, in quanto normalmente le EDO fanno vedere un "comportamento aggregato", ovvero un *comportamento medio*. Il comportamento che si vuole ora studiare invece è un'approssimazione della risultante di molteplici comportamenti individuali. Bisogna quindi capire come descrivere tali sistemi e un modo è quello di usare la **Chemical Master Equation (*CME*)** che è una rappresentazione precisa di questo comportamento, al costo di essere molto "intrattabile" dal punto di vista analitico (essendo quasi impossibile trovare soluzioni chiuse di questa equazione, dovendo procedere a farne una simulazione numerica). L'idea è quella quindi procedere numericamente ma procedendo in modo da essere molto fedeli alla simulazione di una popolazione di individui.
In un modello deterministico, fissate le condizioni iniziali, il comportamento complessivo del sistema è determinato. In un modello stocastico, date le stesse condizioni iniziali, si possono avere comportamenti che sono qualitativamente diversi e sono comportamenti che derivano da termini nel nostro modello che rappresentano *fluttuazioni casuali*. Si procede quindi, ad esempio, introducendo del *rumore* in una EDO, perdendo immediatamente la possibilità di trovare soluzioni analitiche ma ottenendo un comportamento comunque verosimile. L'introduzione di rumore normalmente non comporta

un cambiamento nella traiettoria complessiva del sistema ma a volte può accadere e quello che si vuole garantire è che il modello permetta di riprodurre comportamenti rilevabili anche in un normale esperimento di laboratorio.

## 7.1 Modelli di Markov

Il modo più semplice per rappresentare questo tipo di modelli è comunque quello di usare un **processo di Markov**, che siano **catene di Markov** o **sistemi di transizione**.
Preso un processo di Markov si possono aggiungere ulteriori vincoli sullo scatto di una transizione, associando *condizioni* e/o *azioni* agli archi, in modo tale che un sistema passi da uno statoi all'altro sse una condizione è vera e facendo scattare una certa serie di azioni che modificano lo stato del sistema. Tutto questo fa giustificato dal punto di vista matematico.
Una cosa che si fa con le **catene di Markov** è quello di studiare il comportamento nel lungo periodo, studiando il cosiddetto **steady state (*stato stazionario*)**. Data la matrice di transizione $P$ e il vettore di stato $v$ che rappresenta una distribuzione di probabilità (sommando 1) sugli stati, possiamo definire le varie proprietà (***guardare appunti di Modelli Probabilistici per le Decisioni***).
Si possono anche definire varie estensioni, tra cui:

- Semi-Markov processes
- Generalized Semi-Markov processes
- Reti di Petri stocastiche
- ...

Torniamo alle definizioni di base ricordiamo che un **vettore di probabilità** ha tutte entry non negative che sommano a 1, avendo che ogni entry rappresenta una certa probabilità associata ad un dato stato. Solitamente si parte da un vettore $v_0$ che rappresenta la distribuzione iniziale di probabilità e si ha che $v = v_0 \cdot P$, se si fa un singolo step. Se volessi ottenere la distribuzione dopo $n$ passi (avendo passi discreti) avrei $v_n i = v_0 \cdot P^n$. Se si ha che $v \cdot P = v$ ho che $v$ è un *vettore di steady state*, essendo in uno **stato stazionario**. Lo *stato stazionario* si noti essere un *autovettore* della matrice $P$.

