

Data and Computational Biology

UniShare

Davide Cozzi
@dlcgold

Indice

1	Introduzione	4
2	Introduzione alla Biologia Computazionale	5
2.1	Accenni di biologia molecolare	5
2.1.1	DNA ed RNA	5
2.1.2	Esoni, Intronni e Splicing alternativo	7
3	Esempio del Repressilator	15
3.1	Il Modello Biologico	15
3.2	Il Modello Matematico	17
4	Studio di Sistemi Biologici	21
4.1	Microarrays	22
4.2	Next Generation Sequencing	26
4.2.1	Dal Sequenziamento alle Analisi	28
4.3	Single-Cell Analysis	28
4.4	Risorse Online	29
5	Introduzione ai Prerequisiti	31
5.1	Biochimica	32
5.1.1	Biochimica e Metabolismo	34
5.2	Modellazione Matematica	38
5.2.1	Legge di Azione di Massa	39
5.2.2	Equazioni di Michaelis-Menten e Hill	40
6	Simulazioni Deterministiche e Ibride	47
6.1	Equazioni Differenziali Ordinarie	47
6.2	Modelli Discreti	48
6.2.1	Modellazione di Cellule Staminali	49
6.2.2	Simulatore di FSA	53
6.2.3	Simulatore di EDO	54

6.2.4	Sistemi Ibridi	56
7	Simulazioni Stocastiche	59
7.1	Modelli di Markov	60
7.2	Reti di Petri	61
7.2.1	Reti di Petri Temporizzate	62
7.2.2	Reti di Petri Stocastiche	62
7.3	Algoritmi di Gillespie	64
7.3.1	Chemical Master Equation	67
7.3.2	Implementazione degli Algoritmi di Gillespie	69
7.3.3	Variante Tau-Leaping	72
8	Simulazioni Spaziali	74
8.1	Cripte Coloniche	74
8.2	Evoluzione Tumorale	76
8.3	Simulazioni 3D	78
8.3.1	Modelli In-Lattice	79
8.3.2	Modelli Lattice-Free	83
8.3.3	Obiettivi Futuri	89
9	Flux Balance Analysis	91
9.1	Pathway Metabolici	91
9.2	Programmazione Lineare	95
9.3	Questioni Avanzate per FBA	100
9.3.1	Esplorazione dei Flussi	101
9.3.2	The Enhanced Growth Model	103
9.3.3	Ricerca di Valori Sub-Ottimali	108
9.3.4	Metabolic Rewiring	110
10	Progressione Tumorale	112
10.1	Approfondimento Biologico	112
10.2	Studio della Progressione	116
10.2.1	Filogenesi	117
10.3	Cross Sectional Data	120
10.3.1	Tipologie di Studio	121
10.3.2	Algoritmo CAPRI	125
10.3.3	Usi reali di TRONCO e CAPRI	132
10.3.4	Analisi Cancro al Colon Via PiCnIc	134
10.4	Individual Data	139
10.4.1	Tipologie di Alberi	143
10.4.2	TRaIT	146

10.5 Teoria della Filogenesi	151
10.5.1 Algoritmi di Filogenesi	152
10.5.2 Software per Filogenesi	156
10.6 Problemi Aperti	156
10.7 Analisi Longitudinale	157
10.7.1 LACE	159
10.7.2 VERSO	166
10.7.3 Problemi Aperti	168
11 Single-Cell Data Preprocessing	169
11.1 Single-Cell Sequencing	169
11.1.1 KNN-Smoothing	170
11.1.2 Deep Count Autoencoder	171
11.1.3 Batch Effect	172
11.1.4 Copy Number Alterations	174
11.2 Pipeline	178
11.2.1 Quality Check	179
11.2.2 Data Integration & Clustering	180
11.2.3 Differential Expression Analysis	182
12 Control Theory in Computational Biology	184

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 *appaiaiati* 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 $S = \text{"TAGGCCATATGAC"}$ 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:

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

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**, simile 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 aminoacidi 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	$\sim 100\text{s}$
trascrizione di un gene	$\sim 1\text{m} (80 \frac{\text{bp}}{\text{s}})$	$\sim 100\text{s}$
generazione di una cellula	da 30m a ore	da 20h a statico
transizione di stato proteico	da $1\mu\text{s}$ a $100\mu\text{s}$	da $1\mu\text{s}$ a $100\mu\text{s}$
rate di mutazione	$\sim \frac{10^{-9}}{\frac{\text{bp}}{\text{generazione}}}$	$\sim \frac{10^{-8}}{\frac{\text{bp}}{\text{anno}}}$

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. Leibler¹, 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

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

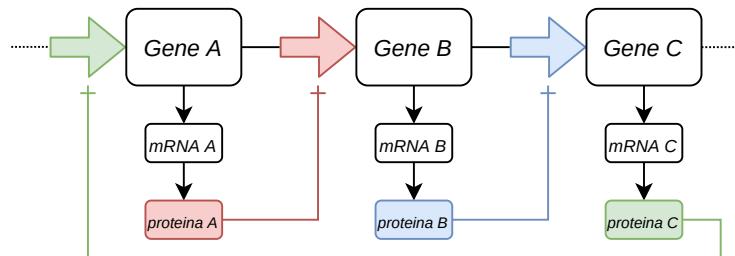


Figura 3.1: Schema di base del Repressilator, con le frecce bidimensionali che rappresentano l'azione di inibizione delle proteine.

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

- *TetR* prodotta dal gene *tetR*
 - *AcI* prodotta dal gene *λcI*
 - *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 sì 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 sì 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 coltura di batteri.

Ricordiamo che la fluorescenza è in realtà abbastanza comune in natura. Si ha un ulteriore “trucco”. Se si lascia una coltura 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 β-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**):

- α_0 , numero di copie di proteine per cellula prodotte da un certo promotore in presenza del represso
- α , numero di copie di proteine per cellula prodotte da un certo promotore in assenza del represso (sarebbe $\alpha + \alpha_0$)

- β , 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:

$$\frac{dm_i}{dt} = -m_i + \frac{\alpha}{1 + p_j^n} + \alpha_0$$

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:

$$\frac{dp_i}{dt} = \beta(m_i - p_i)$$

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>i</i>	<i>lacI</i>	<i>tetR</i>	<i>λcI</i>
<i>j</i>	<i>λcI</i>	<i>lacI</i>	<i>tetR</i>

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 τ_{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 coltura 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, sia 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]
```

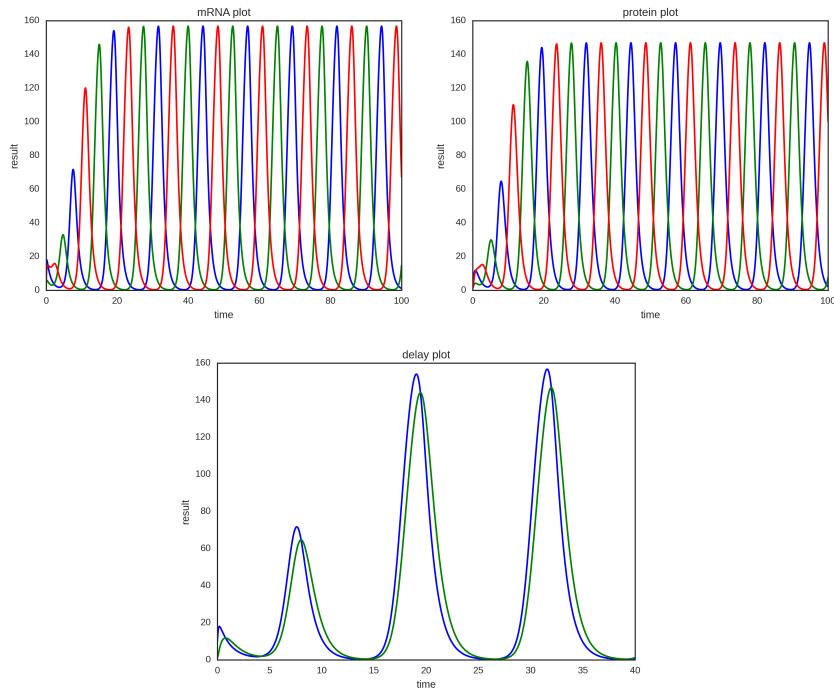


Figura 3.2: Grafici relativi al modello del Repressilator ottenuti tramite Python e Matplotlib, con $n = 2$, $\alpha_0 = 0.005$, $\alpha = 220$ e $\beta = 2$. In primis, a destra quella di mRNA mentre a sinistra la quantità di repressore/proteina rispetto al tempo. In basso le quantità di mRNA (nel caso 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 esseri viventi sono di topologia isomorfa a dei tubi.

Tornando al discorso delle cellule staminali si ha che esse 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 che 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 colpiti 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 condizioni 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 una *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*,

estratto 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 microarray in questa soluzione, procedendo poi alla scansione con il laser

2. nella griglia si ottiene quindi che del materiale genetico delle cellule nella prima condizione si attaccano ad alcune sonde mentre quella delle seconde condizioni 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 unire 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 sistematizzazione 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 sì 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 GO¹ 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 entità 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*)

¹www.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 trascrittori.

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 transcriptomic**, 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 appuntotecniche 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 micròbiche 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 Information*), 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

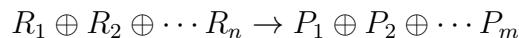
La **materia** è studiata in varie forme, tra cui, in **biochimica**, quella di *miscele*. Le miscele possono essere:

- **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:



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 dagl'organizzazioni 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

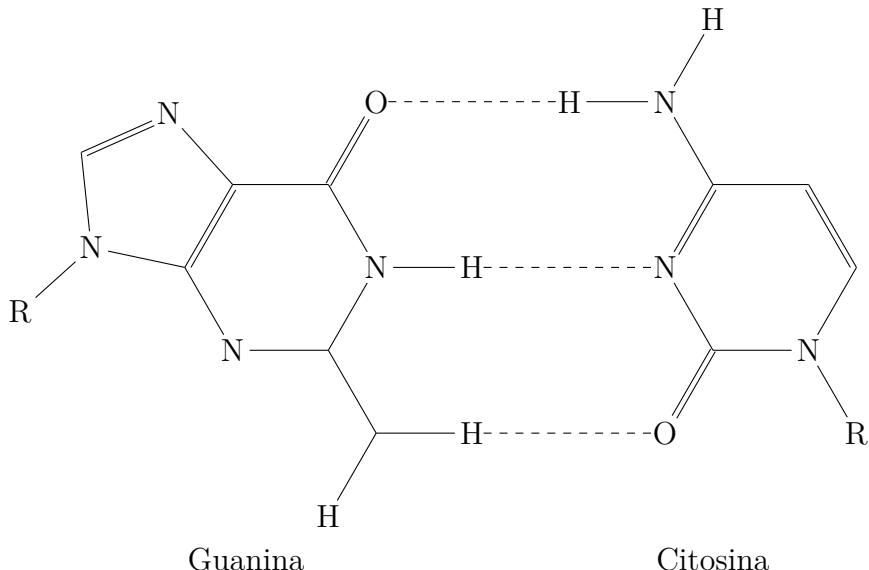


Figura 5.1: I tre legami a idrogeno tra Guanina e Citosina

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**.

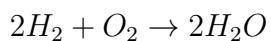
Ogni atomo di carbonio forma 4 legami con altri atomi, l’Azoto 3, l’ossigeno 2 e l’Idrogeno 1. Nelle figure delle molecole dove non si ha nulla indicato si ha un Carbonio.

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:



*Il rapporto quantitativo tra i composti in una reazione è chiamato **stechiometria della reazione** e l’**energia** è la chiave di questi ragionamenti .*

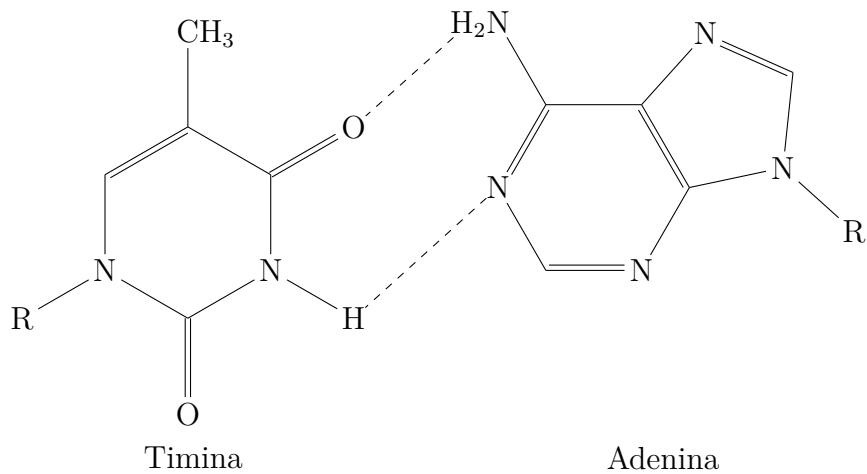


Figura 5.2: I due legami a idrogeno tra Timina e Adenina

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 avvenire 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 ΔH , 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 ΔG 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 reazioni 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. **catabolismo**, 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 *Gene Ontology*.

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 reazioni 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:



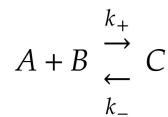
Il reaction rate k è dovuto a vari aspetti:

- 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:

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

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



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

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

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:

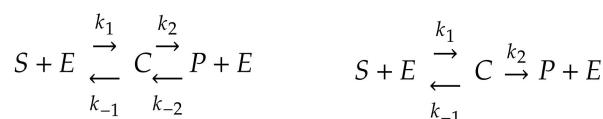
$$\frac{k_-}{k_+} = k_{eq} = \frac{[A]_{eq}[B]_{eq}}{[C]_{eq}}$$

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 Menten**.

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):



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 $A + B \xrightarrow{k} C$, e la sua versione differenziale $\frac{d[C]}{dt} = k[A][B]$ 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:

$$\begin{aligned}\frac{d[S]}{dt} &= k_{-1}[C] - k_1[S][E] \\ \frac{d[E]}{dt} &= (k_{-1} + k_2)[C] - k_1[S][E] \\ \frac{d[C]}{dt} &= k_1[S][E] - (k_2 + k_{-1})[C] \\ \frac{d[P]}{dt} &= k_2[C]\end{aligned}$$

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:

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

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

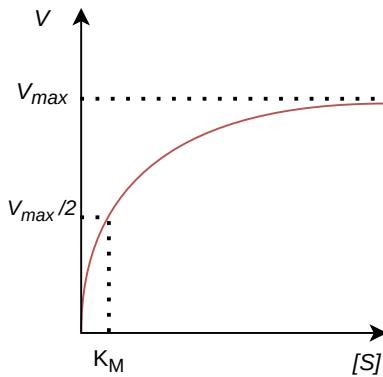


Figura 5.3: Grafico approssimativo rappresentante l'andamento della velocità di reazione.

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.

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 quanità 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:

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

Sapendo poi che a V_{max} si ha 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:

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

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:

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

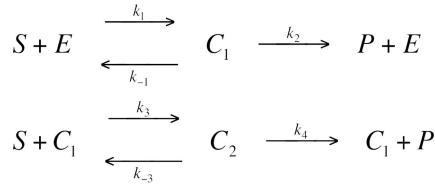
Si arriva quindi ad una definizione.

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 le 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 :



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, \dots, K_l che sono le l costanti 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:

$$\begin{aligned}\dot{y}(t) &= F(y(t), t) \\ y(t_0) &= k\end{aligned}$$

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'equazione 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:

$$\begin{aligned}\frac{dm_i}{dt} &= -m_i + \frac{\alpha}{1 + p_j^n} + \alpha_0 \\ \frac{dp_i}{dt} &= -\beta(p_i - m_i)\end{aligned}$$

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 espansive 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**.

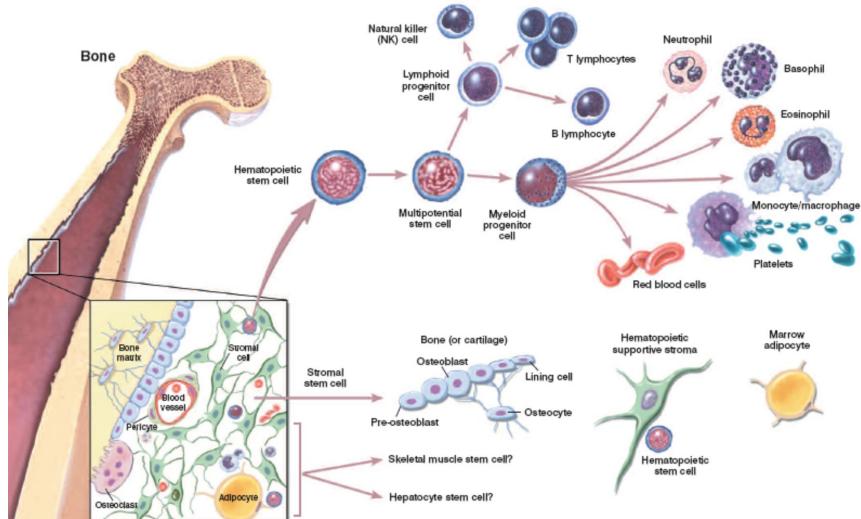


Figura 6.1: Schema completo del sistema ematopoietico con le varie suddivisioni delle cellule.

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
- ...

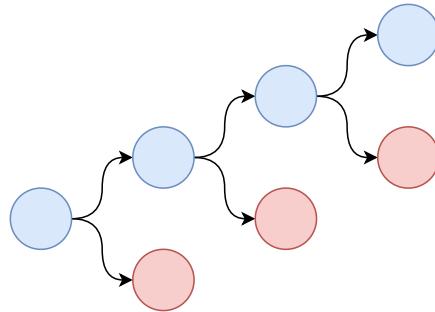
Con **sistema ematopoietico** si intende il sistema che da luogo alle cellule che compongono il nostro sangue. Le cellule ematopoietiche derivano dalle **cellule stromali** contenute nel *midollo osseo*.

Modellazione del sistema in figura 6.1 ¹

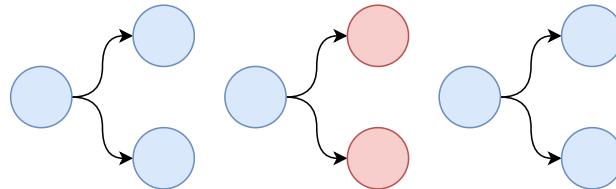
¹https://stemcells.nih.gov/info/Regenerative_Medicine/2006Chapter2.html

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

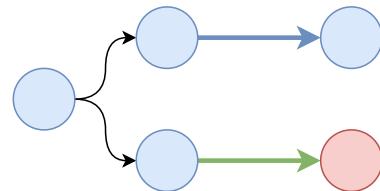
1. **divisione asimmetrica**, che è quello che si nota quando da una cellula staminale si produce una cellula differenziata:



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

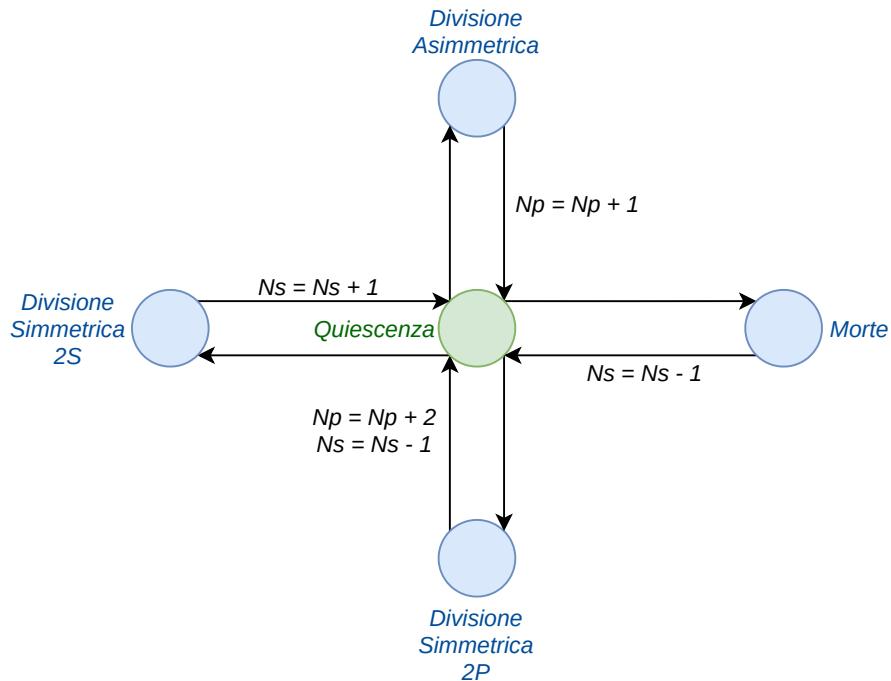
ciclo cellulare è il processo tramite il quale una cellula si divide. L'ingresso nel ciclo cellulare è normalmente indicato con G_1 mentre lo stato di esecuzione del processo con G_2 , processo dove si ha anche uno stato S (*non viene specificato altro a lezione*). Ci sono poi altri stadi nel ciclo cellulare, 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. N_s che è il numero di cellule staminali
2. N_p 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 δ , 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 “programmatico” 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 $\langle \text{tempo}, \text{valore} \rangle$, 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:

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

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 :

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

Dove si nota che ogni k_i è calcolato a partire da F e da valori 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):

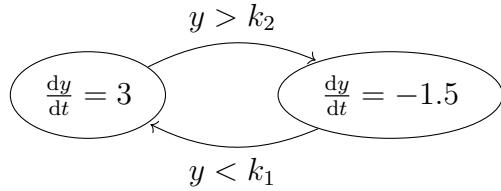


Figura 6.2: Esempio di sistema ibrido con due stati e le corrispondenti EDO, avendo che nel sistema vale $k_1 > k_2$.