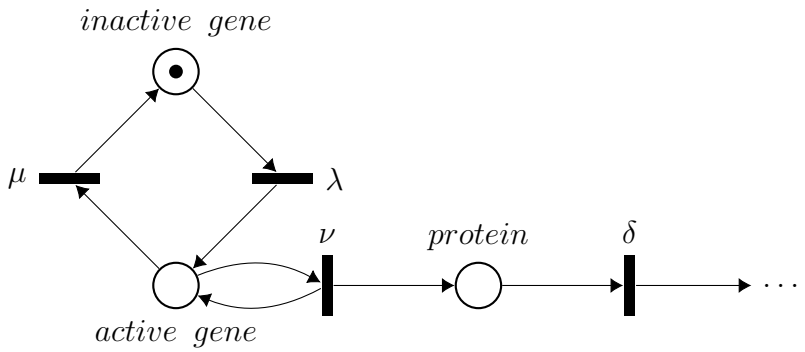



Figura 7.1: Esempio di porzione di rete di petri stocastica

## 7.2 Reti di Petri

Passiamo ora alle **reti di Petri**.
Le *reti di Petri*, specialmente quelle *stocastiche*, sono un interessante formalismo che si può usare per rappresentare le interazioni (di tipo biochimico etc...) tra varie *entità biologiche*. Le *reti di Petri* hanno infatti trovato in primis uso nella rappresentazione di reazioni chimiche.
In figura 7.1 troviamo un semplice esempio dove si rappresenta sostanzialmente il processo di traduzione e trascrizione di una proteina. Se abbiamo un *gene inattivo*, che può diventare un *gene attivo*. Quando attivo il gene può sia produrre una proteina che rimanere attivo, fino a quando non torna ad essere inattivo. Le varie lettere greche associate alle varie transizioni hanno una precisa interpretazione e sono il **tasso di attivazione** di ognuna delle transizioni, ovvero si associa un ritardo, distribuito di fatto in modo esponenziale, ad ogni transizione, aggiungendo un tempo esterno al sistema e ottenendo una **rete di Petri stocastica**. Le *reti di Petri*, intese come *sistemi elementari* o *reti P/T*, sono solo una delle estensioni delle *reti di Petri* utili. Una delle principali limitazioni è quella dell'assenza della rappresentazione del tempo. Tra le estensioni più usate se ne menzionano due, in grado di aggiungere informazioni alle *catene di Markov*, perlomeno dal punto di vista del formalismo grafico:

1. **reti di Petri temporizzate**
2. **reti di Petri stocastiche**, già citate

### 7.2.1 Reti di Petri Temporizzate

Le **reti di Petri stocastiche** permettono di avere l'informazione temporale associata ad ogni singolo elemento della rete:

- *posti*
- *transizioni*, che è il caso più comune
- *archi*
- *marche*
- ...

Vediamo il caso più comune.
Tipicamente ogni transizione $t$ ha un valore di ritardo, ovvero un intervallo di tempo nel quale può attivarsi dal momento in cui diventa abilitata. Tale valore è rappresentato dalla coppia $[d, D]$, dove $d$ è il minimo quantitativo di unità temporali per l'esecuzione mentre $D$ il massimo.
Uno degli utilizzi di tali reti è quello di misurare le prestazioni ma hanno una forte limitazione d'uso in presenza di **conflitti** nella rete.
Quando si hanno diverse transizioni che sono contemporaneamente abilitate si sceglie quella che deve avvenire prima, ovvero quella con la *scadenza più vicina*. In ogni momento ho quindi una priorità su quale transizione far scattare. Potrei comunque avere più transizioni abilitate con la medesima scadenza e questo è un problema, dovendo fare un'ulteriore scelta, magari facendole scattare tutte (per questo il discorso legato ai *conflitti*) oppure scegliendone un sottoinsieme in modo arbitrario.

### 7.2.2 Reti di Petri Stocastiche

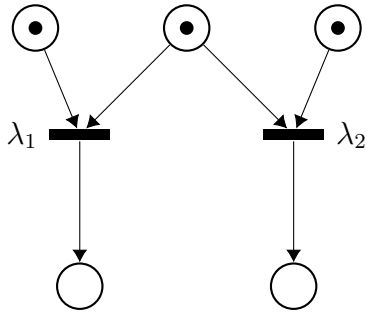
Con le **reti di Petri stocastiche** quello che succede è che si associa ad ogni transizione un *ritardo/delay* normalmente distribuito con un tasso di ritardo costante $\lambda_T$.
Si ha quindi che, indicando con $P(x)$ la probabilità che avvenga un evento $x$, che la probabilità che una certa transizione accada esattamente al tempo $\tau$:

$$P(X_T = \tau) = \lambda_T e^{-\lambda_T \tau}$$

e che la stessa transizione accada prima di un certo tempo:

$$P(X_T \leq \tau) = 1 - \lambda_T e^{-\lambda_T \tau}$$

Avendo un ritardo medio pari a $\frac{1}{\lambda}$.
Il valore di ritardo/delay associato alla transizione $T$ è appunto $X_T$ che è una variabile casuale e la sua funzione di densità di probabilità è distribuita in modo esponenziale sul parametro $\lambda$. Come conseguenza si ha che la probabilità che due transizioni siano abilitate contemporaneamente è estremamente bassa, praticamente nulla. Di fatto quindi possiamo osservare che per costruzione il sistema si comporta come se ci fosse una singola transizione abilitata in ogni istante, risolvendo il problema riscontrato con le *reti di Petri Temporizzate*.
Vediamo un ulteriore semplice esempio.
Si hanno due transizioni, $t_1$ e $t_2$, con associati i rispettivi $\lambda_1$ e $\lambda_2$.




Entrambe le transizioni sono abilitate nello stato iniziale e bisogna capire chi scatta per prima. Inoltre lo scatto di una delle transizioni disabilita l'altra, avendo appunto un *conflitto*. In realtà in questo caso, essendo entrambe le transizioni abilitate, si assegnano alle due transizioni due nuovi tassi:

$$\lambda_1' = \frac{\lambda_1}{\lambda_1 + \lambda_2}$$

$$\lambda_2' = \frac{\lambda_2}{\lambda_1 + \lambda_2}$$

Poi si calcolerà cosa accade dopo lo scatto di una delle due.
Con queste reti si ha un equivalente delle **catene di Markov a tempo continuo**, grazie al fatto che il rateo è esponenziale e che le reti sono *senza memoria*.
(***Parte non capita a fondo***).
Avendo quindi l'approssimazione che si ha una sola transizione abilitata in ogni istante quello che si può dire è che la semantica delle **reti di Petri stocastiche** è la stessa delle **catene di Markov** ma si ha, con le reti, un vantaggio dal punto di vista della mera rappresentazione grafica, avendo una rappresentazione molto concisa rispetto a quella delle catene, permettendo

di dare più informazione in "meno spazio". Con le *reti di Petri stocastiche* si perde però un pezzo di espressività rispetto a quelle standard, in quanto, introducendo i vincoli di rateo, si riduce la rete ad un FSM a cui sono aggiunte le probabilità, anche se le FSM avrebbero meno espressività. Le reti di Petri generano linguaggi che non sono *context-free* ma ci sono linguaggi *context-free* che non possono essere generati dalle reti. In ogni caso l'insieme dei linguaggi generati/riconosciuti contiene propriamente quello relativo alle FSM.
L'analisi quantitativa di tali reti è comunque permessa solo se si ha conoscenza dei ratei di ritardo delle varie transizioni e questo permette l'uso delle stesse per valutare prestazioni.
Ci sono molte estensioni relative alle *reti di Petri stocastiche* e si hanno vari pacchetti standard per le simulazioni.
Vediamo ora il rapporto tra le reti e le applicazioni biologiche[1] in questa tabella riassuntiva delle relazioni tra gli elementi della rete e la biologia:

| **Elemento della rete** | **Corrispettivo biologico** |
|---|---|
| posti | specie molecolari |
| marca | molecola |
| numero di marche | numero di molecole |
| transizioni | reazioni |
| posto di input | reagente |
| posto di output | prodotto |
| funzione di peso | tasso della reazione |
| transizione abilitata | reazione possibile |
| scatto | avvenimento della reazione |

Questo è quindi **un modo** per rappresentare sistemi biologici, tramite *reti di Petri stocastiche*, quindi con la semantica delle *catene di Markov*.

## 7.3 Algoritmi di Gillespie

Passiamo quindi a studiare le *simulazioni stocastiche*, ovvero, data una rappresentazione delle reazioni in un sistema discreto e stocastico, effettuare la simulazione ottenendo risultati ragionevoli, nel dettaglio tramite i cosiddetti **algoritmi di Gillespie**.
Si hanno alcuni aspetti biologici da considerare:

[1] Quantitative Modeling of Stochastic Systems in Molecular Biology using Stochastic Petri Nets, Peter J.E. Goss and Jean Peccoud, PNAS, 95, 6750-6755, June 1988

- molti processi biologici coinvolgono un numero basso di molecole
- trascrizione e traduzione hanno un comportamento stocastico