```

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

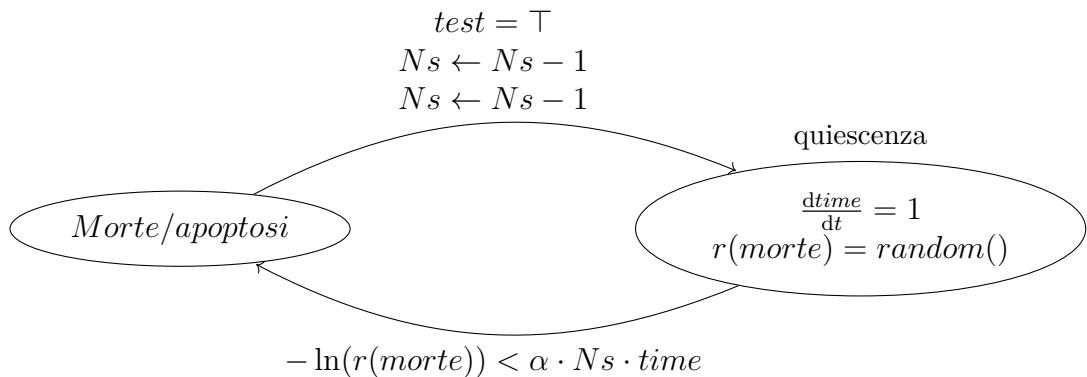
```

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:

$$\frac{\text{dtime}}{\text{dt}} = 1$$

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:

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

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

Simulazioni Stocastiche

Si introducono 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 stato all'altro se 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 $v_n = v_0 \cdot P^n$. 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 .

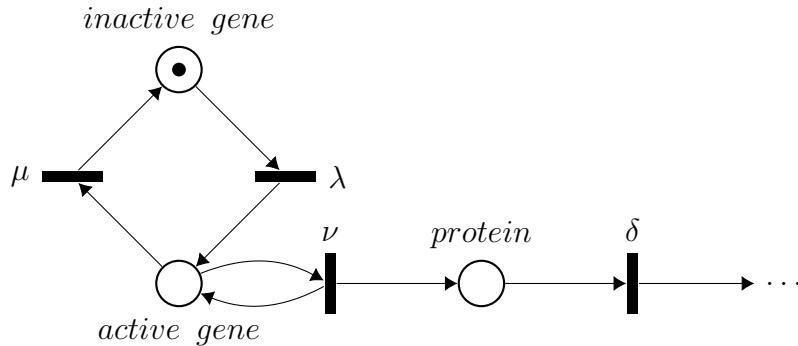


Figura 7.1: Esempio di porzione di rete di petri stocastica

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 λ_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 τ :

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

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

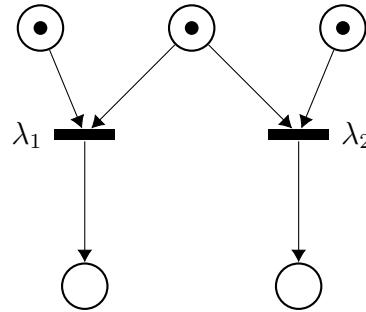
$$P(X_T \leq \tau) = 1 - \lambda_T e^{-\lambda_T \tau}$$

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 λ . 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 λ_1 e λ_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 biologiche¹ 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:

¹Quantitative 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-β*, 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 :

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

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*):

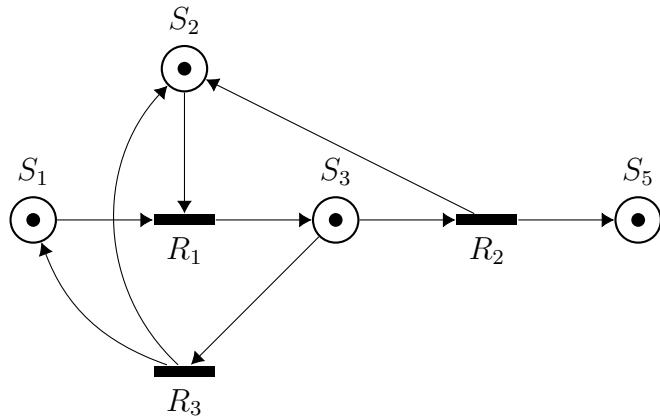
$$\beta_j = (\beta_{1j}, \dots, \beta_{nj})$$

che nelle *reti di Petri* è il *vettore di scatto*. Si ha quindi che β_{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 stoichiometrica**, 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 $a_j(x) dt$ 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 dt , quindi in $[t, t + dt]$. Questa funzione quindi dipende dallo stato attuale

Esempio 2. Vediamo quindi un piccolo esempio:



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

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

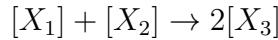
Si ha quindi banalmente che:

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

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:



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 β_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 dt 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 dt . 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:

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

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

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

Dove $1 - \sum_{j=1}^M a_j(\mathbf{x}) dt$ 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 dt . Questo però può succedere sse la reazione j -esima è accaduta nell'intervallo di tempo dt 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:

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

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 dt , partendo dal tempo t non avvenga alcuna reazione e quella che avvengano un certo numero M di reazioni:

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

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 dt e ne calcoliamo il limite che tende a 0 otteniamo:

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

Ovvero ottengo la variazione della probabilità nell'intervallo di tempo dt , 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, \dots, S_N
- un numero M , $M > 1$, di possibili reazioni: R_1, R_2, \dots, 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 τ 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 $[t + \tau, t + \tau + dt]$. 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 τ** .

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:

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

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

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

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

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

Ma a questo punto, ricordando che:

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

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 τ 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 τ 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 funziona 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 τ (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 τ predefinita. Quindi τ è passato come parametro.

Si ha quindi l’algoritmo:

1. si avanza della quantità di tempo pre-stabilita τ (*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 τ

3. si avanza il tempo di τ 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 τ 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

Capitolo 8

Simulazioni Spaziali

Si studiano ora simulazioni più complesse, che hanno a che vedere con la geometria della strutture biologiche (dei tessuti, dell'interno delle cellule etc...) che si vogliono simulare. Si è quindi partiti da simulazioni discrete, simulazioni continue, simulazioni stocastiche per proseguire ora con simulazioni di strutture tridimensionali. Precedentemente si erano studiati modelli dove lo spazio non aveva un vero e proprio ruolo.

8.1 Cripte Coloniche

Per introdurre al meglio le simulazioni spaziali dobbiamo prima tornare a parlare delle **cripte coloniche**.

Il **cancro al colon** è una delle principali cause delle morti di tumore ed è un cancro molto ben studiato con una fenomenologia e una progressione più uniformi dal punto di vista istologico. Si hanno anche alcuni insiemi di mutazioni già identificati.

Il cancro al colon si origina dalle *cripte coloniche* dove si ha uno strato di cellule staminali che si suddividono per riprodurre l'*epitelio* dell'intestino. Normalmente il “fondo” della cripta, dove troviamo lo strato di cellule staminali epiteliali, lo indichiamo come *base* mentre la “cima/uscita”, ovvero la parte che da sull'intestino, *lumen*. Le cellule staminali continuano a riprodursi e spingono verso il *lumen* le cellule che si trovano “sopra” (tra virgolette in quanto l'intestino è comunque avvolto su se stesso oltre ad essere effettivamente un “tubo”) di loro. Mentre si muovono verso il *lumen* queste cellule, che sono cellule che sono il risultato della divisione delle cellule staminali, si suddividono ulteriormente e si differenziano e continuano a spingersi a vicenda verso il *lumen*. Continuano a differenziarsi fino a che non diventano **completamente differenziate** ed escono dal *lumen*, trovandosi sulla super-

ficie dell'intestino (in realtà risalgono i **villi intestinali**).

Le cellule staminali si dividono in diversi modi, partendo con cellule un po' più differenziate, all'incirca a metà della cripta, fino ad essere completamente differenziate nei pressi del *lumen*. Quando arrivano alla cima del villo intestinale vengono poi rimosse tramite azioni meccaniche.

Su slide, lezione 6, varie immagini delle cripte coloniche.

Il meccanismo di suddivisione è molto interessante e nel suo proseguire produce quattro tipi di cellule:

1. **absorptive cell (*cellule assorbenti*)**
2. **goblet cell (*cellule caliciformi*)**
3. **enteroendocrine cell (*cellule enteroendocrine*)**
4. **Paneth cell**

Ognuna di queste cellule ha uno scopo ben preciso anche se ancora non si è ben definito quale per le *Paneth cell*, ovviamente le funzioni sono legate all'**apparato digerente** (ad esempio le *absorptive cell* si occupano di assorbire i nutrienti dalle pareti dei villi). Modellare in 3D questa struttura non è affatto semplice, soprattutto se si vuole coniugare la dinamica interna di ogni cellula con la dinamica intercellulare, con le interazioni tra le varie cellule. Un altro aspetto sono le tempistiche dell'intero processo, avendo che la differenziazione e la migrazione verso la cima del villo avviene tra le 24 e le 48 ore, mentre prima la migrazione verso il *lumen* era avvenuta in circa 24/36 ore. Si ha quindi un continuo riciclo della "fodera" dell'intestino e questo è uno dei motivi per cui è semplice che si sviluppi un tumore all'interno del colon.

Quando si osserva come le cellule si suddividono e differenziano possiamo costruire uno schema/modello di come si suddividono queste cellule:

- le cellule staminali si suddividono e restano cellule staminali
- le cellule staminali si suddividono e diventano dei *progenitori proliferativi*, che non sono ancora completamente differenziati. A questo punto:
 - il progenitore proliferativo si suddivide in un altro progenitore proliferativo
 - il progenitore proliferativo si suddivide in una *Paneth cell*. Da cui in poi, oltre ad avere ancora suddivisione in *Paneth cell* abbiano due ulteriori "percorsi":

- * le cellule si differenziano fino a diventare *absorptive cell*
- * le cellule si differenziano fino ad ottenere due sotto-popolazioni, una per le *goblet cell* e una per le *enteroendocrine cell*

Siamo in una situazione analoga a quella del **sistema ematopoietico**, con un “modello ad albero”.

Interessante, dal punto di vista della simulazione 3D di questo tipo di modelli, è che è possibile, *in silico*, tenere traccia della progenie di tutte le cellule che si ottiene durante i vari step di divisione. Questo è interessante perché i biologi sono riusciti poi a trovare dei metodi per tracciare questa evoluzione, tramite dei *marker biochimici*, per capire da dove si origina una cellula differenziata.

Una delle domande che ci si pone è quella di capire cosa regola questo meccanismo. Chiaramente c'è un influsso di nutrienti che fa sì che le cellule aumentino di volume e quindi successivamente si dividano ma si hanno anche altri “segnali” che sostanzialmente dicono alle cellule che sono in loro prossimità quale sia la distanza rispetto al *lumen* della cripta. Si ha quindi un “gradiente lineare” di una serie di proteine e altre molecole, nel complesso vengono chiamati **morfogeni**, che è molto elevato nel fondo della cripta e molto basso nei pressi del *lumen*. Si ha quindi una correlazione tra la quantità di *morfogeni* e la velocità di divisione e differenziazione delle cellule. Il primo articolo che parlò di questa cosa usò nei grafici i colori blu, bianco e rosso e da quel momento il sistema venne chiamato **sistema della bandiera francese**. Si noti che la progenie, dal punto di vista schematico, tende a crescere in “colonne” sulla superficie della cripta, avendo una serie di meccanismi che fanno sì che la crescita sia **ordinata**.

8.2 Evoluzione Tumorale

Vediamo anche, prima ancora di passare alle vere e proprie simulazioni spaziali, come si evolve un tumore. Vediamo quindi brevemente come funziona la **progressione del cancro al colon**, descritta per la prima volta ad inizio anni novanta. Il tutto può essere riassunto con un semplice schema:

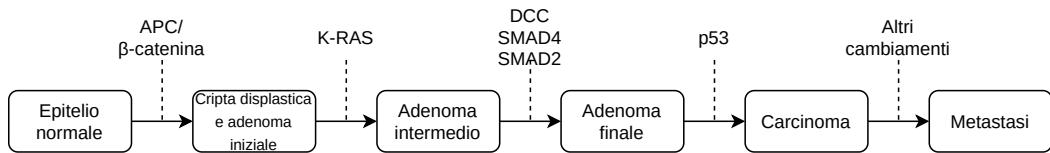


Figura 8.1: Schema semplificato di una progressione del cancro al colon.

Dove:

- si parte con un *epitelio normale* che è quello che normalmente abbiamo nell'intestino per tantissimi anni
- ad un certo punto può darsi che alcune delle cellule, normalmente alcune delle cellule staminali o alcuni dei progenitori semi-differenziati, acquisiscono una mutazione sui geni *APC* e *β-catenina*. Non sappiamo come vengono acquisite tali mutazioni
- se le mutazioni rimangono fisse, ovvero le cellule che hanno la mutazione si riproducono senza essere rimosse dal sistema immunitario o da altri sistemi di controllo. Queste cellule quindi crescono in maniera **non ordinata**, perdendo il meccanismo di regolazione della crescita, avendo una **crescita displastica**. In questa situazione si ha quindi un **adenoma iniziale**
- dopo questa situazione può darsi che una di queste cellule mutate acquisisca anche una mutazione sul gene *K-RAS* che è un gene implicato nella regolazione della proliferazione delle cellule. A questo punto il sistema diventa un **adenoma intermedio**
- a questo punto le cellule possono acquisire anche le mutazioni sui geni *DCC*, *SMAD4* e *SMAD2* (dove si noti che *DCC* sta per *Deleted in Colorectal Carcinoma*) portando il sistema ad essere un **adenoma finale**
- se si aggiunge anche la mutazione su *p53*, che è uno dei geni, più importante della categoria, che controlla il meccanismo di “suicidio volontario” della cellula, ovvero il processo di **apoptosi** che porta la cellula a smettere di riprodursi e ad essere rimossa e “ricicidata” dal sistema. Se però si ha la mutazione su *p53n* il sistema smettere di rispondere ai segnali *apoptotici* e di fatto la cellula diventa “immortale” e il sistema diventa un **carcinoma**

- a causa di altri cambiamenti, infine, si ha la **metastasi**

Questo è stato uno dei primi modelli di progressione del tumore, in particolare del cancro al colon. Ovviamente ora si sa che questo schema è troppo semplificato ma è comunque un buon riferimento per il “normale” cancro al colon.

Si hanno però altri tipi di cancro al colon, come il cosiddetto **Hereditary Non-polyposis Colorectal Cancer (HNPCC)**, che presenta una serie diversa di mutazioni. Un’altra variante di progressione è quella detta **Hypermethylation MSI-H**, dove la sigla sta per *MicroSatellite Instability-High*. Si hanno quindi vari tipi di progressione anche se gli ultimi elencati hanno incidenza minore rispetto a quello “normale”.

Si hanno anche altri meccanismi relativi allo sviluppo di tumori, tra cui il **VEGF (Vascular endothelial growth factor) pathway** che è relativo allo **angiogenesi**, ovvero il processo che porta alla formazione di nuovi vasi sanguigni da altri vasi preesistenti. Un altro problema è l’interazione delle cellule con lo **stroma**, ovvero la trama fondamentale di un organo composto generalmente dal tessuto connettivo e dai vasi sanguigni, e con il microambiente complessivo, dove si hanno *morfogeni* presenti in concentrazione diversa.

8.3 Simulazioni 3D

Vediamo quindi, comunque in modo molto sintetico e schematico, come fare simulazioni tridimensionali.

Per modellare un *tessuto tumorale* si hanno differenti **paradigmi di modellazione** e diversi **livelli di astrazione, parametri etc....**

Bisogna considerare, ovviamente oltre all’implementazione stessa, quali siano i *vincoli* da considerare e quali sono le grandezze da osservare. Si vedranno:

- **modelli su griglia**, detti anche **in-lattice**
- **modelli senza griglia**, detti anche **off-lattice** o **lattice-free**

Bisogna considerare anche *forze esterne, flussi, condizioni a contorno* etc... Sarebbe bello dare regole locali per ogni agente e osservare ex-post l’organizzazione in colonne della riproduzione delle cellule. La cripta è una struttura tridimensionale, a semi-cilindro, che però può essere “srotolata” in un piano, ottenendo una trasformazione del problema in un problema bidimensionale, in 2D, che è molto più semplice da studiare.

8.3.1 Modelli In-Lattice

Passiamo quindi ai **modelli su griglia**, detti appunto **modelli in-lattice**. Il punto chiave che differenzia i vari modelli è come modellare gli agenti principali, nel dettaglio le varie cellule. Nei modelli *in-lattice* una cellula è modellata come una composizione di posizioni sulla griglia e quindi non come un oggetto a se stante con una sua geometria, come nei modelli **off-lattice**.

Cellular Potts Mode