Spesso quindi si hanno fenomeni davvero molto complessi e quindi si "ripiega" verso una simulazione stocastica e non esatta.
Ci sono molti modelli scaricabili dai già citati database, come *BioModels, Reactome, KEGG* ... e molti di questi modelli sono **pathway**, ovvero, si ricorda, collezioni di reazioni connesse tra loro organizzate in una rete, che spesso è un grafo completamente connesso.
(**Su Moodle due esempi di pathway, quello di $Wnt$, una proteina molto importante, e quello di $TGF$-$\beta$, importante nello studio del *colon***).
Possiamo usare vari modi per costruire tali modello di pathway, che siano *reti di Petri* ma anche linguaggi e sistemi *rules-based*, come *Bionetgen*. In entrambi i casi si ottiene un formalismo per rappresentare un sistema che poi viene simulato tramite algoritmi di Gillespie. Tali algoritmi sono algoritmi di simulazione stocastica della classe degli **algoritmi/metodi Montecarlo** che hanno la peculiarità di essere molto semplici da implementare e sono molto utili permettendo di considerare comportamenti dove gli effetti di alcune, poche, molecole possono influire su tutto il sistema. Questi algoritmi sono giustificati in modo molto rigoroso dal punto di vista fisico (da notare che il primo fu pubblicato nel 1974 su *Journal of computational physics* e anche i successivi articoli sono sempre legati a riviste legate al mondo della fisica, come ad esempio *Journal of Chemical Physics*). Questi studi sono stati poco considerati fino agli anni 2000 e poi sono "esplosi" dal punto di vista delle citazioni.
Vediamo quindi il funzionamento di tali algoritmi.
Si parte dal descrivere lo *stato dinamico del sistema* che è dato da un vettore numerico $\mathbf{X}(t)$, non negativo, dove ogni elemento $X_i(t)$ rappresenta esattamente il numero di molecole che si hanno della specie $S_i$ al tempo $t$:

$$\mathbf{X}(t) = [X_1(t), \ldots, X_N(t)]$$

Questo vettore è direttamente rappresentabile come un vettore di marcatura nelle *reti di Petri*.
L'obiettivo è quindi studiare l'evoluzione di $\mathbf{X}(t)$ a partire da un certo stato iniziale $\mathbf{X}(0) = \mathbf{x}_0$.
Ogni reazione $R_j$ è caratterizzata da due oggetti matematici:

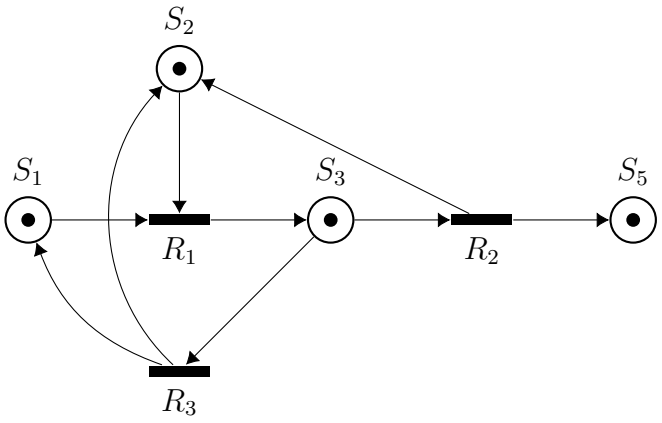
1. il **vettore di cambio di stato (*state change vector*)**:

$$\beta_j = (\beta_{1j}, \ldots, \beta_{nj})$$

che nelle *reti di Petri* è il *vettore di scatto*. Si ha quindi che $\beta_{kj}$ è il cambio di popolazione della specie molecolare $S_k$ dopo un'occorrenza della reazione $R_j$. Quindi se il sistema è nello stato $\mathbf{x}$ e $R_j$ occorre allora lo stato del sistema passa a $\mathbf{x} + B_j$.
La matrice $[\beta_{ij}]$, data da specie/reazione (specie sulle righe e reazioni sulle colonne), è detta **matrice stochiometrica**, che quindi rappresenta i possibili cambiamenti di stato dell'intero sistema date tutte le reazioni

2. la **funzione di propensità** $a_j$. Si ha che $a_j(x)\,\mathrm{d}t$ è la probabilità che, dato $X(t) = \mathbf{x}$, un'occorrenza di $R_j$ accadrà all'interno del **volume** $V$ (dato che si assume che le reazioni avvengano in un certo volume ben definito) in un prossimo intervallo infinitesimale di tempo lungo $\mathrm{d}t$, quindi in $[t, t + \mathrm{d}t]$. Questa funzione quindi dipende dallo stato attuale