Un modello *in-lattice* molto importante è il **Cellular Potts Model (CPM)**, un modello che nasce nel mondo della fisica che poi è stato riutilizzato in ambito biologico. Questo è un modello ad **automi cellulari**.

Uno dei primi usi del *CPM* è quello sviluppato da Glazier e Graner, nel 1992, che è basato sul modello di Potts di meccanica statistica, ovvero uno *spin model*. Quando si misero a studiare questa simulazione lo fecero per studiare la cosiddetta **Differential Adhesion Hypothesis (DAH)**, proposta da Steinberg nel 1952 e validata sperimentalmente nel 1962 da Moscona e Moscona. Questa ipotesi è stata proposta da dei biologi ed enuncia, in modo semplice, che se si hanno due popolazioni di cellule mischiate (normalmente si vede con popolazioni batteriche) e queste cellule aderiscono l'una con l'altra tra le due popolazioni allora dopo un po' il sistema tenderà a riorganizzarsi in strati (per dire come succede quando si mischiano olio e acqua anche se ovviamente con le cellule, che sono vive, è un discorso più complesso).

L'implementazione di questo sistema viene fatta mediante il calcolo complessivo dell'energia del sistema (indicata da Potts con J), energia che dipende dai *parametri di adesione* tra una cellula e un'altra, di tipo diverso. L'energia del sistema dipende quindi dalla posizione relativa di diverse cellule. L'evoluzione del sistema ci permette di calcolare, passo per passo, di calcolare J e di scegliere le prossime mosse per **minimizzare** J . Dal punto di vista dei vincoli abbiamo quello dell'**energia elastica** che dipende dal volume della cellula.

In questo sistema si osservano vari fenomeni:

- *auto-ordinamento delle cellule*
- *posizionamento delle cellule*
- *riarrangiamento e migrazione delle cellule*

Vediamo quindi la rappresentazione interna di questo sistema.

Abbiamo una griglia bidimensionale e l'idea chiave è quella di discretizzare lo spazio in questa griglia. Ad ogni quadratino/pixel, identificato con le

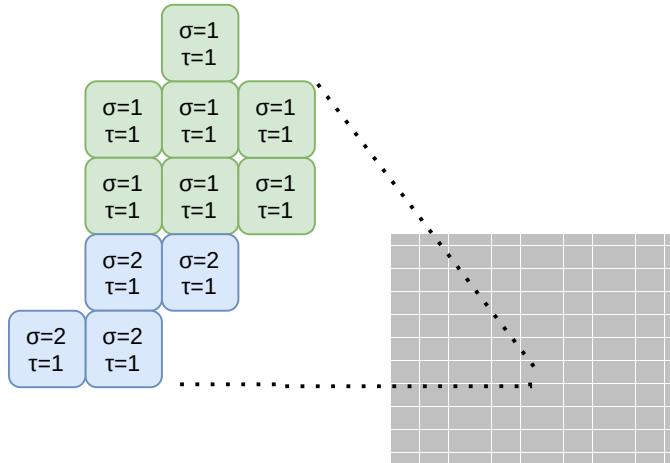


Figura 8.2: Esempio di parte della griglia, con due cellule diverse ($\sigma = 1$ e $\sigma = 2$) dello stesso tipo ($\tau = 1$)

coordinate (i, j) , di questa griglia si assegnano dei numeri (come in figura 8.2):

- $\sigma(i, j) = id$, ovvero una funzione che calcola un **id**, per indicare a quale cellula appartiene il pixel, infatti ogni cellula è rappresentata come un insieme连通的 di pixel sulla griglia
- $\tau(\sigma(i, j)) = type$, ovvero una funzione che assegna un **type** ad ogni cellula

L'energia totale del *CPM* è legata alla superficie delle cellule. Si ha quindi la funzione:

$$J(\tau, \tau')$$

che rappresenta l'energia di superficie per unità di contatto tra due cellule. L'energia è minore tra i *pixel* appartenenti a due cellule dello stesso tipo, maggiore per cellule di tipo diverso e nulla tra *pixel* della stessa cellula.

Definizione 4. Definiamo l'energia totale del *CPM* è definita come la somma totale delle energie calcolare sulle superfici delle cellule più una costante moltiplicata per l'energia ottenuta dall'area di una cellula, ovvero:

$$H_{Potts} = H_{surface} + \lambda H_{Area}$$

Dove H_{potts} è un operatore Hamiltoniano e λ è la **costante di rigidità** per il citoscheletro. Tale costante è usata nel ruolo di moltiplicatore Lagrangiano.

L'algoritmo di simulazione è molto semplice. Ad ogni passo si seleziona randomicamente il pixel in posizione (i, j) e si fa un calcolo probabilistico per convertire l'*id* $\sigma(i, j)$ nel σ' di un pixel confinante. Si calcola quindi la seguente probabilità (indicando con \rightarrow il passaggio di un *pixel* da una cellula all'altra), per $T \neq 0$:

$$P(\sigma(i, j) \rightarrow \sigma'(i, j)) = \begin{cases} e^{-\frac{\Delta H}{kT}} & \text{se } \Delta H > 0 \\ 1 & \text{se } \Delta H \leq 0 \end{cases}$$

dove:

- ΔH è il cambio di energia totale ad ogni step
- k è una costante
- T è il tempo

mentre se si ha $T = 0$, avendo quindi la condizione iniziale:

$$P(\sigma(i, j) \rightarrow \sigma'(i, j)) = \begin{cases} 0 & \text{se } \Delta H > 0 \\ 0.5 & \text{se } \Delta H = 0 \\ 1 & \text{se } \Delta H < 0 \end{cases}$$

È quindi un sistema molto semplice, ancora più semplice da implementare di quello di Gillespie.

Alla fine anche questo è un algoritmo della categoria dei **metodi Montecarlo**, infatti passare da uno stato della griglia all'altro è fare un passo con quel metodo.

Inoltre si ha anche il **gradiente DAH** che “guida” la probabilità che le cellule cambino da un certo tipo ad un altro specifico tipo, facendo un *passo di Montecarlo*. Si osserva quindi il “movimento” atteso delle cellule semplicemente implementando questo metodo.

Su slide, parte 6, varie immagini di simulazioni, dove in base a diversi vingoli si ottengono separazioni nette tra tipi di cellule o miscugli, in assenza di particolari vincoli.

Si hanno numerosi problemi, durante le simulazioni più complesse, del mantenimento finale della forma di determinate cellule, cosa che si risolve tendenzialmente aggiungendo vincoli.

Modello di Wong

Vediamo quindi il *modello in-lattice* per le cripte coloniche proposto da Wong¹. Questo modello ci permette di modellare la **motilità**, ovvero proprietà del-

¹Wong et al., Computational model of cell positioning: directed and collective migration in the intestinal crypt epithelium, J. Of The Royal Soc. Interface 7, S351-363 (2010)

l'organismo vivente (anche unicellulare), o di una sua parte, di modificare attivamente e in modo reversibile la propria posizione rispetto all'ambiente, delle cellule staminali e delle cellule progenitrici all'interno delle cripte.

È un modello a due dimensioni dove sono stati aggiunti alcuni elementi, tra cui il fatto che la cripta è “srotolata” e viene forzato esternamente tramite un gradiente il movimento verso il *lumen* da parte delle cellule.

Dal punto di vista dei vincoli si hanno:

- si ha un *termine di energia* che fa sì che le cellule staminali rimangano fisse in posizioni predefinite
- regole particolari che servono a modellare la crescita cellulare, la divisione (asimmetrica con proliferazione e differenziazione) e l'*apoptosi*
- si ha un'assunzione fondamentale di modello che ci dice che il morfogene più importante che viene considerato, che è quello che fa sì che le cellule si dividano e salgano verso il *lumen*, ovvero l'**adesione** sia data da livelli di attivazione di *Eph/efrina* lungo la cripta

Il risultato finale della simulazione consente di osservare che:

- l'adesione differenziale regola il posizionamento delle cellule nella cripta intestinale
- le cellule epiteliali si muovono verticalmente verso l'alto verso il *lumen* della cripta
- il movimento è coordinato, nonostante le relativamente poche assunzioni
- l'omeostasi delle cellule epiteliali intestinali è mantenuta nel modello

Nonostante, si ripete, la simulazione sia semplice e con poche assunzioni.

Modello Ibrido di Glazier, Graner e Hogeweg

Un modello un poco più complesso è stato proposto da Glazier, Graner e Hogeweg e consiste nel *CPM* con l'aggiunta della modellazione della **chemiotassi**, ovvero il fenomeno con cui i corpi cellulari, batteri ed altri organismi unicellulari o multicellulari direzionano i loro movimenti a seconda della presenza di alcune sostanze chimiche nel loro ambiente.

Questo è quindi un modello sostanzialmente **ibrido**, in quanto viene appunto modellato il movimento esplicito delle cellule nell'ambiente.

Per ottenere ciò si usano dei gradienti modellati con delle **equazioni a derivate parziali (EDP)**. Si usano **campi discretizzati** per descrivere la concertazione di un attrattore chimico, definiti da un insieme di EDO all'interno di ogni cellula, che descrivono la secrezione cellulare e l'assorbimento della sostanza chimica, ovvero dei morfogeni. Si ha inoltre un insieme di *EDP* per calcolare la diffusione e il decadimento della sostanza chimica, ovvero a regolare l'attivazione dei morfogeni. Si ha quindi un **reaction-diffusion system**. Si ignora invece l'**avvezione**, ovvero il trasporto di massa o proprietà fisica che in fluidodinamica avviene durante il moto del fluido. Si ha quindi un contributo aggiuntivo all'energia effettiva del *CPM* per tenere conto della nuova variabile di campo.

Il *CPM* quindi è usato per modellare il movimento delle cellule, tramite la griglia, e per tener traccia dell'energia complessiva del sistema.

L'uso del sistema di *EDP* consente di osservare:

- la chemiotassi
- la proliferazione delle cellule tumorali
- secrezione e assorbimento del *fattore pro-angiogenico VEGF-A*, che è un fattore di crescita che serve per regolare l'afflusso di nutrienti al tumore (???)
- la proliferazione di *cellule neovascolari*
- la produzione, l'assorbimento e la diffusione di ossigeno

Dal punto di vista software si ha un implementazione con **CompuCell3D** (sul sito ci sono anche vari video di simulazioni etc...).

8.3.2 Modelli Lattice-Free

Passiamo quindi ai modelli **lattice-free/off-lattice**, quindi senza griglia. Questi modelli arrivano dal mondo della fisica.

Questi modelli 3D si basano sulla **partizione dello spazio di Voronoi-Delaunay** (anche se sarebbero nel dettaglio la **partizione di Voronoi** e la **triangolazione di Delaunay**). Tali modelli sono modelli *biomeccanici, ibridi* e *agent-based* di sistemi multicellulari.

Modello di Schaller e Meyer-Hermann

Come anticipato questo tipo di modelli deriva dal mondo della fisica, infatti il primo modello che viene trattato è stato fatto in quell'ambiente da Schaller e Meyer-Hermann².

Si hanno le seguenti assunzioni in generale (comuni anche ad altri modelli):

- la **cellula** è un *corpo semi-sferico* e la forma varia da sferica in soluzione sottile a *convessa poliedrica di Voronoi* in tessuti densi. Questa modellazione rende tutto più semplice
- dal punto di vista delle **forze** si hanno:
 - la *forza elastica* tra cellule
 - *interazioni/forze di adesione e frizione* tra cellule
 - *interazioni/forze di adesione e frizione* tra cellula e substrato
- lo **stato della cellula** è rappresentato da:
 - posizione
 - concentrazione nella membrana cellulare di recettori e ligandi. Questa informazione è necessaria per parlare di adesione
 - il ciclo interno alla cellula
 - altre caratteristiche, rappresentate da costanti, utili per le interazioni elastiche e di adesione. Tali costanti sono dipendenti dal tipo preciso di cellula
- per la **dinamica** con cui le cellule si muovono nello spazio si usano **equazioni di Newton** per le singole cellule, corredate da *EDP* per modellare campi in cui si muovono le cellule, dove ci sono processi di reazione e diffusione che coinvolgono nutrienti, come Ossigeno e Glucosio. Le *EDP* determinano quindi la distribuzione spazio-temporale dei nutrienti
- si hanno infine **segnali di crescita** che fanno sì che aumenti la *biomassa* (aumentando di volume fino ad un certo limite fino alla divisione), quando le cellule consumano i nutrienti

²Schaller, Meyer-Hermann, PHYSICAL REVIEW E 71, 051910 s2005d

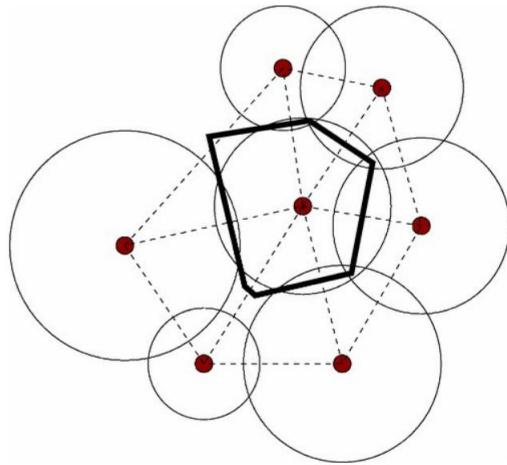


Figura 8.3: Esempio di triangolazione bidimensionale dove viene specificata la partizione del punto rosso centrale, tutti i punti che posso prendere in quell'area sono più vicini al punto rosso centrale che a qualsiasi altro punto rosso della figura.

La **triangolazione di Delaunay** viene utilizzata per fornire in modo efficiente l'elenco dei prossimi vicini per l'interazione cellula-cellula.

La **partizione di Voronoi** e la **triangolazione di Delaunay** sono duali e sono usate per rappresentare in modo approssimato cellule e tumori sferici. Si hanno estensioni per un numero n di dimensioni.

Parliamo dell'estensione bidimensionale. Si hanno dei punti in un piano che vengono connessi tra loro con linee tratteggiate, come in figura 8.3. Non si possono avere linee che si intersecano tra loro e tali linee rappresentano appunto la *triangolazione di Delaunay*. Ci sono algoritmi che costruiscono la triangolazione in modo efficiente, basandosi su una *scan-line* e man mano che si ha la scansione si costruisce la triangolazione. Si procede poi prendendo il punto medio di ogni segmento e costruisco un poligono in grassetto che passa coi suoi lati attraverso tali punti medi. La zona delimitata da questo poligono è la **cellula di Voronoi** e corrisponde alla costruzione *della partizione di Voronoi* nel piano. Ovviamente la partizione dipende dalla posizione dei punti e si possono generare più partizioni, dove ogni partizione contiene un punto di quelli di partenza e tutti i punti ipotetici della ragione hanno quel tal punto come punto più vicino. Si possono ottenere così approssimazioni accettabili.

Rappresentare le cellule con questi **diagrammi di Voronoi** facilita di molto la computazione di misure e quantità di interesse per capire come si evolve il sistema, potendo identificare schematicamente le forze, atte a calcolare poi

i cambiamenti nella posizione delle cellule (con le forze che producono una forza risultante che muove la cellula in certe direzioni), ma anche a poter rappresentare la suddivisione delle cellule. Le varie forze vengono calcolate come agenti sulle facce della superficie calcolata (si associa un singolo valore ad ogni superficie, nel punto medio di ogni lato) e quindi si può computare meglio come ottenere la risultante. È anche questa un'approssimazione accettabile (anche perché tali calcoli su una curva non sarebbero gestibili). Tali calcoli, come quello della forza risultante dipende poi dal calcolo di integrali non banali. Per la divisione si ha che si costruisce un nuovo *diagramma di Voronoi*, avendo che da un punto se ne producono due. Si hanno questioni algoritmiche non banali in quanto, dato un insieme di punti, su un piano o anche in uno spazio 3D, è semplice costruire il *diagramma di Voronoi/triangolazione di Delaunay*, avendo algoritmi noti, mentre si hanno problemi quando si ha uno spazio partizionato e si vuole modificare la struttura del sistema. Tutti gli algoritmi noti infatti agiscono a livello *globale* ma per modellare una divisione cellulare ho una modifica *locale* al *diagramma di Voronoi*, non cambiando quasi nulla nelle aree lontane. È quindi inefficiente ricalcolare tutto ad ogni divisione cellulare e quindi si hanno varie soluzioni algoritmiche non banali, di natura numerica, che permettono di fare aggiornamenti locali in modo abbastanza efficiente. La letteratura dedicata si ritrova sotto il nome di **dynamic update Voronoi diagrams** e spesso trattano soprattutto dell'errore introdotto dagli algoritmi, usando calcoli numerici non semplici da gestire, avendo problematiche dal punto di vista della **robustezza degli algoritmi**.

Tali simulazioni hanno anche avuto uso in *meccanica muscolare* e *meccanica scheletrica*, oltre che a vari problemi ingegneristici (turbine, centrali nucleari etc...).

Questo modello include la **proliferazione**, secondo regole predefinite, e si voleva modellare la dinamica della popolazione e il mantenimento della morfologia sferoidale del tumore, studiando i meccanismi per l'induzione della **necrosi cellulare**.

Modello di Buske e Galle

Un modello diverso è quello proposto da Buske, Galle et al³, tra cui Drasdo. Questo è un modello biomeccanico *agent-based* per sistemi multicellulari. Le cellule sono trattate come corpi elastici deformabili, pur tenendo a mantenere di base forma sferica, in grado di:

³P. Buske, J. Galle, et al, A Comprehensive Model of the Spatio-Temporal Stem Cell and Tissue Organisation in the Intestinal Crypt, PLoS Comput Biol, vol. 7, no. 1, 2011