**Esempio 2.** *Vediamo un quindi un piccolo esempio:*




*E ci chiediamo quanto vale il cambiamento di stato nel caso scatti $R_2$ (che scatta sse $S_3 > 0$):*

$$\beta_2 = (\beta_{12}, \beta_{22}, \beta_{32}, \beta_{42})$$

*Si ha quindi banalmente che:*

$$\beta_2 = (0, 1, -1, 1)$$

*Avendo che la specie $S_1$ non viene toccato, alle specie $S_2$ e $S_4$ si aggiunge una marca mentre alla specie $S_3$ ne viene tolta una.*

**Esempio 3.** *Vediamo un altro esempio.*
*Si ha una reazione $R_1$ data da:*

$$[X_1] + [X_2] \to 2[X_3]$$

*Si ha quindi:*

$$a_1(\mathbf{x}) = c_1 x_1 x_2$$

*con:*

- $x_1 = \mathbf{x}(1)$
- $x_2 = \mathbf{x}(2)$
- $c_1$ *costante correlata alla reazione quando si fa il trattamento della stessa nel caso deterministico, ovvero quando si considera, ad esempio la legge di massa/azione. Tale costante è quindi legata al rateo costante $k_1$ in questo modo:*

$$c_1 = \frac{k_1}{V}$$

*Si ha comunque un certo insieme di valori per $\beta_1$.*
*Data questa forma di rappresentazione delle reazioni calcolo direttamente sia l'equazione differenziale ma anche la* ***funzione di propensità*** *(che ha una derivazione simile per quanto ci sia una forte differenza concettuale essendo una probabilità).*
La forma della **funzione di propensità** segue direttamente dalla fisica molecolare e dalla teoria cinetica delle reazioni.

### 7.3.1 Chemical Master Equation

Uno dei problemi principali in fase di simulazione è che non si conosce la precisa posizione e la precisa velocità delle varie molecole nel volume $V$ (*si pensi alla teoria molecolare dei gas*). Si procede quindi predicendo la probabilità che una reazione accada nel tempo successivo d$t$ e predicendo che questa reazione sia $R_j$, parlando di *dinamica probabilistica.*
Ogni reazione è caratterizzata dal *vettore di cambio di stato* e dalla *funzione di propensità* e siamo in grado di calcolare la probabilità che $R_j$ accada nel prossimo intervallo di tempo di ampiezza d$t$. Possiamo quindi calcolare la probabilità a partire da uno stato iniziale e ad un tempo iniziale:

$$P(\mathbf{x}, t|\mathbf{x}_0, t_0)$$

che è appunto la probabilità di avere il vettore di stato $\mathbf{x}$ per la popolazione al tempo $t$ dati uno stato di partenza $\mathbf{x}_0$ e un tempo di partenza $t_0$. Questa probabilità, una volta ben scritta, viene chiamata **chemical master equation**.
Per studiare meglio tale equazione consideriamo anche un'altra probabilità:

$$P(\mathbf{x}, t + \mathrm{d}t \,|\mathbf{x}_0, t_0)$$

dove si è aggiunto il "passo temporale" $\mathrm{d}t$. Dobbiamo capire come trovarci allo stato $\mathbf{x}$ al tempo $t + \mathrm{d}t$. Uno dei modi è pensare di trovarsi al tempo $t$ nello stato $\mathbf{x}$ e non avere alcune reazione per un tempo $\mathrm{d}t$, in modo da essere ancora in $\mathbf{x}$ al tempo $t+\mathrm{d}t$. In pratica mi serve la seguente probabilità congiunta:

$$P(\mathbf{x}, t + \mathrm{d}t \,|\mathbf{x}_0, t_0) = P(\mathbf{x}, t|\mathbf{x}_0, t_0) \cdot \left(1 - \sum_{j=1}^{M} a_j(\mathbf{x})\,\mathrm{d}t\right)$$