- muoversi
- dividersi
- differenziarsi
- “comunicare” tra loro

La dinamica viene rappresentata tramite le **equazioni di Langevin**.

Si usano variabili di stato dinamiche per:

- la posizione delle cellule
- la dimensione delle cellule
- il vettore dello stato interno (*internal state vector*) per l'*internal activity status*, con ad esempio la trascrizione dei geni target, con, ad esempio, il *Wnt-pathway* o il *Notch-pathway*

La forma delle cripte intestinali è quella di una “scalfatura” 3D abbastanza fedele alla vera forma di una cripta.

Il vettore dello stato interno può essere cambiato tramite la dinamica esterne alle cellule, avendo dinamica delle posizioni e delle dimensioni. Non si hanno invece equazioni di dinamica per lo stato interno, non avendo, ad esempio, relazioni metaboliche e questo è importante visto che spesso si vuole accoppiare la dinamica interna di una cellula con quella esterna, modellando sia i processi intra-cellulari che inter-cellulari. La scala dei tempi in cui questi processi avvengono è però abbastanza diversa avendo che la simulazione potrebbe rallentare molto.

Dal punto di vista delle forze si hanno quindi le interazioni tra cellule e le interazioni di una cellula con il substrato, in termini di adesione e frizione.

Il modello puntava quindi a descrivere lo stato stazionario delle cripte colo-niche, stato in cui si ha una *dinamica stazionaria*, dove si ha la divisione e la differenziazione delle cellule. Si parla di **omeostasi**, la capacità di autoregola-zione degli esseri viventi, importantissima per mantenere costante l’ambiente interno nonostante le variazioni dell’ambiente esterno, ovvero l’*equilibrio dinamico*.

Il modello tiene traccia della crescita delle cellule, della divisione e della dif-ferenziazione delle stesse, per tutte e tre le cose con regole predefinite scelte tramite studi sperimentalni. A questi vincoli si aggiungono:

- dal punto di vista dell’energia di interazione abbiamo:
 - interazione di adesione

- l'energia di deformazione per contatto, secondo il modello di Hertz
- energia di compressione elastica per le sfere
- per le forze si ha che essere sono sia deterministiche che stocastiche, anche se i termini stocastici sono solamente *termini di rumore*
- si assume l'ancoraggio al substrato. Le cellule interagiscono con la *membrana basale*, ovvero la struttura laminare specializzata della matrice extracellulare di spessore compreso tra 70 e 300 nm che fa da interfaccia tra un tessuto connettivale e un tessuto non connettivale, tipicamente epitelio. Questa interazione però avviene se la loro distanza è minore di quella del loro raggio e si ha che esse sono rimosse dal sistema della cripta se si perde il contatto con la membrana basale

Si ha inoltre un *ciclo cellulare* programmato, per controllare:

- la regolazione e il controllo della crescita cellulare
- la forma di inibizione della crescita mediata dal contatto tra cellule (???)
- l'arresto del ciclo cellulare in dipendenza dal contatto tra una cellula e il substrato
- la morte cellulare programmata in dipendenza dal contatto tra una cellula e il substrato, simulando quindi processi apoptotici

Questo modello consente di osservare:

- l'omeostasi del sistema
- l'auto-organizzazione spazio-temporale del tessuto
- la migrazione delle cellule verso il *lumen* della cripta
- la riorganizzazione delle cellule nel sistema

Tutto questo studiando *sensitività*, *stabilità* e *flessibilità* al cambiamento dei livelli di attivazione dei segnali di *Wnt* e *Notch*. La proliferazione e la differenziazione dipendono infatti dall'attivazione del *Wnt-pathway* e del *Notch-pathway*, dalla posizione delle cellule, tramite il *Wnt-gradient* e il *Notch-gradient* e dalla curvatura della membrana basale del substrato.

Modello di Drasdo

Un altro modello è quello detto **modello di Drasdo**⁴.

L'articolo indicato contiene questo modello anche se in realtà Drasdo aveva in primis introdotto un modello 1D, usando “stringhe di cellule”, per le *cripte intestinali* e aveva ha introdotto le prime idee sui modelli *agent-based* individuali, ovvero aveva introdotto i modelli *off-lattice/lattice-free*. Solo dopo Galle ha iniziato ad applicare tali idee.

8.3.3 Obiettivi Futuri

Dopo aver visto alcuni modelli è interessante pensare a qualche obiettivo futuro, elencando le varie *problematiche aperte*. Alla base si ha sempre la ricerca di interazioni tra i diversi livelli di simulazione e tra le idee possibili abbiamo:

- combinare il modello di Galle per le cripte con l'idea di Schaller e Meyer-Hermann per la rappresentazione delle cellule tramite *partizioni di Voronoi* e *triangolazione di Delaunay*
- combinare il modello di Wong delle cripte dinamiche con il modello di Hogeweg per la diffusione dei nutrienti e le reazioni, nel caso 2D. Si otterrebbe un modello bidimensionale che sarebbe quello di Wong con l'aggiunta di *EDP*
- combinare il modello di Wong delle cripte dinamiche con il modello di Hogeweg per la diffusione dei nutrienti e le reazioni, nel caso 3D. Si otterrebbe un modello tridimensionale che sarebbe quello di Wong con l'aggiunta di *EDP* ma bisognerebbe capire come implementare *CPM* in una forma per le cripte non cilindrica, come nel caso del modello di Galle delle cripte. Qui la situazione diventa davvero complessa

Gli sforzi futuri dovrebbero puntare alla formazione di modelli ragionevoli in cui stimoli meccanici e processi intracellulari biochimici sono accoppiati in un quadro dinamico unificato, con un modello che permetta di ragionare in modo accettabile su tutti questi aspetti nel loro complesso. I modelli di crescita, divisione e differenziazione cellulare dovrebbero essere in grado di descrivere, a livello microscopico, l'esatto stimolo meccanico che innesta la

⁴J. Galle, G. Aust, G. Schaller, T. Beyer, and D. Drasdo, “Individual cell-based models of the spatial-temporal organization of multicellular systems—Achievements and limitations,” *Cytometry*, vol. 69, no. 7, pp. 704–710, 2006.

crescita e successivamente le reazioni biochimiche che portano alla divisione e alla differenziazione. Inoltre questi modelli dovrebbero chiarire come le cellule percepiscono i segnali meccanici e li convertono in quelli chimici che alla fine inducono una risposta biologica come la crescita. Bisognerebbe quindi modellare il meccanismo sensoriale delle cellule, che dal punto di vista biologico avviene tramite le membrane e permette di far capire ad una cellula cosa accade attorno, ovvero cosa c'è sulle membrane delle cellule vicine, vario segnali molecolari presenti nell'ambiente in cui le cellule si trovano etc... .

Al momento, la ristretta conoscenza complessiva delle diverse reazioni biochimiche coinvolte nella crescita, divisione e differenziazione cellulare rende molto difficile fornire un modello meccanico macroscopico che spieghi tutti i diversi passaggi microscopici coinvolti nei pathway di trasduzione del segnale. Non conosciamo tutti i dettagli dell'interazione tra cellule. Quindi la ricerca di settore si concentra attualmente nella **modellazione multiscala** (***multiscale modelling***), cercando di risolvere quindi il problema della differenza di scale temporali, per la rappresentazione congiunta di dinamiche intra-cellulari e inter-cellulari. Il problema del *multiscale modelling* non è molto diverso dal problema visto parlando del *tau-leaping* per Gillespie, facendo un passo ad una certa scala cercando di assicurarsi che ci siano delle condizioni che garantiscano che il passo non fa “sforare troppo” da quanto avrei ottenuto muovendomi con i micro-passi a scale inferiori. In questo caso ovviamente gli strumenti matematici per ottenere questo risultato sono assai più variegati e complessi.

Capitolo 9

Flux Balance Analysis

Finora sono state viste diverse tecniche di simulazione. Si presenta ora una tecnica di analisi, detta **Flux Balance Analysis (FBA)**.

In questo contesto si analizzano i sistemi biologici da un punto di vista diverso, ovvero quello del **metabolismo**. Si ricorda che il metabolismo di un sistema è rappresentato da un insieme di *reazioni metaboliche*, cioè cioè dalla *matrice stechiometrica* dell'insieme delle reazioni. Nel dettaglio con la *FBA* si studiano i **flussi** di sostanze chimiche nel sistema¹².

9.1 Pathway Metabolici

Il punto chiave è che rappresentare il metabolismo, con tutti i suoi pathway (eventualmente poi rappresentati tramite insiemi di equazioni), è veramente complesso per questo lo studio viene fatto *in silico*, nell'ambito della **biologia computazionale**.

Il **metabolismo cellulare** a partire dal cibo e tramite *pathway catabolici* produce:

- energia
- componenti utili alla sintesi delle cellule
- calore

e, tramite i primi due e i *pathway anabolici*, vengono prodotte varie macromolecole.

¹K. Raman and N. Chandra, “Flux balance analysis of biological systems: applications and challenges,” *Briefings in Bioinformatics*, vol. 10, no. 4, pp. 435–449, Jun. 2009.

²J. D. Orth, I. Thiele, and B. Ø. Palsson, “What is flux balance analysis?,” *Nature Publishing Group*, vol. 28, no. 3, pp. 245–248, Mar. 2010.

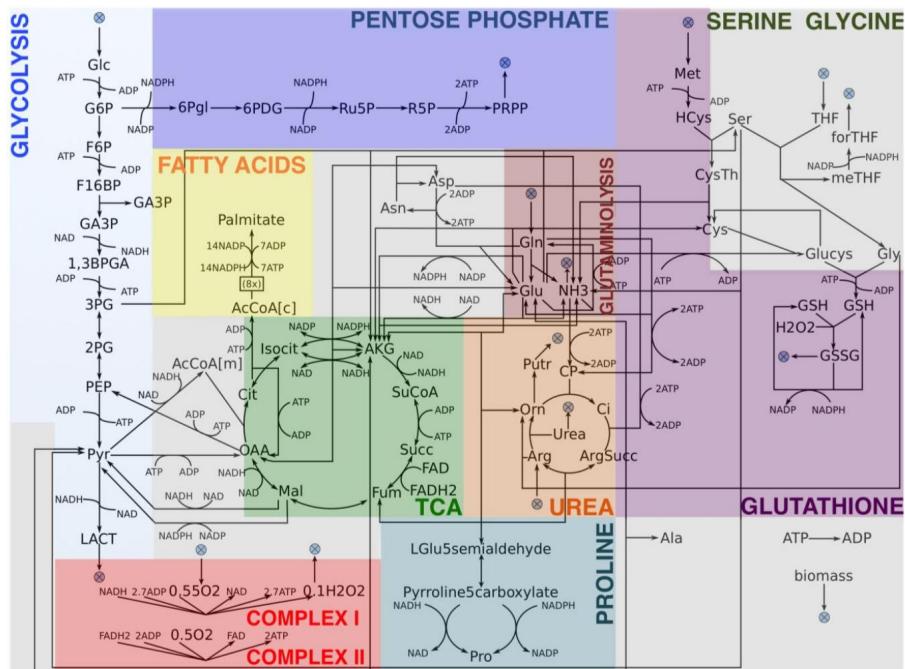


Figura 9.1: Sezione del metaboloma, rappresentata in modo “astratto”, con una dozzina di pathway connessi, tra cui i già citati pathway della glicolisi e del ciclo di Krebs.

Definizione 5. Si definisce **pathway metabolico** una sequenza, o meglio un grafo, di reazioni metaboliche.

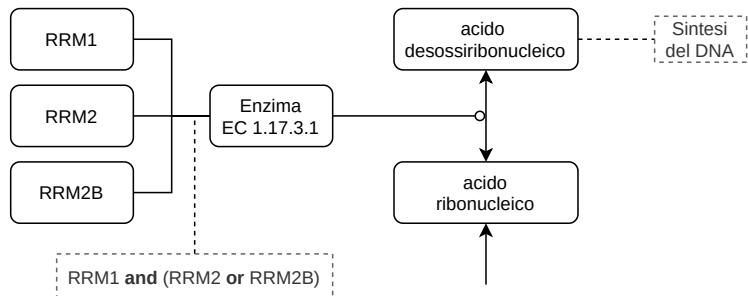
Un esempio di pathway metabolico è quello già introdotto della **glicolisi** che, a partire da una molecola di Glucosio produce *ATP* e due molecole di piruvato, che, nel caso di un processo di respirazione/fermentazione anaerobica, diventa lattato.

Un altro pathway importante, che lavora in combinazione con la **glicolisi**, è **ciclo di Krebs**, detto anche **ciclo TCA** (da *tricarboxylic acid cycle*), un meccanismo metabolico presente in praticamente tutti gli organismi, attivo a gestire l'ossigeno fornito dalla respirazione.

Se vengono presi tutti i pathway metabolici e vengono rappresentati insieme si ottiene la **rete metabolica**, detta anche **metaboloma**, di cui una sezione è visibile in figura 9.1.

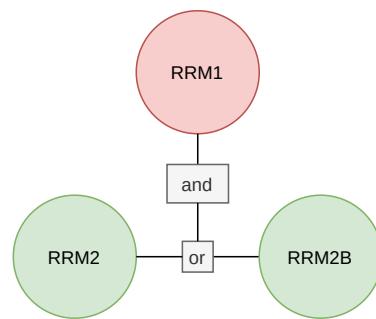
Un altro aspetto che bisogna considerare quando si vuole rappresentare sistemi biologici di questo tipo è il livello di astrazione. Una stessa reazione o uno stesso evento biologico può infatti essere rappresentato in modo più granulare o più astratto, anche se sottostante si ha comunque lo stesso comportamento.

Ad esempio potremmo avere la seguente rappresentazione più granulare:



Dove si ha una regola booleana per l'enzima/gene che coinvolge l'enzima **RRM1** (*Ribonucleotide Reductase Large Subunit*), che è una sub-unità catalitica, l'enzima **RRM2** (*Ribonucleoside-diphosphate reductase subunit*) e l'enzima **RRM2B** (*p53-Inducible Ribonucleotide Reductase subunit*), dove le ultime due sono sub-unità regolatorie isoformi. Si nota che l'uso di *funzioni booleane* rende semplice il passaggio dalla modellazione astratta al software.

Tutto questo, che non approfondiamo dal punto di vista biologico, però può essere rappresentato in modo più astratto nel seguente modo, ad esempio:



dove si ha astrazione a *livello molecolare*, a cui si potrebbe aggiungere la conformazione delle proteine.

Si hanno vari motivi per studiare il metabolismo, andando oltre il mero studio di sequenze tipico della **bioinformatica**, infatti la deregolamentazione del metabolismo cellulare è coinvolta in molte malattie, come:

- cancro, dove si noti le cellule usano anche pathway differenti da quelli standard per la produzione di energia. In merito si ha anche lo studio dei tumori sia dal punto di vista interno alla cellula, *intra-tumor*, che in un insieme di cellule, *inter-tumor*
 - diabete

- obesità
- malattie legate al fegato grasso
- Parkinson
- Alzheimer

Ma lo studio è anche legato ad analizzare gli effetti dell'invecchiamento, infatti il metabolismo cambia con l'avanzare degli anni.

Un altro motivo per cui si studia il metabolismo è l'ingegnerizzazione dello stesso, specialmente parlando di batteri. Si è visto con l'esempio del Repressilator come si possa indurre una popolazione a compiere una certa azione. Studi simili sui batteri hanno portato al loro uso per la produzione di:

- prodotti chimici di base
- sostanze di *chimica fine*
- farmaci
- combustibili

Per capire, all'interno del nostro sistema, quali siano le parti più attive possiamo quindi studiare il *flusso*. All'atto pratico:

- **si può** misurare il **metaboloma**, misurando la concentrazione dei *metaboliti*, ovvero un qualsiasi prodotto terminale o intermedio del metabolismo, ad un certo tempo t
- **non si può** misurare il **flussoma**, ovvero i flussi metabolici nell'intervallo Δt

Definizione 6. In questo contesto definiamo **flusso** come il tasso delle reazioni "in avanti" meno il tasso delle reazioni "all'indietro":

$$\text{flux} = \text{rate}_{\text{forward}} - \text{rate}_{\text{backward}}$$

Sarebbe comunque molto interessante per studiare *metaboloma* e *fluxome*, per capire quali reazioni sono **up-regolate** e quali **down-regolate**.

9.2 Programmazione Lineare

Si hanno quindi varie soluzioni modellistiche, che si contrappongono alle difficoltà di una vera e propria simulazione (tra cui capire le costanti, scegliere i parametri, eseguire la vera e propria simulazione etc...):

- **modellazione statica**
- **modellazione dinamica**
- **modellazione dello steady-state**, che è una via di mezzo che assume lo stato di equilibrio del sistema, permettendo un'analisi più semplice

In questo contesto viene anche utilizzata una delle nozioni base della *Ricerca Operativa*, ovvero la ricerca di ottimo tramite approssimazione con **programmazione lineare**. Si hanno quindi:

- la matrice stoichiometrica S , dove le colonne sono le reazioni (magari nel caso semplice *reazione \rightarrow prodotto*) e le colonne sono i metaboliti
- lo steady state $S \cdot v = 0$
- i vincoli di flusso:

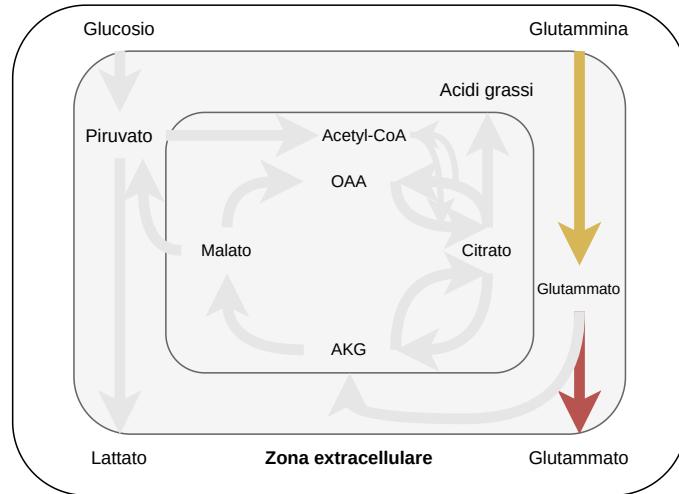
$$v_{min} < v < v_{max}$$

che rappresentano diversi tipi di flusso:

- *flussi in entrata*, tramite vincoli sui nutrienti
- *flussi interni*, tramite vincoli termodinamici e vincoli sul tasso sul *reaction rate*
- *flussi di secrezione*, per rappresentare che il metabolismo è in grado di accumulare, rappresentando quindi la crescita e i suoi vincoli

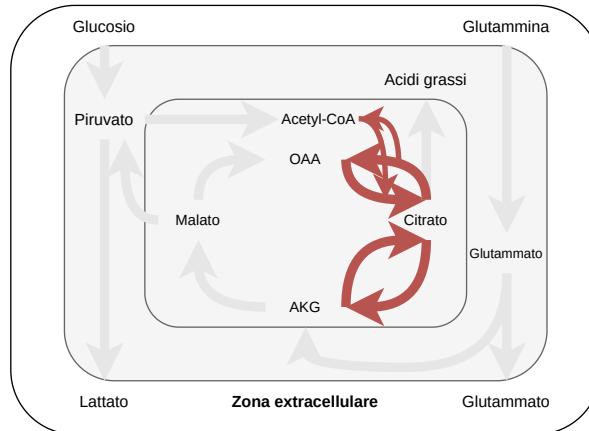
Una volta fatti i vari conti conti con al programmazione lineare otteniamo un'area dei **possibili fenotipi**, ovvero l'area delimitata dai vincoli stessi.

Prendiamo ora un piccolo esempio:



dove eventuali frecce gialle rappresentano il flusso, quindi la quantità di sostanza trasformata per unità di tempo, mentre se rosse rappresentano il massimo flusso possibile (questa notazione può variare, ad esempio usando frecce non piene per il primo caso).

Ovviamente, a seconda di dove si hanno i flussi, si possono rappresentare vari comportamenti e, nel dettaglio di questo esempio, si potrebbe anche creare un ciclo:



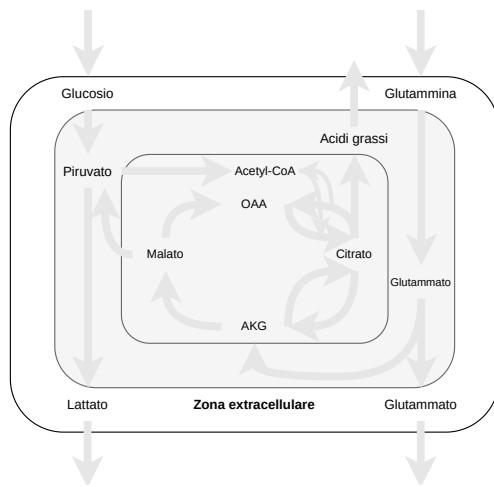
Ottenendo un Ciclo di flusso termodinamicamente irrealizzabile. Una tale situazione deve essere quindi risolta modificando o aggiungendo vincoli al sistema.

A tale sistema si aggiungono degli scambi necessari. Le reazioni di scambio

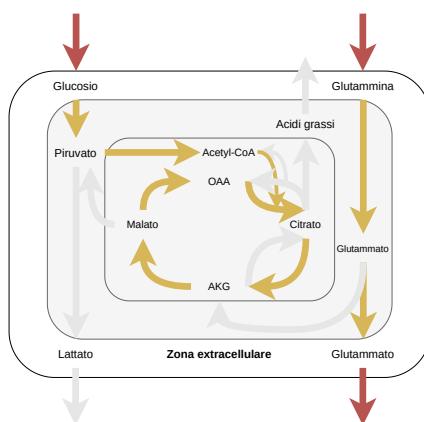
sono definite tramite due quantità e questo, per la reazione R , si indica con $R \iff$:

1. **uptake (assorbimento)**: la rimozione dall'ambiente extracellulare, ovvero il *flusso negativo*. Il valore è compreso tra $-\infty$ e 0
2. **secretion (secrezione)**: l'inserimento nell'ambiente extracellulare, ovvero il *flusso positivo*. Il valore è compreso tra 0 e $+\infty$

Si ottiene quindi:



A questo punto possiamo avere molteplici steady state rappresentabili in questo modo. Ad esempio potremmo rappresentare uno (ma non l'unico) steady state per l'assorbimento forzato di Glucosio:



Su slide lezione 7 altri possibili steady state.

Vediamo quindi come funziona una **modellazione basata su vincoli** e la **Flux Balance Analysis (FBA)**. Si ha:

- si parte con la ricostruzione metabolica su scala genomica, studiando le reazioni e capendo come funzionino i vari pathway
- si continua rappresentando matematicamente le reazioni metaboliche e i vincoli. Nel dettaglio si usa una matrice stocheometrica S , a cui vengono aggiunte delle colonne in fondo a destra per le “variabili” di interesse, per esempio biomassa, Glucosio, Ossigeno etc.... Si moltiplica tale matrice per il vettore $v = \{v_1, \dots, v_n, v'_1, \dots, v'_m\}$ contenente i vari flussi più i flussi legati alle variabili di interesse (indicate con i vari v'). Si calcola quindi $S \cdot v = 0$, si ha quindi un sistema lineare omogeneo in quanto si studia lo steady state
- il bilancio di massa definisce un sistema di equazioni lineari a partire da $S \cdot v = 0$, che sono i vincoli del sistema da studiare con la programmazione lineare
- si definisce una funzione obiettivo su cui cercare l’ottimo del tipo:

$$z = c_1 \cdot v_1 + c_2 \cdot v_2 \cdots$$

e si usa z per predire i valori di interesse, ad esempio per la biomassa, si userebbe banalmente $z = v_{biomassa}$, con $v_{biomassa}$ che è uno dei v'_i indicati sopra

- si calcola infine il flusso che massimizza z tramite programmazione lineare

La forma generica, nel caso di ricerca di un massimo, di questa funzione obiettivo e dei suoi vincoli è:

$$\begin{aligned} \max \quad & z = c^T v \\ \text{s.t.} \quad & S \cdot v = 0 \\ & \alpha \leq v_i \leq \beta \end{aligned}$$

Si ha che la funzione obiettivo può anche essere riscritta come:

$$z = \sum_i c_i \cdot v_i = c \cdot v$$

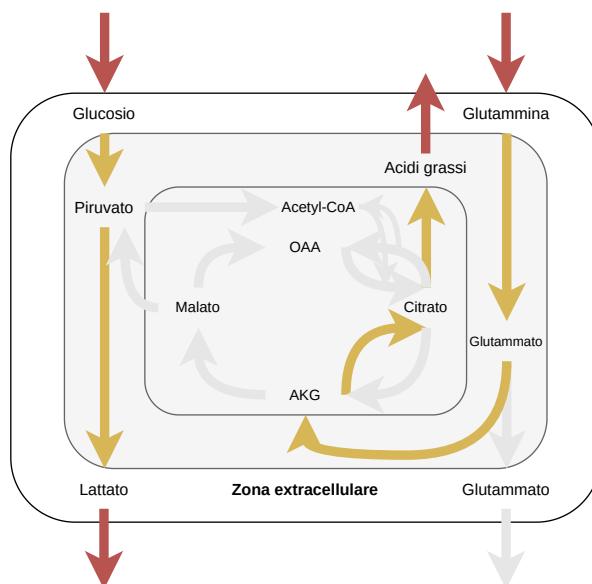
dove si identificano meglio:

- i pesi c_i
- i coefficienti obiettivo v_i

Ovviamente ci sono situazioni in cui l'ottimo è un massimo e altre in cui l'ottimo è un minimo, infatti, tra le altre, potremmo voler studiare funzioni che:

- minimizzano il consumo di Glucosio
- massimizzano la produzione di composti chimici e biocarburanti
- massimizzare la crescita
- massimizzare la biomassa
- massimizzare/minimizzare qualche particolare aspetto legato alle reazioni

Per esempio se volessi massimizzare l'acido grasso otterrei, sempre nell'esempio "giocattolo" di prima, un sistema di questo tipo:



Questi studi vengono usati ad esempio per esperimenti con il chemostato, ovvero un reattore biologico (o bioreattore) ideale che lavora in condizioni di stato stazionario.

Nel reattore biologico, i microrganismi presenti usano il substrato presente nella portata di alimentazione per la crescita. Le condizioni stazionarie implicano da un lato che il substrato non si accumuli all'interno del reattore e,

dall'altro, che la biomassa prodotta sia uguale a quella allontanata per unità di tempo.

In questo contesto si fanno esperimenti sui vincoli, come la produzione e il consumo di nutrienti, studiando ad esempio come massimizzare la crescita, cercando di predire questi fenomeni in *dry-lab*.

Si ha quindi che la *programmazione lineare* è usata insieme alla *FBA* per assicurare la fattibilità dei flussi, per ottimizzare la funzione che è combinazione lineare dei prodotti del sistema e per ottimizzare le condizioni a contorno. Ovviamente tale studio può risultare complesso dal punto di vista computazionale. Solitamente si parte con il famoso **algoritmo del simplesso**, che solitamente è sufficiente anche se può diventare di complessità esponenziale in certi casi. In caso di “fallimento” di questa soluzione, al variare anche della presenza di variabili discrete o continue, si procede selezionando altri algoritmi (ed eventualmente altri risolutori software).

9.3 Questioni Avanzate per FBA

Come abbiamo visto usando i metodi di programmazione lineare si ottiene una soluzione ottima ma questo aspetto può essere a volte limitante. Il problema da affrontare è essenzialmente quello di classificare/scegliere i diversi “stati” metabolici che emergono in un’analisi, al cambiare, ad esempio, dei parametri o della funzione obiettivo. Questo tipo di discorso diventa interessante parlando di **metabolic rewiring (ricablaggio metabolico)** nell’ambito del cancro.

La *FBA* può essere usata in molti modi, con varie applicazioni ed estensioni, come riassunto in figura 9.2³. Tali problemi variano al cambiamento dei pesi e della funzione obiettivo, alcuni di essi sono risolubili in parte numericamente etc... e tutti questi sono aspetti che devono essere studiati e approfonditi di volta in volta.

Dal punto di vista della programmazione lineare si hanno varie categorie di risolutori per il caso continuo ma possiamo tutte racchiuderle in due macro-categorie, considerando comunque che dal punto di vista della biologia computazionale sono usati come **black-box**:

1. **risolutori basati sull’algoritmo del simplesso**, che, come già anticipato, funzionano bene nella maggioranza dei casi anche se si hanno casistiche “patologiche” che rendono il problema esponenziale nel tempo. Nonostante questo, anche perché tali casi sono

³K. Raman and N. Chandra, “Flux balance analysis of biological systems: applications and challenges,” *Briefings in Bioinformatics*, vol. 10, no. 4, pp. 435–449, Jun. 2009.

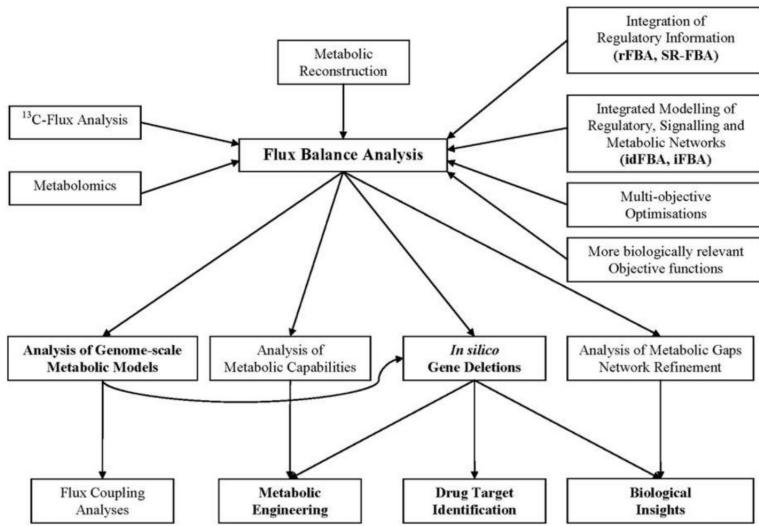


Figura 9.2: Vecchio schema riassuntivo degli usi della *FBA*.

estremamente rari, e grazie alla semplicità di implementazione sono solitamente la soluzione più adeguata

2. **risolutori basati sui metodi del punto interno**, che, a differenza dei risolutori basati sull'algoritmo del simplesso, sono garantiti essere polinomiali. Quest'ultimo aspetto è interessante perché pone i problemi di programmazione lineare continua nella classe di complessità \mathcal{P} e non \mathcal{NP} . Purtroppo tali metodi sono difficili da implementare

Oltre al caso continuo si ha anche quello in cui si introducono vincoli interi, con le variabili che appartengono solamente a \mathbb{N} , avendo così la *programmazione lineare intera*, che sappiamo essere \mathcal{NP} -complete. Si possono inoltre avere risolutori per problemi misti, sia continui che discreti, e questi sono parecchio complessi e sofisticati.

9.3.1 Esplorazione dei Flussi

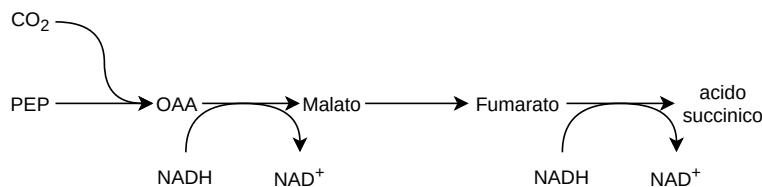
Durante una *FBA* siamo interessati a studiare i flussi ma prima bisogna anche ben capire quali siano quelli di interesse. Si parte da un modello completo, che diciamo appartenere alla categoria dei **Genome Wide Reaction Models**, ci si rende conto di avere davanti qualcosa di molto complesso da studiare. Tali modelli infatti sono:

- biologicamente molto realistici e completi, rappresentando quanto più possibile
- “a grana fine”, con anche 7000 reazioni
- efficaci, per esempio, per l’analisi dei metaboliti o l’analisi della delezione genica
- poco adatti alla quantificazione del flusso

Si procede quindi alla creazione, a partire da tali modelli, dei **Core Reaction Models**. Tali modelli vengono ottenuti semplificando i modelli complessi e “riassumendo” le reazioni, ottenendo quindi modelli sono:

- curati manualmente
- pronti per la simulazione
- “a grana grossa”, con circa 100 reazioni
- utili per la stima della distribuzione del flusso
- utili per l’identificazione dei principi cardine del sistema

In pratica si prende un pezzo di pathway, ad esempio:



e si “ignorano” gli step intermedi, compattando la rappresentazione, semplificando le equazioni e la complessità computazionale, ottenendo:



che presenta quindi, come *steichiometria globale*, solo:

$$-PEP - CO_2 - 2NADH + Succint\ Acid + 2NAD^+ = 0$$

Per scegliere quindi tra i **genome wide models** e i **core models** si utilizza il principio del **rasoio di Occam**, cercando di mediare da un lato per il livello dei dettagli, il costo computazionale, gli errori e gli eventuali cicli termodinamici infattibili e dall'altro mediando lo studio dello spazio di fattibilità, la cura manuale del modello, la “simulabilità” e la visualizzazione del modello stesso. Se il processo della costruzione del **core model** è ben fatto si ottengono con un centinaio di reazioni risultati praticamente analoghi a quelli che si avrebbero con 7000 e passa reazioni.

9.3.2 The Enhanced Growth Model

Studiamo ora, come esempio, il cosiddetto **The ENhanced GROWth (ENGRO) model**⁴.

Tale modello, visualizzabile in figura 9.3⁴, rappresenta il **core model**, quindi curato manualmente, di una crescita migliorata, detta appunto *ENGRO*, che è stato utilizzato per studiare l'*effetto Warburg*, ovvero la produzione aerobica di lattato dal glucosio. Tale effetto descrive l'osservazione che le cellule cancerose e molte cellule cresciute *in vitro* mostrano la fermentazione del glucosio anche quando è presente una quantità sufficiente di ossigeno per respirare adeguatamente. In altre parole, invece di respirare completamente in presenza di ossigeno adeguato, le cellule tumorali fermentano. L'*ipotesi di Warburg* era che l'*effetto di Warburg* fosse la causa principale del cancro. L'attuale opinione popolare è che le cellule cancerose fermentino il glucosio mantenendo lo stesso livello di respirazione che era presente prima del processo di carcinogenesi, e quindi l'*effetto Warburg* sarebbe definito come l'osservazione che le cellule cancerose mostrano glicolisi con secrezione di lattato e respirazione mitocondriale in presenza di ossigeno⁵. Per studiare questo sistema non ci si è limitati ad una singola funzione obiettivo di programmazione lineare ma se ne sono studiate diverse, avendo di fatto un **problema di meta-programmazione**. Si studia quindi tutto il possibile spazio delle soluzioni della *FBA* nonché come si possa perturbare un modello e utilizzare ancora la *FBA* per analizzare la perturbazione.

Studio delle Multiple Soluzioni

In primis vengono quindi studiati i possibili risultati che si ottengono con la *FBA*. Un esempio potrebbe essere quello di voler studiare il tasso di crescita

⁴Damiani et al. "A metabolic core model elucidates how enhanced utilization of glucose and glutamine, with enhanced glutamine-dependent lactate production, promotes cancer cell growth: The Warburg effect." PLOS Computational Biology 13.9 (2017): e1005758.

⁵https://it.wikipedia.org/wiki/Ipotesi_di_Warburg

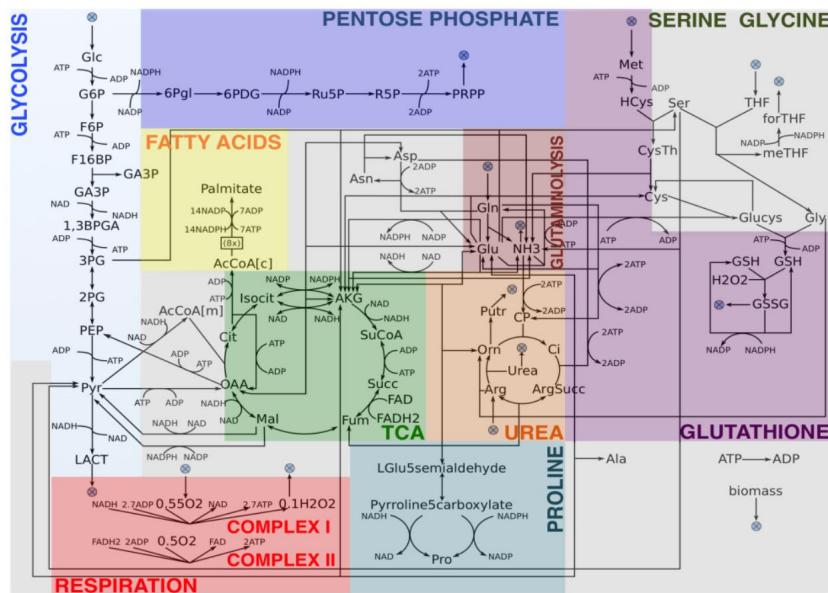


Figura 9.3: Rappresentazione dei pathway della mappa metabolica del modello *ENGR0*

in funzione della disponibilità di nutrienti in ingresso la sistema e cercare l'ottimo, notando come al calare del Glucosio si necessita di più Ossigeno e viceversa.

Un primo studio con il modello *ENGR0* è quindi quello di porre determinati vincoli alle sostanze in ingresso al sistema, ovvero a certi flussi, e studiare, ottica di una certa funzione obiettivo, la distribuzione ottimale dei flussi, vedendo quali sono up-regolati e quali down-regolati.

Ovviamente le soluzioni ottime possono essere differenti e quindi è bene esplorare l'intero spazio delle soluzioni accettabili, variando i vari pesi etc..., facendo anche in modo che certe reazioni si abilitino/disabilitino di volta in volta. Ad esempio si considerino le seguenti condizioni sull'insieme delle soluzioni e su un numero j di interazioni:

$$\begin{aligned} \sum_{i \in NZ_{j-1}} y_i &\geq 1 \\ \sum_{i \in NZ_j} w_i &\leq |NZ_k| - 1, \quad \forall k = 1, 2, \dots, j-1 \\ y_1 + w_i &\leq 1, \quad \forall i \\ \alpha w_i &\leq v_i \leq \beta w_i, \quad \forall i \end{aligned}$$

Ad ogni iterazione j almeno uno dei flussi non nulli provenienti dalla soluzione precedente, ovvero NZ_{j-1} , deve essere impostato a zero, dove la variabile binaria y_i è 1 se quel flusso è selezionato per essere rimosso dalla base all'iterazione j . La variabile binaria w_i viene successivamente forzata a zero se y_i è uno, e i limiti superiore e inferiore per quel particolare flusso sono quindi vincolati a zero.

Le equazioni assicurano che le basi alternative non vengano rivisitate eliminando almeno una variabile diversa da zero trovata nelle iterazioni precedenti. Questo è quindi un algoritmo ricorsivo per il calcolo di ottimi alternativi utilizzando la **programmazione lineare intera mista, mixed integer linear programming (MILP)** e ci permette di enumerare le varie soluzioni alternative.

Studio delle Perturbazioni al Sistema

Un'altra cosa interessante da studiare è come cambi il *fenotipo* al variare di determinate perturbazioni al sistema. Si parte quindi con un **fenotipo wild-type** e gli si applicano vari cambiamenti al fine di ottenere un **fenotipo modificato**. Tra le prime perturbazioni che si applicano si hanno:

- perturbazioni ai nutrienti, che non cambiano la termodinamica del sistema
- delezione di geni, ottenendo le vere e proprie mutazioni

Inoltre si possono fare altri test, meno frequenti, come:

- delezione di reazioni
- shock termico, avendo che, essendo le reazioni sensibili alla temperatura, si può avere la rottura dell'equilibrio del sistema avendo quindi reazioni di tipo diverso da studiare

Dal punto di vista più modellistico/matematico si hanno vari modi di studiare la perturbazione del sistema.

Un primo approccio è usare semplicemente la *FBA*, simulando le risposte metaboliche, ad esempio studiando i *knockout* (che potrebbero essere multipli nel sistema in analisi) e la variazione di nutrienti. Prendiamo quindi un esempio dove, nel caso wild-type, vogliamo massimizzare la crescita. Si ha quindi una certa funzione obiettivo con determinati vincoli, del tipo:

$$\begin{aligned} \max \quad & f_{growth} \\ \text{s.t.} \quad & b_L \leq f \leq b_U \\ & Sf = [0] \end{aligned}$$

Usando semplicemente la *FBA* vengono magari aggiunti dei vincoli, ottenendo:

$$\begin{aligned} \max \quad & f_{growth} \\ \text{s.t.} \quad & b_L \leq f \leq b_U \\ & Sf = [0] \\ & b_L \leq f_{mut} \leq b_U \end{aligned}$$

Dove tali vincoli aggiuntivi magari rappresentano una delle perturbazioni sopra elencate.

Si hanno comunque dei limiti:

- la mutazione potrebbe non crescere in modo ottimale se la selezione naturale non ha avuto la possibilità di agire sul nuovo background genetico
- si è “solo” trovata un’ottimo alternativo, avendo comunque, dal punto di vista della programmazione lineare, soluzioni che si trovano sui vertici

Questa procedura comunque viene normalmente automatizzata, magari usando una lista di geni da attivare/disattivare, tenendo traccia di ogni cambiamento e della variazione di risultati ottenuta tramite esso.

Superando l’approccio classico il primo passo è quello di introdurre una particolare ipotesi biologica, detta **Minimization of Metabolic Adjustment (*MOMA*) hypothesis**, che ipotizza che una mutazione tenderà ad approssimare quando si ha con il wild-type il più possibile. Più formalmente viene trovato un vettore di flusso *MOMA* con distanza euclidea minima da una singola entità wild-type ottimale, soggetta ai vincoli della mutazione. Si ha quindi una certa funzione obiettivo con determinati vincoli, del tipo:

$$\begin{aligned} \max \quad & f_{growth} \\ \text{s.t.} \quad & b_L \leq f \leq b_U \\ & Sf = [0] \end{aligned}$$

e, usando l’ipotesi *MOMA*, si potrebbe arrivare a qualcosa del tipo:

$$\begin{aligned} \min \quad & |m - f_{opt}| \\ \text{s.t.} \quad & b_L \leq f \leq b_U \\ & Sm = [0] \\ & b_L \leq f_{mut} \leq b_U \end{aligned}$$

Ottenendo quindi non un ottimo su un vertice ma la soluzione più vicina, sotto l’ipotesi *MOMA*. Ovviamente anche in questo caso si hanno delle limitazioni:

- l’ipotesi *MOMA* spingerà il metabolismo nel mutante verso la distribuzione arbitraria di flusso ottimale singola ottenuta con la *FBA* ma esistono comunque soluzioni alternative che massimizzano la crescita e altre soluzioni sub-ottimali
- la crescita non è comunque assicurata nelle mutazioni

Un altro approccio è quello più probabilistico, basato su un **sampling randomico** dei vari parametri di programmazione lineare. In questo caso con:

$$\begin{aligned} \max \quad & f_{growth} \\ \text{s.t.} \quad & b_L \leq f \leq b_U \\ & Sf = [0] \end{aligned}$$

Si otterrebbe di volta un volta un certo spazio di soluzioni accettabili per l’entità wild-type mentre con:

$$\begin{aligned} \min \quad & |m - f_{opt}| \\ \text{s.t.} \quad & b_L \leq f \leq b_U \\ & Sm = [0] \\ & b_L \leq f_{mut} \leq b_U \end{aligned}$$

semplicemente si riduce tale spazio.

Si hanno quindi due tipologie di approccio per il *sampling*:

1. **Hit-and-Run (*HR*)**, dove si ha *sampling* uniforme all’interno della regione delle soluzioni permesse, ovvero un punto valido iniziale viene spostato ripetutamente all’interno dello spazio secondo regole probabilistiche. Si ottiene quindi⁶ una funzione obiettivo del tipo, avendo i e j una coppia casuale di reazioni:

$$F = w_i f_i + w_j f_j$$

2. **Convex Basis (*CB*)** dove si usa l’*algoritmo del simplex* con un insieme casuale di funzioni obiettivo da massimizzare. La massimizzazione di ciascuna di queste funzioni obiettivo darà un angolo nello spazio delle soluzioni. In pratica in questo caso il *sampling* è solo sugli spigoli

Ovviamente il numero di combinazioni che si ottengono con il *sampling* può essere veramente enorme quindi deve essere effettuato “sotto controllo”.

⁶Bordel et al., PLoS Comp. Bio. 2010

Un altro elemento interessante è il cosiddetto ***ENGR* Z-score**⁷. Tale valore altro non fa misurare la differenza tra la mutazione e l'entità wild-type, facendo misure su media del wild-type e deviazione standard:

$$Z = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{\sigma_1^2}{n} + \frac{\sigma_2^2}{n}}}$$

Avendo quindi che un valore alto per Z corrisponde ad un'alta differenza tra l'entità wild-type e la mutazione.

9.3.3 Ricerca di Valori Sub-Ottimali

Attualmente lo studio tramite i metodi basati sulla *FBA* presenta diverse limitazioni, anche se alcune estensioni sono possibili per superare alcune di queste:

- si basa sull'assunzione di *steady-state*, comportando difficoltà nello studiare la dinamica del sistema
- non fornisce alcuna informazione sulle concentrazioni dei metaboliti
- obbliga una scelta limitata della funzione obiettivo
- presenta limite nel predire il comportamento netto, ovvero la somma dei flussi, di cellule possibilmente eterogenee all'interno di una popolazione. Questa è una forte limitazione nel caso dei tumori dove si hanno differenti popolazioni e dove non si sa da quale popolazione arrivi una certa produzione

Inoltre ci si è chiesti se il metabolismo funzioni unicamente per massimizzare il tasso di crescita. Si hanno evidenze, studiate in laboratorio⁸ su *Bacillus subtilis*, un batterio, che la delezione di alcuni geni metabolici causano un aumento dei tassi di crescita e della resa di biomassa rispetto all'entità wild-type. La pressione selettiva per aumentare il tasso di crescita deve essere bilanciata da altre richieste sul metabolismo, come il mantenimento cellulare o l'apparato sensoriale, riducendo il tasso di crescita a favore della forma fisica complessiva. Come detto questo si è studiato in laboratorio ma nulla

⁷Damiani et al. "A metabolic core model elucidates how enhanced utilization of glucose and glutamine, with enhanced glutamine-dependent lactate production, promotes cancer cell growth: The Warburg effect." PLOS Computational Biology 13.9 (2017): e1005758.

⁸Fischer, Nat Genet 2005

assicura che la crescita legata alla delezione di alcuni geni possa avvenire anche in natura in quanto magari quel gene codificava per qualche aspetto che era vantaggioso ad altri aspetti della sopravvivenza e senza non si ha possibilità di sopravvivere. Inoltre le limitazioni del tempo evolutivo e della variabilità genetica possono significare che il metabolismo non è ottimale per nessun obiettivo, quindi non possiamo necessariamente escludere dalla considerazione le molte configurazioni di flusso che supportano, ad esempio, il 90% (ma anche percentuali) di crescita massima. Queste configurazioni di supporto tornano comodo in fase di decisione dei vincoli di programmazione lineare.

Il punto chiave del discorso è che magari con una soluzione ottima dal punto di vista del tasso di crescita si porta a massimizzare la biomassa mentre con una soluzione sub-ottimale, sempre lato tasso di crescita, si ottiene non solo una biomassa accettabile ma magari anche, per esempio, la produzione di altri componenti, magari biocarburanti.

Partendo quindi dalla classica:

$$\begin{aligned} \max \quad & f_{growth} \\ \text{s.t.} \quad & b_L \leq f \leq b_U \\ & Sf = [0] \end{aligned}$$

Si hanno tre alternative:

1. aggiungere vincoli, lavorando come già visto con la *FBA* classica, ottenendo una soluzione sub-ottimale che però comporta anche qualche produzione aggiuntiva (come nell'esempio ipotetico quella di biocarburanti)
2. vincolare la biomassa ed esplorare lo spazio della soluzione risultante tramite sampling dello spazio delle soluzioni, avendo quindi, con una configurazione di supporto che porta la biomassa al 90%:

$$\begin{aligned} \text{sampling} \\ \text{s.t.} \quad & b_L \leq f \leq b_U \\ & Sf = [0] \\ & f_{biomass} \leq 0.9 \cdot OPT_{biomass} \end{aligned}$$

Si ha quindi una soglia per il livello minimo di biomassa accettabile, nell'esempio il 90% della biomassa dell'entità wild-type, e si ha come risultato un insieme di soluzioni e non una singola soluzione

3. usare una funzione obiettivo multi-pesata, cambiando quindi la forma stessa della funzione. Si ottiene, ad esempio una cosa del

tipo:

$$\begin{aligned} \max \quad & f_{MW} \\ \text{s.t.} \quad & b_L \leq f \leq b_U \\ & Sf = [0] \\ & f_{biomass} \leq 0.9 \cdot OPT_{biomass} \end{aligned}$$

dove la funzione obiettivo multi-peso, f_{MW} è, ad esempio, del tipo:

$$f_{MW} = 0.8f_G + 0.1f_F + 0.1f_I$$

Quindi non si massimizza il 100% della biomassa ma un certo $(100 - x)\%$, avendo quindi che il sub-ottimo della biomassa viene comunque cercato in uno spigolo, e distribuendo quella percentuale residua x ad altre reazioni

9.3.4 Metabolic Rewiring

Si torna quindi a parlare di cancro.

Possiamo vedere il cancro come una “malattia stocastica”, risultante da una serie di mutazioni che colpiscono le cellule del nostro corpo e comportante la selezione di un fenotipo utile unicamente ai propri scopi. Le cellule cancerogene sono quindi, all’incirca, parte di insiemi di cellule normali che hanno accumulato una serie casuale di mutazioni e modificazioni epigenetiche. Si ha quindi che l’evoluzione cellulare è stata in questi casi in qualche modo modificata e quindi possiamo studiare il normale funzionamento del metabolismo, questa sorta di “metabolic wiring” (“collegamento metabolico”) di quegli insiemi che presentano generiche proprietà che corrispondono statisticamente a quelle delle cellule nella realtà. In termini più matematici riconosciamo l’area di fattibilità tramite programmazione lineare. All’esterno di tale area si hanno i fenotipi impossibili mentre all’interno si ha l’insieme dei fenotipi possibili, insieme in cui si possono riconoscere vari sottoinsiemi caratterizzati da una proprietà simile (che ha un corrispettivo reale nelle cellule). Si assume quindi un metabolismo simile tra cellule normali e cancerogene ma se ne fa uno studio diverso tramite la *FBA*.

Per esplorare lo spazio dei possibili “collegamenti” si procede tramite *sampling* causale dei pesi, cambiando quindi funzione obiettivo ogni volta, e facendo anche *sampling* delle reazioni, puntando infine a massimizzare la somma pesata dei flussi attraverso determinate reazioni. Le distribuzioni di flusso ottenute possono essere una rappresentazione efficiente di una popolazione cellulare mutata casualmente, ad esempio l’attivazione mutata di un gene che codifica per un enzima metabolico corrisponderebbe a un tentativo di aumentare il flusso attraverso quell’enzima.

Dopo aver fatto varie analisi tramite la *FBA* si creano insiemi di diverse risposte metaboliche alle perturbazioni e si procede creando una sorta di albero binario, che si dirama in base alle percentuali di riuscita di determinate azioni. Da tale albero si possono poi ottenere diverse informazioni sul sistema, in quanto le diverse diramazioni permettono di distinguere il comportamento normale da quello cancerogeno.

Tutte queste analisi sono ora usate anche in ambito *single-cell*.

Capitolo 10

Progressione Tumorale

Si introduce ora in modo più dettagliato lo studio della **progressione tumorale** e della modellazione della stessa. Si parla quindi dell'analisi dei dati relativi a tumori con una gran varietà di fini, tra cui la *medicina traslazionale*. Si introducono quindi una serie di tecniche e tecnologie atte al ricostruire possibili modalità/sequenze relative alla progressione tumorale (introducendo anche qualche lavoro fatto in *DCB*, ex *BIMIB*, in *Bicocca*, fatti i collaborazioni con molti altri atenei sparsi nel mondo, da quello di *UCLA* a quello di *Trieste*).

10.1 Approfondimento Biologico

Prima di parlare di vera e propria modellistica bisogna introdurre alcuni concetti biologici essenziali.

Riprendendo la figura 8.1 possiamo riconoscere i vari step della progressione, arrivando fino ad un **adenoma**, ovvero un cosiddetto *polipo intestinale* e al **carcinoma** e questo è visualizzabile ancora meglio, in modo meno stilizzato anche se manca la rappresentazione della **metastasi**, in figura 10.1¹, ovvero nel dettaglio del **Colon Rectal Cancer (CRC)**. Come detto le modalità di progressione sono molteplici e lo studio delle stesse è centrale nel discorso della cosiddetta **medicina personalizzata**. Sempre parlando di *CRC* bisogna anche introdurre il concetto di **staging**². La progressione del tumore avviene infatti in vari step, nel dettaglio del *CRC* se ne individuano tendenzialmente quattro, che portano dallo stato iniziale allo sviluppo ormai incontrollato e di *metastasi* del tumore anche in altri organi. Un esempio è lo *stage 3*, dove si hanno già più tumori all'interno del colon, tumori che si

¹Batlle E. et al. Nat Genet, 39, 1376-83 (2007)

²<https://www.cancer.net/cancer-types/colorectal-cancer/stages>

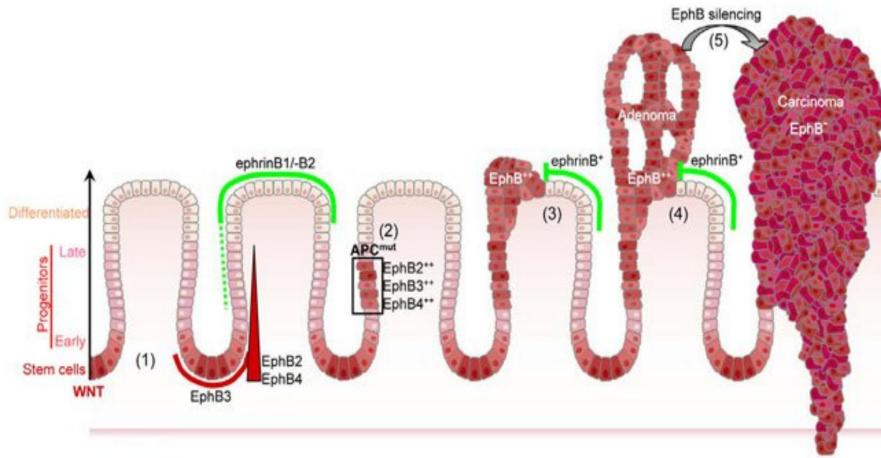


Figura 10.1: Rappresentazione meno schematica di uno dei modelli di progressione tumorale del cancro.

scoprono tramite *colonoscopia*, avendo quindi una **metastasi locale** ma non avendo comunque la **metastasi distante**, ovvero che ormai ha raggiunto anche altri organi. In questo contesto è anche interessante lo studio delle immagini, sia tramite una commissione di tre dottori che stabiliscono l'entità della progressione tumorale, che, in modo più computazionale, tramite modelli di *machine learning* e tecniche di *intelligenza artificiale* per l'analisi di immagini.

Si deve quindi studiare la progressione della malattia e una buona ipotesi sulla maggior parte dei tipi di cancro è che il disturbo progredisce attraverso fasi/stage accumulando alterazioni che influenzano la funzione e le interazioni dei geni. Ad ogni step quindi si accumulano mutazioni etc.... Si giunge quindi ad alcune conclusioni:

- il malfunzionamento di singoli geni non può, da un certo di vista purtroppo, causare il cancro. Questa è stata una speranza anni fa in quanto avrebbe semplificato di gran lunga lo studio di cure per i vari tumori mentre si è arrivati a capire che un tumore deriva da multipli malfunzionamenti di multipli geni
- il cancro si sviluppa attraverso multipli percorsi evolutivi, essendo quindi un problema capire dove effettivamente si sviluppa il tumore, avendo che questo aspetto dipende dai nutrienti etc...
- le alterazioni si possono categorizzare in vari modi, ma tra i principali si hanno:

- **mutazioni somatiche.** Una *mutazione somatica* è una mutazione a livello del DNA di una qualunque cellula del corpo (cellule somatiche) eccezion fatta quindi per le cellule della linea germinale; pertanto essa non viene trasmessa alla discendenza. Nel caso in cui l'elemento colpito sia una cellula ancora in grado di diversi la mutazione viene trasmessa a tutte le cellule che derivano da essa per *mitosi*. In questo caso l'organismo diviene un mosaico cioè sarà costituito da una popolazione di cellule normali ed una di cellule mutate³
- **Copy number variation (CNV)**, ovvero il fenomeno in cui si ripetono sezioni del genoma e il numero di ripetizioni nel genoma varia da individuo a individuo. La variazione del numero di copie è un tipo di variazione strutturale: nello specifico, è un tipo di evento di duplicazione o cancellazione che interessa un numero considerevole di coppie di basi, comportando ad esempio *over-espressione* (ma anche l'opposto) di certe proteine, con conseguenze non sempre positive⁴

Questi, e molti altri, eventi sono dovuti, ad esempio, dal *microambiente* etc..., avendo che le cause per la progressione del tumore possono essere davvero molteplici

Come detto quindi tutto questo studio di modellazione è essenziale per lo sviluppo di farmaco e per decisioni terapeutiche, ad esempio, è noto che per lo stesso tipo di cancro, i pazienti in diverse fasi/stage di diverse progressioni rispondono in modo diverso a diversi trattamenti. Un certo stadio richiede infatti un certo dosaggio di *chemioterapia* a differenza di un altro, avendo quindi che la *quantificazione della terapia* è essenziale in questo contesto medico.

Le cellule cancerogene sono come le altre cellule, da un certo punto di vista, e possono quindi essere studiate come dei *replicatori individuali*. Lo scopo è comunque quello di **riprodursi/replicarsi** e per questo fine tali cellule, ma come quelle normali del resto, devono fare tre cose:

1. *crescere*, attraverso il consumo di energia e tramite i nutrienti etc...

³<https://www.molecularlab.it/principi/dizionario/definizione.asp?w=Mutazione%20somatica>

⁴https://en.wikipedia.org/wiki/Copy_number_variation

2. *vivere a lungo*, nonché *prosperare*, e per questo devono limitare/evitare l'*apoptosi*
3. *muoversi*, e nel caso delle cellule cancerogene questo avviene tramite la **metastasi**