Dove $1 - \sum_{j=1}^{M} a_j(\mathbf{x})\,\mathrm{d}t$ rappresenta la probabilità che non accada alcuna reazione (essendo uno meno la probabilità che una qualsiasi reazione accada).
Ovviamente questo non è l'unico modo, infatti potrebbe succedere che occorra una reazione $j$ nell'intervallo di tempo lungo $\mathrm{d}t$. Questo però può succedere sse la reazione $j$-esima è accaduta nell'intervallo di tempo $\mathrm{d}t$ e il sistema si trovava nello stato $\mathbf{x} - \beta_j$, ovvero nello stato antecedente all'occorrenza della reazione. In tal caso si ha quindi:

$$P(\mathbf{x}, t + \mathrm{d}t \,|\mathbf{x}_0, t_0) = P(\mathbf{x} - \beta_j, t|\mathbf{x}_0, t_0) \cdot a_j(\mathbf{x} - \beta_j)\,\mathrm{d}t$$

Ovviamente si9 hanno $M$ possibili reazione e quindi bisogna sommare tutti i $j$-esimi termini ottenuti con questa formula (anche se normalmente si ha l'occorrenza di una sola reazione).
Avendo quindi visto i due possibili casi possiamo ottenere la formula finale per la **chemical master equation**.
Definiamo prima la la somma tra la probabilità che nell'intervallo di tempo $\mathrm{d}t$, partendo dal tempo $t$ non avvenga alcuna reazione e quella che avvengano un certo numero $M$ di reazioni:

$$\begin{aligned} P(\mathbf{x}, t + \mathrm{d}t \,|\mathbf{x}_0, t_0) =& P(\mathbf{x}, t|\mathbf{x}_0, t_0) \cdot \left(1 - \sum_{j=1}^{M} a_j(\mathbf{x})\,\mathrm{d}t\right) \\ &+ \sum_{j=1}^{M} (P(\mathbf{x} - \beta_j, t|\mathbf{x}_0, t_0) \cdot a_j(\mathbf{x} - \beta_j)\,\mathrm{d}t) \end{aligned}$$

Pur ricordando, per la seconda sommatoria, che raramente accadono più di una reazione.
Studiamo un po' la formula appena ottenuta. Se sottraiamo da entrambe le parti $P(\mathbf{x}, t|\mathbf{x}_0, t_0)$, dividiamo per d$t$ e ne calcoliamo il limite che tende a 0 otteniamo:

$$\frac{\partial P(\mathbf{x}, t + \mathrm{d}t\,|\mathbf{x}_0, t_0)}{\partial t} = \sum_{j=1}^{M} \left(a_j(\mathbf{x} - \beta_j) \cdot P(\mathbf{x} - \beta_j, t|\mathbf{x}_0, t_0) - a_j(\mathbf{x}) \cdot P(\mathbf{x}, t|\mathbf{x}_0, t_0)\right)$$

Ovvero ottengo la variazione della probabilità nell'intervallo di tempo d$t$, che è la vera e propria **Chemical Master Equation (*CME*)**.
Si noti che la *CME* è un'**equazione differenziale stocastica** che è intrattabile, se non in casi molto semplici, dal punto di vista analitico. Un pathway non rientra nella casistica dei casi semplici, avendo un elevato numero di specie solitamente rappresentate. L'equazione, per quanto intrattabile, ci dice comunque come poter costruire un sistema che ci permetta di simularla, simulando l'evoluzione nel tempo del sistema.
**Gillespie dimostrò che si può derivare la *CME* dal suo algoritmo di simulazione (*forse anche viceversa*).**

### 7.3.2 Implementazione degli Algoritmi di Gillespie

Vediamo quindi come implementare un sistema di simulazione stocastica che ci permetta di seguire l'evoluzione di un insieme di reazioni nel tempo.
Dato che la *CME* è intrattabile generiamo **traiettorie** di $\mathbf{X}(t)$ da studiare. Ovviamente non sono tutte le possibili traiettorie ma se ne generano abbastanza per poter fare, *ex-post*, uno studio statistico su queste traiettorie. Questo non è uguale a fare uno studio numerico della *CME* in quanto si costruisce un algoritmo che ogni volta genera una certa traiettoria, ripetendo più volte la simulazione e osservando infine un insieme di realizzazioni che sono consistenti con la *CME*.
Comunque ci serve un algoritmo numerico e questo è appunto l'**algoritmo di Gillespie**.
Tale algoritmo prevede alcune assunzioni:

- un volume fissato $V$ a temperatura costante
- un numero $N$, $N > 1$, di specie molecolari in una miscela ben mischiata che interagiscono chimicamente nel volume $V$: $S_1, S_2, \ldots, S_N$
- un numero $M$, $M > 1$, di possibili reazioni: $R_1, R_2, \ldots, R_M$
- ci sia equilibrio termico ma non equilibrio chimico (avendo appunto che alcune reazioni possono accadere)

L'assunzione di miscela ben mischiata è un'assunzione essenziale per avere una buona simulazione. Il citoplasma non è una miscela ben mischiata, portando quindi a "prendere con le pinze" le simulazioni.
Per vedere come funziona l'algoritmo, di per sé molto banale, bisogna capire la matematica che c'è dietro.
IL primo punto chiave è la seguente probabilità, dato $\mathbf{X}(t) = \mathbf{x}$, con $j$ indice della prossima reazione e $\tau$ tempo della prossima reazione:

$$P(\tau, j|\mathbf{x}, t)$$

ovvero la probabilità che la reazione $j$-esima, ovvero $R_j$, accada nell'intervallo di tempo infinitesimale $[t+\tau, t+\tau+\mathrm{d}t]$. Questa è in pratica la probabilità che voglio calcolare e si nota che non è la stessa della *CME* ma è la **la funzione di densità di probabilità congiunta condizionata di due variabili casuali $j$ e $\tau$.**
Considero ora la probabilità:

$$P_0(\tau|\mathbf{x}, t)$$

ovvero la probabilità che non accada alcuna reazione nell'intervallo $[t, t+\tau]$, dato lo stato $\mathbf{X}(t) = \mathbf{x}$ (si nota come si stia ragionando come nel caso della *CME*). Ne segue, facendo derivazioni simile a quelle fatte per la *CME*, che:

$$P(\tau, j|\mathbf{x}, t)\,\mathrm{d}t = P_0(\tau|\mathbf{x}, t) \cdot a_j(\mathbf{x})\,\mathrm{d}t$$

Si ha anche qui la probabilità accada una qualche reazione, avendo in tal caso:

$$P_0(\tau + \mathrm{d}t\,|\mathbf{x}, t) = P_0(\tau|\mathbf{x}, t) \cdot \left(1 - sum_{j=1}^{M} a_j(\mathbf{x})\,\mathrm{d}\tau\right)$$

Prendendo infine il limite per $\mathrm{d}\tau$ che tende a 0 e risolvendo l'equazione differenziale per ottenere una soluzione analitica, si ottiene:

$$P_0(\tau|\mathbf{x}, t) = e^{-a_0(\mathbf{x})\tau}$$
$$a_0(\mathbf{x}) \equiv \sum_{j=1}^{M} a_j(\mathbf{x})$$

Ma a questo punto, ricordando che:

$$P(\tau, j|\mathbf{x}, t)\,\mathrm{d}t = P_0(\tau|\mathbf{x}, t) \cdot a_j(\mathbf{x})\,\mathrm{d}t$$

si ottiene che:

$$P(\tau, j|\mathbf{x}, t) = a_j(\mathbf{x}) \cdot e^{-a_0(\mathbf{x})\tau}$$

Data quest'ultima equazione si può andare a calcolare effettivamente il $\tau$ e il $j$ che serve semplicemente usando una generazione di numeri pseudo-casuali. Si usa una distribuzione uniforme per generare due numeri, $r_1$ e $r_2$ in $[0, 1]$ e a quel punto si ha che:

$$\tau = \frac{1}{a_0(\mathbf{x})} \ln\left(\frac{1}{r_1}\right)$$

mentre $j$ è il più piccolo intero tale che sia soddisfatta la seguente condizione:

$$\left(\sum_{i=1}^{j} a_i(\mathbf{x})\right) > r_2 a_0(\mathbf{x})$$

Ho quindi stabilito in modo causale quando accadrà la prossima reazione e quale sarà questa reazione.
Possiamo quindi definire i vari passi dell'**algoritmo di Gillespie**:

1. si inizializzano $t = t_0$ e $\mathbf{x} = \mathbf{x}_0$

2. a partire dallo stato $\mathbf{x}$ al tempo $t$ si calcolano tutte le $a_j(\mathbf{x})$ (quindi per ogni reazione) e quindi posso calcolare $a_0(\mathbf{x})$. Questo è un passaggio costoso dal punto di vista computazionale