Una cellula cancerosa è quindi essenzialmente una cellula che ha scambiato, a causa di una molteplicità di fattori, una serie di comportamenti, dettati dal suo corredo genetico, per altri, più “di successo” nel suo microambiente, sfortunatamente il concetto di “più di successo” per la cellula è un problema per l’organismo. Si avvicina quindi l’evoluzione tumorale a quanto studiato da Darwin in merito alla **selezione naturale**. Riuscendo a replicarsi, queste cellule danno origine a una nuova popolazione (*sub*)*clonale* che continua ad espandersi, generando così una *crescita neoplastica*, ovvero la crescita anormale delle cellule che porta poi al tumore, avendo quindi una *crescita cancerogena*.

Negli anni, tra le cause di tumori, si sono scoperte anche le cause sterne. In Giappone, anni fa, è stato fatto infatti un esperimento in cui si usava del catrame per “dipingere” le orecchie di alcuni conigli scoprendo che di conseguenza si aveva la formazione di tumori proprio in quelle zone. Questo aspetto aggiunge ulteriore complessità ai vari studi.

Bisogna quindi ora capire quali siano i “segni distintivi” del cancro, i cosiddetti **cancer hallmarks**⁵. Tali *hallmarks* di tale “nuovo” modo di funzionamento, ovvero quello cancerogeno, conferendo alle cellule tumorali forme di vantaggio selettivo, sono stati descritti come segue:

- autosufficienza dal punto di vista dei *segnali* della crescita, ricordando che a livello cellulare i *segnali* sono ricevuti dalle cellule tramite le proteine
- insensibilità ai segnali anti-crescita che cercano di limitare la crescita tumorale incontrollata
- evitare la morte cellulare programmata, evitando l'*apoptosi* segnalata da *p53*
- cercare di raggiungere un potenziale replicativo illimitato
- fare angiogenesi, ovvero il processo che porta alla formazione di nuovi vasi sanguigni da altri vasi preesistenti, sostenuta. In questo modo si comunica al sistema circolatorio di creare vasi sanguigni

⁵Hanahan D, Weinberg RA. The hallmarks of cancer. Cell. 2000 Jan 7;100(1):57-70. doi: 10.1016/s0092-8674(00)81683-9. PMID: 10647931.

nei pressi del tumore in modo da rifornire meglio nutrienti etc... e favorire la crescita incontrollata

- procedere con l'invasione tissutale e la metastasi
- deregolamentare il metabolismo (ed è quanto si studia con la *FBA*)
- eludere il sistema immunitario, che non reagisce solo con l'apoptosi ma anche con molte altre operazioni
- portare all'instabilità del genoma

Tutti questi comportamenti estremizzano i comportamenti che comunque avrebbe ogni cellula per ottenere le tre cose elencate sopra, necessarie alla replicazione: crescere, vivere a lungo e muoversi.

Sempre in merito al confronto con Darwin possiamo notare come si parta, nell'evoluzione tumorale, con una certa cellula, staminale o progenitrice, in un certo microambiente, che si divide, accumulando nei vari microambienti varie mutazioni e creando la situazione tumorale. Un processo analogo, **ad albero**, è quello che Darwin ha osservato per lo studio dell'*evoluzione delle specie*. Un *clone*, nel contesto tumorale, arriva e sopravvive quando è "migliore" delle cellule normali e degli altri cloni, arrivando alla fine, eventualmente, ad avere solo cloni.

10.2 Studio della Progressione

Indicativamente ormai lo studio di queste problematiche avviene tramite la creazione di grafici, come in figura 10.2⁶, in cui comunque si parte dai dati prodotti dal *sequenziamento*. I dati hanno comunque spesso problemi:

- dati mancanti, per varie motivazioni tecniche
- mancanza di *time point* utili. Si hanno infatti spesso *time point* molto distanti e con distanze non uniformi e crescenti, dovuti principalmente al modo in cui i biologi vogliono/possono fare le sperimentazioni in laboratorio (magari all'inizio si hanno check ogni due giorni, poi ogni quattro etc...)

Si possono anche avere grafici manuali (**esempi alla slide 17 della lezione 9**), anche se ormai è più raro.

⁶A. Fischer et al., High-Definition Reconstruction of Clonal Composition in Cancer, Cell Reports 7, 2014

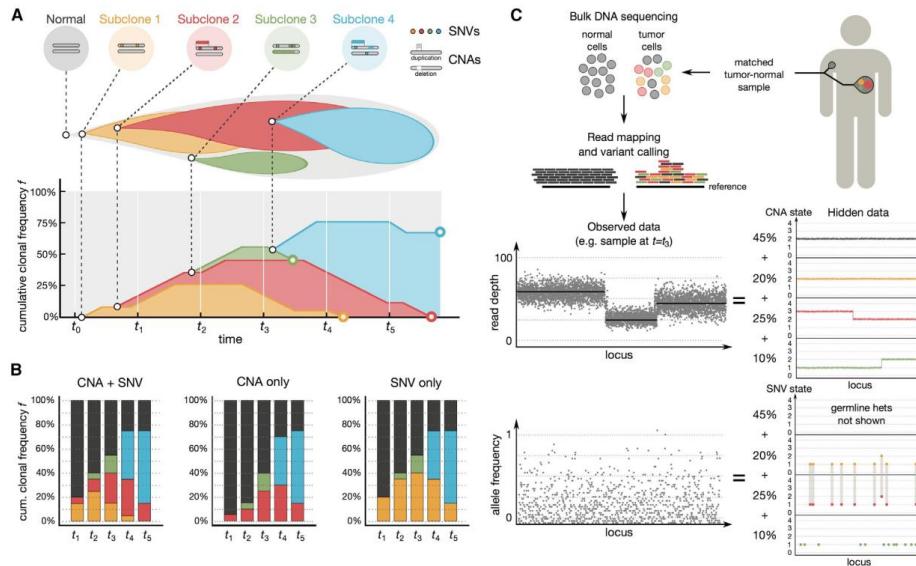


Figura 10.2: Esempio di grafici ottenuti dai dati durante uno studio di progressione tumorale, dove *SNV* sta per *single nucleotide variation* mentre *CNA* per *copy number alterations*.

In generale tali studi possono anche essere utili, con i giusti *time point*, per capire se la *chemioterapia* ha avuto effetto, infatti alcuni ceppi potrebbero sopravvivere al trattamento e potrebbero riportare alla formazione del cancro.

Un’altro aspetto interessante è la **Inter-Tumor and Intra-Tumor Heterogeneity (ITH)**, che è stata già introdotta. La *ITH*, per vari discorsi, è comunque una tematica difficile da trattare e studiare. All’interno di un tumore nello stesso paziente si hanno delle variazioni, come le *eterogeneità spaziali* (ora molto studiate) e le *eterogeneità clonali*, ma si hanno anche variazioni tra lo stesso tumore in diversi pazienti, avendo quindi fare sfaccettature di *eterogeneità*.

10.2.1 Filogenesi

Come già introdotto nel corso di bioinformatica si ha il concetto di **filogenesi**. Date le tecnologie NGS è ora possibile eseguire progetti di sequenziamento che rivelano diversi aspetti di un fenomeno biologico, in primis il cancro.

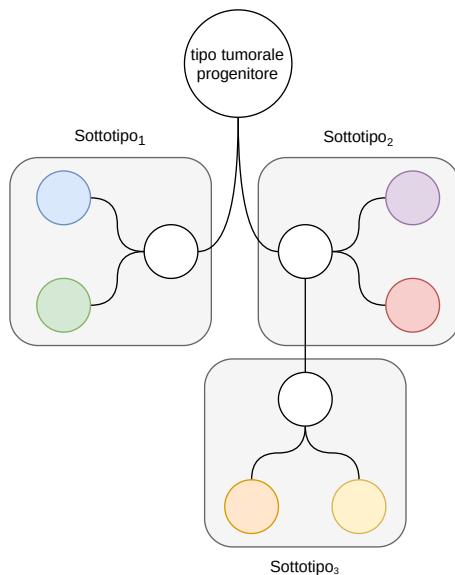
In generale si ha la seguente definizione:

Definizione 7. *Definiamo filogenesi, o albero filogenetico, come un albero che descrive le sequenze di eventi di speciazione che hanno portato alle*

specie attuali, rappresentate nelle foglie di questo albero. I nodi interni non rappresentano specie viventi.

In realtà questa definizione, nello studio dalle sequenze alle informazioni sulle mutazioni nel cancro, può assumere delle varianti⁷:

- parlando di analisi *cross-sectional*, in ambito di *oncogenetica*, si ha un primo studio *inter-patient*, ovvero studiando più pazienti. In questo caso si possono studiare varie tipologie di cancro, partendo da un probabile progenitore plausibile, che sarà la radice dell'albero di filogenesi e arrivando ad identificare vari sottotipi di tumore, provenienti dai vari pazienti:

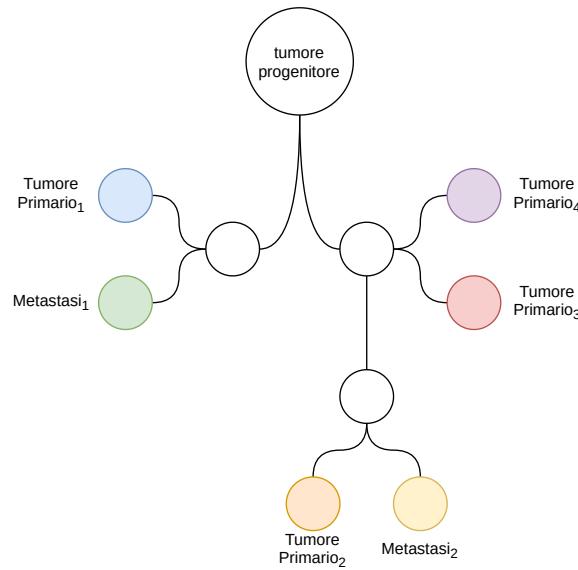


Tali dati sono disponibili nelle varie banche dati, tra cui il *The Cancer Genome Atlas*

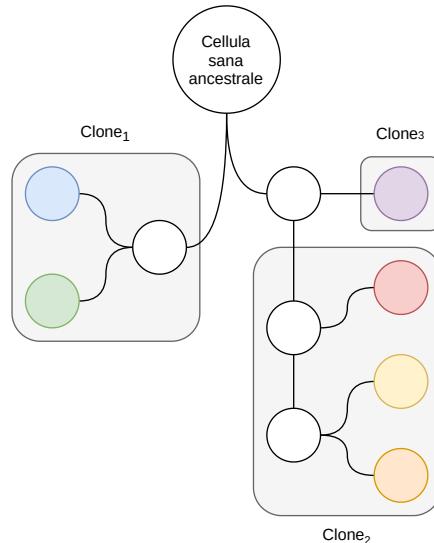
- parlando di analisi *regional bulk*, tornando a parlare dell'individuo singolo e non di una molteplicità di pazienti, si può sequenziare un numero di cellule prelevate da un singolo tumore, facendo slice di tessuto tumorale, facendo poi *bulk sequencing*. In realtà tale analisi si fa usando slice di più tumori relativi a tumori ormai in metastasi in diversi organi. Si arriva quindi a creare un albero

⁷The evolution of tumour phylogenetics: principles and practice, R. Schwartz and A. A. Schäffer, Nature Review Genetics, 2017

che studia l'evoluzione interna del tumore, studiando i vari tumori primari e metastasi derivanti dal progenitore:



- parlando di analisi *single-cell*, schematizzata in figura 10.3⁸ si ha uno studio in merito all'isolamento di singole cellule, producendo una *filogenesi* dei sub-cloni tumorali:



⁸Single-cell genome sequencing: current state of the science, Charles Gawad, Winston Koh & Stephen R. Quake, Nature Reviews Genetics 17, 175–188 (2016)
doi:10.1038/nrg.2015.16

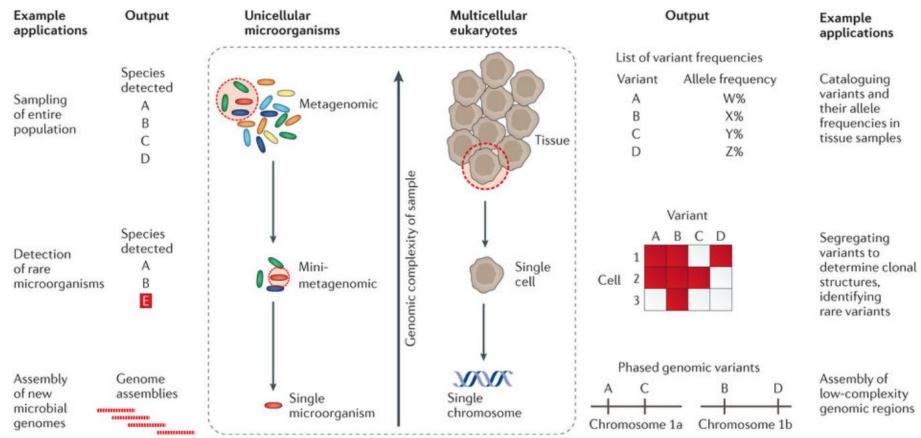


Figura 10.3: Schema riassuntivo di funzionamento della *single-cell analysis*.

Come sappiamo questa è una tecnica innovativa, ad alta precisione non avendo da gestire una moltitudine di “rumori” ma avendo un segnale individuale. Ogni cellula però deve essere sequenziata più volte, portando ad un forte costo (nonostante il singolo sequenziamento ormai sia economico). Bisogna quindi cercare il *tradeoff* migliore tra costi e qualità.

10.3 Cross Sectional Data

Si approfondisce velocemente la tematica della raccolta dati. Si hanno due modalità di studio, due livelli:

1. *ensemble-level*, dove si cerca di ricostruire un insieme di relazioni di precedenza plausibili di eventi a partire da dati dei pazienti, che possono essere anche molti. Questo si fa partendo da n dati *cross-sectional* indipendenti. Possiamo immaginare di avere una matrice con n genomi dei pazienti per le colonne e m “eventi conduttori”. Si ha che $n > m$. Si studiando quindi *SNV* e *CNA*, studiando un albero di filogenesi che approfondisce la selezione e l’eterogeneità
2. *individual-level*, dove si cerca di ricostruire un ceppo di popolazione di cellule tumorali a partire dal prelievo di tessuto tumorale, tramite tecniche filogenesi e single-cell. In questo caso si parte da n campioni dello stesso tumore, ottenuti da una *biopsia*, e si

sequenzia tramite *bulk sequencing*. Si ottiene quindi l'albero evolutivo di un singolo tumore e si studiano il tumore ancestrale e i cloni senza generalizzare la selezione, studiando in modo specifico per il singolo paziente

Si approfondisce quindi l'*ensemble-level*, tramite i *cross-sectional data*, che rappresentano la maggior parte dei dati tumorali disponibili, di contro si hanno pochi dati “longitudinali”.

Tali dati sono raccolti da biopsie al momento della diagnosi e, in altre parole, sono dati di insieme, con pochi “follow-up”, e “time-stamp” (spesso magari raccolti ma non resi disponibili). Tutti questi dati insieme devono essere “riordinati” per ottenere qualche risultato sull’evoluzione tumorale. Dedurre informazioni temporali da tali dati è una sfida aperta (e non solo parlando di cancro ma parlando in generale) e il problema è stato studiato in diversi campi e nel contesto della ricerca sul cancro dalla fine degli anni ’90. Tali dati sono inoltre molto “rumorosi” e spesso vengono gestiti con tecniche di *machine learning* e *reti neurali*. Un altro aspetto fondamentale da considerare è quello dei *dati mancanti*, che complica ulteriormente il discorso.

Esempio di analisi in figura 10.4.

La gestione di tali dati è comunque un lavoro computazionale e si hanno le varie banche dati, come *The Cancer Genome Atlas* con vari portali per accedere semplicemente, tra cui *cBIO portal* o *TCGA data portal*. Un’altra banca dati è *Firehose*, accessibile tramite *FireBrowse*.

10.3.1 Tipologie di Studio

A questo punto si vorrebbe rispondere a tre domande:

1. possiamo ricostruire un modello di progressione da *cross-sectional data*?
2. cosa è effettivamente contenuto negli insiemi di cosa è effettivamente contenuto nella sezione trasversale, essendo ormai cambiati molto negli anni?
3. che tipo di modelli possiamo ricostruire?

Si vede ora quindi come ricostruire modelli di progressione per un insieme di dati, cioè per dati provenienti da più pazienti, usando dei **Direct Acyclic Graph (*DAG*)**, che codificano l’accumularsi delle alterazioni e delle mutazioni, restando quindi sempre in ottica di *dati cross-sectional*.

Una libreria utile in questo contesto è *TRONCO*, sviluppata per l’ambiente *R*, che contiene varie tecniche ed algoritmi per la ricostruzione di modelli di progressione.

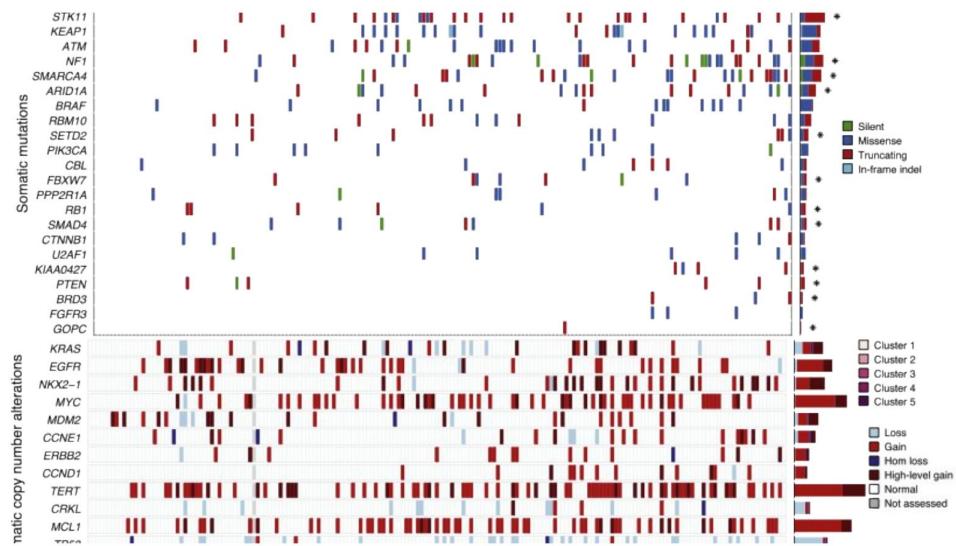


Figura 10.4: Esempio di analisi di mutazioni somatiche e *CNA*, nelle due parti dell'immagine. Ogni colonna rappresenta un paziente.

Primo Quesito

Rispondendo al primo quesito si ha quindi una risposta affermativa, avendo questo schema indicativo, come in figura 10.5⁹:

- si parte da una matrice booleana dove le colonne sono le mutazioni e le righe i tumori, indicando con 1 e 0 la presenza o meno di una certa mutazione in un certo tumore
- si crea una rete causale, inferita dalla matrice, dove si specifica quale mutazione causa. Si ottiene quindi un concetto di “precedenza”
- dalla rete si studia quindi la causalità reciproca tra le mutazioni
- dai risultati si studia la causalità a livello di pathway o si studiano i fenotipi, ovvero gli *hallmark*, avendo quindi livelli ulteriori di astrazione a partire dal generico *DAG*

A partire da questo tipo di modelli si possono ottenere le corrette terapie per curare un paziente.

⁹adattato da: Gerstung et al., PLoS ONE, 6(11), 2011

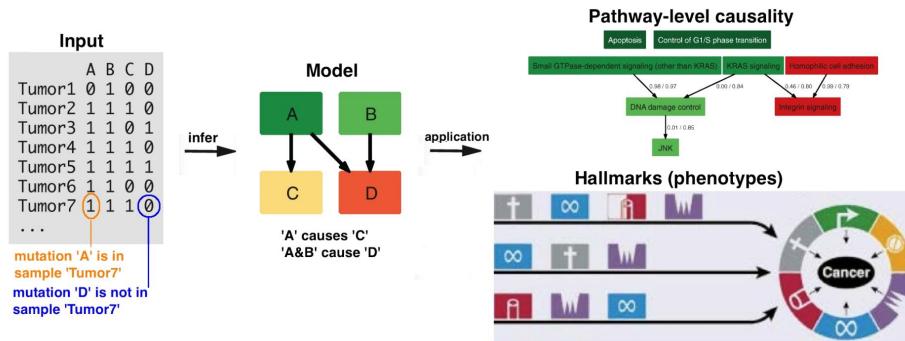


Figura 10.5: Schema rappresentativo di quanto si ottiene a partire da una matrice booleana per studiare la ricostruzione di modelli di progressione.

Secondo Quesito

Per il secondo quesito bisogna citare nuovamente le banche dati pubbliche per i dati relativi al cancro, tra cui:

- The Cancer Genome Atlas (*TCGA*)
- International Cancer Genome Consortium (*ICGC*)

Tutte queste banche dati rispondono al problema della centralizzazione dei dati e delle informazioni, che fino a qualche anno fa erano anche parecchio costosi.

Direttamente collegato è anche il tema centrale della *gestione dei dati sperimentali*, avendo che essi devono:

- essere ben catalogati
- essere facilmente accessibili e studiabili
- garantire interoperabilità
- garantire conformità con gli standard
- etc...

e tutte queste sono tematiche prettamente computazionali.

Ovviamente in ottica di studio del cancro le informazioni più interessanti delle banche dati, tra le tante informazioni disponibili, sono quelle relative alle mutazioni, che sono di solito ereditate e persistenti tra generazioni successive di cellule tumorali (*questa è una sorta di assunzione*). Le mutazioni più interessanti sono solitamente quelle di carattere *somatico*, ovvero le **mutazioni**

somatiche. Inoltre si può dire che La probabilità di insorgenza di una data alterazione/mutazione è correlata a quanto funzionale è allo sviluppo del tumore, avendo quindi che si hanno le mutazioni dette *driver* e quelle dette *passenger*, oltre a diversi tassi di mutazione tumorale (*anche questa è una sorta di assunzione*). Ovviamente anche le mutazioni possono essere di vario tipo:

- i cosiddetti **single nucleotide variant (SNP)**, ovvero mutazioni su singole basi
- le **perdite di eterozigosità**, con perdite di regioni genomiche, perdite di porzioni più piccole, mutazioni copy-number etc...
- intere **delezioni** che siano terminali o interne alla sequenza
- **duplicazioni** di parti di sequenze che possono portare a **tandem**, dove le duplicazioni si appaiano una accanto all'altra, o a duplicazioni dove le porzioni duplicate sono separate tra loro da un'altra porzione di genoma
- **amplificazioni** di porzioni genomiche di tipo **intra-cromosomico** e **inter-cromosomico**
- **duplicazione dell'intero genoma**

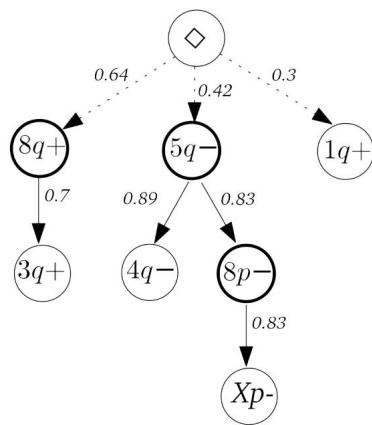
Terzo Quesito

In merito al terzo quesito ci viene incontro la letteratura degli ultimi vent'anni e oltre. Si sono studiati infatti vari tipi di modelli:

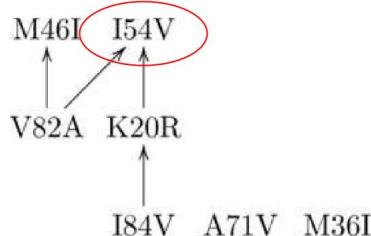
- **modelli ad albero**¹⁰, soprattutto all'inizio quando si avevano dati molto "grezzi" con grosse alterazioni. Tali modelli erano in primis basati sulle correlazioni. Un esempio potrebbe essere il seguente¹¹, dove si ha un esempio di progressione cancerogena (nel dettaglio sul cancro ovarico) rappresentata tramite albero:

¹⁰Desper, Papadimitriou Schäffer et al, 1999, 2000

¹¹Olde Loohuis et al, 2014, originally from Desper et al, 1999



- **modelli congiunti**¹², basati principalmente su modelli Bayesiani. Un esempio potrebbe essere il seguente¹³ ottenuto per progressione congiuntiva ricostruita da un set di dati sulla risposta ai farmaci dell'HIV:



- **modelli DAG**, che sono una generalizzazione dei due modelli precedenti

10.3.2 Algoritmo CAPRI

Quanto trattato in merito a **TRONCO**, **CAPRESE** e **CAPRI** è ritrovabile nei seguenti articoli:

- D. Ramazzotti, G. Caravagna, L. Olde-Looijhuis, A. Graudenzi, I. Korsunsky, G. Mauri, M. Antoniotti, and B. Mishra, “CAPRI: Efficient Inference of Cancer Progression Models from Cross-sectional Data.,” Bioinformatics, p. btv296, May 2015.

¹²Beerenwinkel, Sturmfelds et al, 2005, 2006, 2007

¹³Beerenwinkel et al, 1999