3. si generano pseudo-casualmente $\tau$ e $j$, come visto precedentemente (e la generazione di $r_1$ e $r_2$ è la cosa computazionalmente più costosa qui)

4. si aggiorna $\mathbf{x}$ a $\mathbf{x} + \beta_j$ (simulando l'occorrenza della reazione) e si aggiorna $t$ a $t + \tau$

5. si salva la coppia $(\mathbf{x}, t)$ e si torna al passaggio 2. Alternativamente è possibile concludere la simulazione (in base al fatto che si ha computato per troppo tempo o che magari il sistema non cambia da parecchie iterazioni)

Il ciclo centrale dell'algoritmo è quindi molto semplice *(nel secondo paper di Gillespie c'è il codice in* Fortran), al più di sapere come funzioni la matematica sottostante e conoscere la rappresentazione interna delle reazioni, dei cambiamenti di stato etc...
L'*algoritmo di Gillespie* è definibile **esatto** in quanto le sue realizzazioni sono in accordo con la *CME*. Si osserva inoltre che le proprietà statistiche di un insieme di traiettorie generate dall'algoritmo, in linea di principio, danno un'informazione accurata circa il comportamento stocastico globale di un sistema dinamico come previsto dalla *CME*.

L'*algoritmo di Gillespie* è computazionalmente costoso, dal punto di vista temporale più che spaziale. Potremmo avere un sistema con reazioni molto probabili su cui "oscilla" il sistema per molto tempo, prima che la generazione pseudo-casuale renda possibile un'altra reazione. Un altro problema è che $a_0(\mathbf{x})$ può essere molto grande in presenza di un gran numero di molecole della stessa specie e questo rende $\tau$ (essendo $a_0(\mathbf{x})$ a denominatore nella formula) molto piccolo, rallentando molto la simulazione. Quest'ultimo aspetto si contrappone a quanto accade in un simulatore deterministico dove, ad esempio, si ha un passo di integrazione fisso su cui si ha un certo controllo e che garantisce che la simulazione avanza in modo controllato mentre qui non si ha questo aspetto.
L'*algoritmo di Gillespie* è quindi molto semplice da implementare anche se molto costoso dal punto di vista delle risorse. Si ha inoltre che avendo molte reazioni e molte specie hanno conseguenze sul ciclo e, ad esempio, se una reazione coinvolge pochissime molecole questa viene "ignorata" a favore di reazioni che coinvolgono molte molecole. Bisogna quindi fare molte simulazioni (cambiando il *seed* del generatore pseudo-causale) per capire come funzioni verosimilmente il sistema e caratterizzarlo al meglio. Questa cosa comporta anche molta interazione "manuale" nonché ulteriori costi computazionali.
L'*algoritmo di Gillespie* è molto usato in biologia computazionale, ad esempio in simulatori come *COPASI, StochSim etc...*

### 7.3.3 Variante Tau-Leaping

Abbiamo visto come ci possano essere problemi di prestazioni con l'*algoritmo di Gillespie*. Non ha caso si hanno vari studi in merito al miglioramento delle prestazioni dello stesso in certe condizioni.
Una delle prime varianti è quella detta **tau-leaping** che si basa sull'idea che, trovandosi in un certo stato, si procede vedendo cosa succede se ci si mette a fare un salto in avanti nel tempo di una quantità $\tau$ predefinita. Quindi $\tau$ è passato come parametro.
Si ha quindi l'algoritmo:

1. si avanza della quantità di tempo pre-stabilita $\tau$ (*non capisco se questo step è generale perché sembra che si avanzi due volte, avanzando anche nel punto 3*)

2. si calcola $k_j$, ovvero il numero di volte che la reazione $R_j$ occorre nella quantità di tempo $\tau$

3. si avanza il tempo di $\tau$ e si aggiorna lo stato del sistema $\mathbf{x}$ tramite:

$$\mathbf{x} + k_1\beta_1 + \cdots + k_M\beta_M$$

4. si trova un $\tau$ sufficientemente piccolo per cui la *funzione di propensità* rimane costante, avendo la cosiddetta **leap condition**, e tale che i vari $k_j$ siano grandi, massimizzandole. Questa operazione non è affatto banale e si hanno vari articoli in merito