- L. O. Loohuis, G. Caravagna, A. Graudenzi, D. Ramazzotti, G. Mauri, M. Antoniotti, and B. Mishra, “Inferring Tree Causal Models of Cancer Progression with Probability Raising,” PLoS ONE, vol. 9, no. 10, p. e108358, Oct. 2014.
- L. De Sano, G. Caravagna, D. Ramazzotti, A. Graudenzi, G. Mauri, B. Mishra, and M. Antoniotti, “TRONCO: an R package for the inference of cancer progression models from heterogeneous genomic data.,” Bioinformatics, p. btw035, Feb. 2016.
- G. Caravagna, A. Graudenzi, D. Ramazzotti, R. Sanz-Pamplona, L. De Sano, G. Mauri, V. Moreno, M. Antoniotti, and B. Mishra, “Algorithmic methods to infer the evolutionary trajectories in cancer progression,” Proc. Natl. Acad. Sci. U.S.A., pp. 201520213–10, Jun. 2016.

Si approfondiscono ora alcune delle tecniche sviluppate nel laboratorio di ricerca *BIMIB*, ora *DCB*:

- in primis è stato sviluppato l'algoritmo detto **CAncer PRogression Extraction with Single Edges (CAPRESE)**, per la modellazione tramite alberi
- si è poi lavorato sull'algoritmo **CAncer PRogression Inference (CAPRI)**, un'estensione di *CAPRESE* per lavorare tramite modelli *DAG*. Approfondiremo soprattutto questo
- questi algoritmi, più altre funzionalità di supporto, come l'accesso a *cBIO* e *TCGA*, sono state raccolte nella libreria per il linguaggio *R* detta **Translational ONCOlogy (TRONCO)**. L'algoritmo in grado di ricostruire un *DAG* rappresentativo delle possibili progressioni di un tumore, intese come accumuli di eventi diversi, il *DAG* prodotto è un riassunto delle possibili progressioni osservabili in una popolazione

partiamo da un esempio molto sintetico. Si supponga di voler modellare una progressione tumorale in cui avvengono in primis due mutazioni, una detta *EGFR* e, poi, una detta *CDK*. Si devono fare due importanti assunzioni, abbastanza forti per di più, per poter procedere con la modellazione:

1. **assunzione di persistenza**, ovvero le mutazioni acquisite non scompaiono (*e questo non è vero per le espressioni geniche e gli*

effetti epigenetici). Si ha quindi, riprendendo l'esempio, che con la mutazione *EGFR* si acquisisce un vantaggio selettivo e si procede quindi con l'espansione clonale, aumentando anche la probabilità di acquisire una mutazione *CDK*. Quando si hanno entrambe le mutazioni si ha una specie selettivamente ancora più avvantaggiata

2. **assunzione di selezione di eventi**, ovvero gli eventi rilevanti per la progressione devono essere scelti in anticipo. In pratica, riferendoci all'esempio, si sa già che bisogna studiare un modello con in input le mutazioni *EGFR* e *CDK*. Limitarsi a poche combinazioni di eventi rende la ricostruzione computazionalmente più fattibile, non usando quindi l'intera moltitudine di dati provenienti dagli studi oncologici. Si fa quindi una sorta di *studio supervisionato*, che nel dettaglio viene fornito da vari tool ormai standard atti soprattutto a scovare le mutazioni *driver*

Un altro aspetto fondamentale in questo studio è che ogni paziente ha una sua storia di evoluzione e progressione cancerogena diversa. L'effettivo input all'algoritmo di ricostruzione della progressione può essere pensato come un insieme di possibili traiettorie, cioè di sequenze di alterazioni genomiche che si accumulano in modo diverso, bisogna quindi procedere con il conteggiare e assegnare le giuste probabilità ad ogni singola occorrenza di un evento. Si sono sviluppati vari modi negli anni per ricostruire modelli ad alberi o DAG e utti gli algoritmi necessitano di una misura per decidere se e come includere un arco nella ricostruzione. Vediamo quindi come si è scelto di procedere in *BIMIB/DCB*, che si basa sulla **teoria della causalità probabilistica**.

Un primo punto cardine considerato è infatti quello della **teoria della causalità** di Suppes, un filosofo e matematico. In biologia bisogna comunque considerare che aggiungere dei concetti di causalità complicano molto i modelli, anche se con questa teoria si riesce ad ottenere in modo implicito l'idea di **direzionalità temporale** nei modelli. Il nucleo di questa teoria è molto semplice. Dati due eventi c ed e , occorrenti rispettivamente al tempo $t : c$ e t_e , con:

$$0 < P(c) \text{ e } P(e) < 1$$

si ha che c è detto **causa prima facie** per e se si rispettano due proprietà:

1. la **proprietà di priorità temporale**, ovvero banalmente:

$$t_c < t_e$$

che garantisce un ordine temporale tra gli eventi

2. la **proprietà della crescita delle probabilità**, che ci dice che la probabilità che l'evento e accada essendo accaduto l'evento c sia maggiore (e preferibilmente molto maggiore) di quella in cui non si ha l'evento c :

$$P(e|\neg c) < P(e|c)$$

Queste due proprietà, **necessarie ma non sufficienti**, garantiscono la direzionalità del modello.

Oltre alla teoria di Suppes si ha anche che sia le misurazioni che le analisi teoriche vengono reinterpretate in termini biologicamente plausibili, eventualmente anche ridefinendo le relazioni di vantaggio selettivo. Queste scelte permettono di racchiudere maggiori informazioni negli algoritmi proposti da *BIMIB/DCB* di ricostruzione, che mostrano ottime prestazioni rispetto allo stato dell'arte, ora spesso basati su modelli Bayesiani o sui modelli causali di Pearl.

Approfondiamo quindi meglio le basi per la ricostruzione del modello di progressione.

Le già anticipate cause *prima facie* possono essere di due tipologie:

- **autentiche/genuine**, quelle “corrette”
- **spurie**, quelle “errate”

e per questo le due condizioni risultano solo sufficienti e non necessarie. Inoltre, parlando di dati *cross-sectional*, la priorità temporale tra gli eventi non è nota, quindi il problema della ricostruzione diventa più difficile. Il problema diventa quindi come stabilire se un dato arco tra due eventi corrisponda o meno a una causa spuria, e quindi non vada considerato, avendo comunque che l'interpretazione di un arco sarà quella di una relazione di vantaggio selettivo tra eventi genetici.

Inoltre un'ulteriore complicazione è data dal fatto che nulla assicura che si parli solo di due eventi ma si possono avere combinazioni complesse di causalità tra eventi e per questo nell'algoritmo *CAPRI* queste situazioni vengono gestite tramite **formule booleane**, come le formule congiuntive del tipo utilizzato nelle reti Bayesiane congiuntive. Si distinguono quindi:

- **singleton**, dove una mutazione accade prima di un'altra e si ha relazione causale
- **co-occorrenze**, dove più di una mutazione accade prima di un'altra e si ha relazione causale. Ovviamente in questo caso la complessità aumenta

Nell'algoritmo *CAPRI* si ha anche una condizione finale, ovvero la condizione per la selezione vantaggiosa tramite *patterns* (indicando con \triangleright la corretta causalità):

$$c \triangleright e \iff P(c) > P(e) \wedge P(e|c) > P(e|\neg c)$$

Si parte quindi da un modello causale con i vantaggi selettivi, si fanno le osservazioni sulle *regolarità imperfette* e si fanno studi di frequenza. Più nel profondo l'inferenza dei risultati viene poi ottenuta, ad esempio, tramite tecniche statistiche come:

- **bootstrap**, una tecnica statistica di ricampionamento con reimmissione per approssimare la distribuzione campionaria di una statistica. Permette perciò di approssimare media e varianza di uno stimatore, costruire intervalli di confidenza e calcolare *p-value* di test quando, in particolare, non si conosce la distribuzione della statistica di interesse
- **test del *p-value*** e altri test

Si ottengono quindi relazioni causali tra mutazioni, eventualmente anche con l'effetto co-occorrente (e quindi senza avere interesse sull'ordine) di più mutazioni che causano una terza mutazione. Ad esempio se si ha una mutazione *A* e una mutazione *B* che causano in modo co-occorrente la mutazione *C* si ha, dal punto di vista statistico:

$$\min\{P(A), P(B)\} > P(C) \wedge P(C|A, B) > P(C|\neg(AB))$$

Per pattern complessi si è studiato sperimentalmente come l'algoritmo *CAPRI* che è in grado di capire la causalità tra varie mutazioni e rappresentarle tramite formule booleane. La complessità del modello poi si riconduce alla complessità di tali formule, ottica dei vari studi teorici sulla complessità di *SAT* etc....

Le formule booleane sono della forma **conjunctive normal form (CNF)**, ovvero una congiunzione di clausole, dove le clausole sono una disgiunzione di letterali.

In ottica di limitare la complessità dobbiamo quindi evidenziare le relazioni *autentiche* più significative in quanto l'algoritmo *naive* individuerebbe anche transitività, sotto-formule e altri archi topologici come possibili relazioni di selettività. Un esempio è mostrato in figura 10.6.

L'algoritmo *CAPRI* è una combinazione di sei passaggi che sono responsabili della produzione di un insieme di relazioni di selettività e della successiva pulizia del modello generato:

1. una prima fase gestione dei dati in input

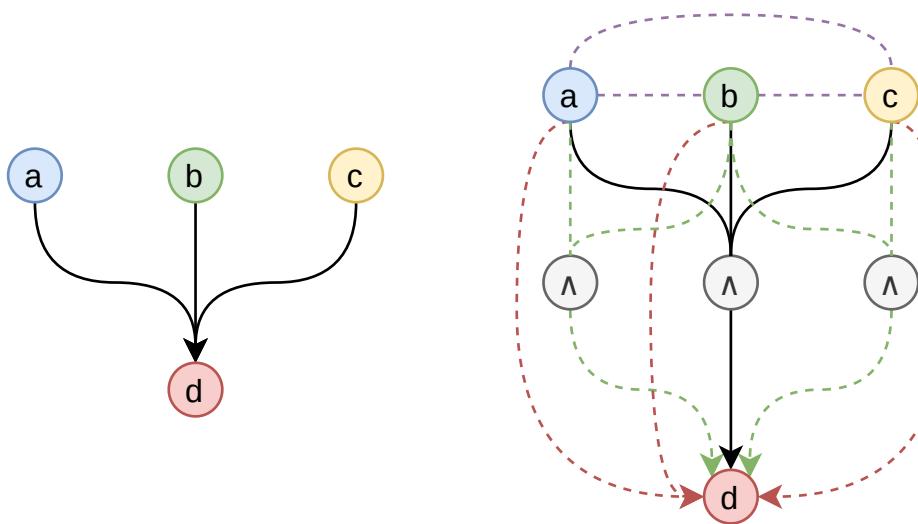


Figura 10.6: Test case di confronto tra una progressione reale a sinistra (dove i collegamenti sottintendono la congiunzione) e una selezione co-occorrente a destra per rappresentare la formula $a \wedge b \wedge c \triangleright d$. Con le frecce piene si rappresentano le relazioni “reali”, quelle autentiche, con le frecce tratteggiate rosse e verdi, rispettivamente, transitività e sotto-formule, che sono una via di mezzo tra relazioni autentiche e relazioni spurie ma comunque eliminabili, mentre con le righe tratteggiate viola gli archi topologici, che sono la massima rappresentazione di relazioni spurie.

2. una fase di preprocessing con aggiunta di informazioni, sotto forma di *CNF*, detta **lifting**, al modello grezzo ottenuto dai dati. Si ha un insieme G di n eventi provenienti da m samples *cross-sectional*, avendo quindi una matrice booleana $n \times m$. Si assume in questa un insieme di ipotesi:

$$\Phi = \{\phi_i \triangleright e_i \mid 1 \leq i \leq k\}, \text{ con } k + n \ll m$$

Ogni singolo $\phi_i \triangleright e_i$ è usato per accresce la matrice in input ottenendo una matrice $D(\Phi)$ che codifica le relazioni di selettività opzionali come eventi

3. una fase di selezione dei nodi del *DAG*
4. una prima fase di selezione degli archi del *DAG*
5. si ha poi una fase di etichettamento degli archi. In questa fase e relazioni di selettività codificate sono formule *CNF*, quindi possono essere trattate in modo composito verificando separatamente ogni congiunzione (ovvero una disgiunzione).

L'algoritmo include nella costruzione archi tra due eventi c e e sse:

$$P(c) - P(e) > 0 \wedge P(e|c) - P(e|\neg c) > 0$$

e questa è la condizione principale. Si noti che le probabilità sono stimate dai dati direttamente, tramite un primo step di *bootstrap*. Il *DAG* ricostruito contiene tutte le corrette relazioni di selettività ma anche parecchie spurie, che non vengono ancora eliminate. Ogni arco è etichettato con una probabilità che è essenzialmente la probabilità di osservare un certo profilo mutazionale in un campione.

Già alla fine di questa fase il modello ricostruito è interpretabile dal punto di vista grafico, tramite varie librerie esterne

6. si ha infine una fase di *likelihood fit*. Ricordando che le assunzioni di Suppes sono solo necessarie per “filtrare” le relazioni spurie, ovvero i falsi positivi che sono stati inclusi nei passaggi precedenti, calcoliamo un *fit di massima verosimiglianza* che include un termine di regolarizzazione, termine che può essere scelto e calcolato in modo diverso. Ad esempio come criteri si hanno attualmente nell’algoritmo *CAPRI*:

- **Bayesian Information Criterion (*BIC*)**, più conservativo

- Akaike Information Criterion (**AIC**)

questa scelta permette quindi vari *tradeoff*.

Infine si usa un passaggio di *bootstrap* per dedurre gli intervalli di confidenza per ciascuna relazione di selettività dedotta.

L'*algoritmo CAPRI* (ma anche in primis l'*algoritmo CAPRESE*) è **corretto** e **completo**, avendo che sono riportate tutte e sole le vere relazioni di selettività desumibili dai dati. Dal punto di vista computazionale invece si nota come gli step più onerosi sono la fase di preprocessing dei dati, ad esempio il parsing dei file *MAF*, e le due fasi di *bootstrap*.

Va sempre ricordato che le relazioni di precedenza che l'algoritmo deduce non spiegano le cause meccanicistiche biochimiche di un modello di progressione, infatti l'algoritmo fa solo un'affermazione sul fatto che l'osservazione di un dato insieme di mutazioni aumenta la probabilità di vederne uno successivo (**questa osservazione è fondamentale**).

Un altro vantaggio dell'*algoritmo CAPRI* è la gestione del *rumore*, migliore di altre soluzioni allo stato dell'arte. Si hanno comunque molti altri punti di forza rispetto ad altre soluzioni, tra le quali si annoverano:

- **Incremental Association Markov Blanket (IAMB)** e **PC**, per quanto riguarda soluzioni strutturali
- **Bayesian Information Criterion (BCI)** e **Bayesian Dirichlet (BDE)**, per quanto riguarda la soluzioni basate sulla verosimiglianza
- **Conjunctive Bayesian Networks**, per quanto riguarda soluzioni ibride

Su slide lezione 10 alle pagine 39 e 40 vari grafici per la misura delle performance di *CAPRI*.

10.3.3 Usi reali di TRONCO e CAPRI

L'uso di *TRONCO*, e quindi dell'*algoritmo CAPRI* è stato usato, ad esempio per:

- lo studio della **leucemia**, nel dettaglio della **Atypical Chronic Myeloid Leukemia (aCML)**, a partire dai dati di 64 pazienti studiati con il dipartimento di Medicina della Bicocca
- lo studio del cancro al colon, il **Colorectal cancer (CRC)**, in collaborazione con l'università di Barcellona e usando dati di circa 200 samples presi da *TCGA*

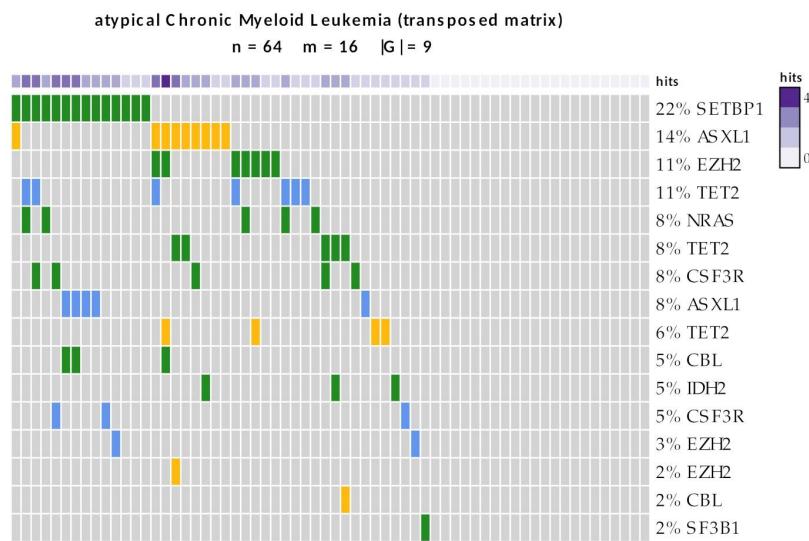


Figura 10.7: Immagine ottenuta tramite la libreria *oncoprint* con una rappresentazione delle analisi fatte per *aCML*. Ogni colonna rappresenta un paziente/sample mentre ogni riga una mutazione. Le mutazioni sono ordinate in ordine decrescente di incidenza.

Il caso Reale dello Studio della aCML

In questo caso si è considerato un set di 64 pazienti con aCML per i quali una *mutazione puntiforme missense*, un tipo di mutazione puntiforme in cui un aminoacido diverso è posto all'interno della proteina prodotta, diverso da quello originale, ricorrente della proteina **SET-binding 1 (*setbp1*)**¹⁴. L'analisi ha considerato:

- geni selezionati in modo che fossero mutati in almeno il 5% dei pazienti
- geni selezionati che si ipotizzava facessero parte di un pattern funzionale di progressione aCML (come indicato in letteratura)

In merito a questo studio sono stati formulati due rigidi vincoli di esclusività:

1. l'esclusività tra le mutazioni *ASXL1* e *SF3B1*, rappresentabile tramite la formula:

$$(\text{ASXL1 Nonsense point} \oplus \text{ASXL1 Ins/Del}) \oplus \text{SF3B1 Missense point}$$

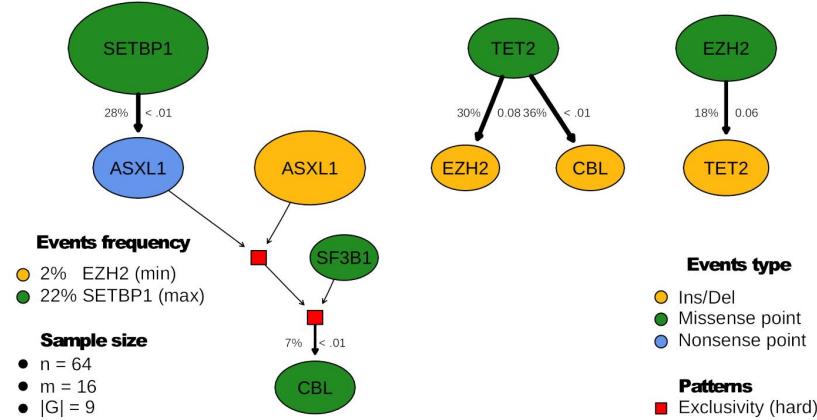
¹⁴Piazza, R., et al. (2013). Recurrent *setbp1* mutations in atypical chronic myeloid leukemia. *Nature genetics*, 45(1), 18–24.

2. l'esclusività tra le mutazioni *TET2* e *IDH2*, rappresentabile tramite la formula:

$$(\text{TET2 Nonsense point} \oplus \text{TET2 Missense point}$$

$$\oplus \text{TET2 Ins/del}) \oplus \text{IDH2 Missense point}$$

Ottenendo i seguenti risultati, direttamente tramite *TRONCO*:



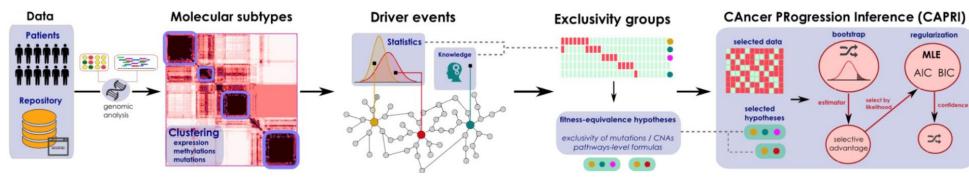
Alla fine l'*algoritmo CAPRI* trova la seguente, tra le altre, relazione di selettività:

$$(\text{ASXL1 Nonsense point} \oplus \text{ASXL1 Ins/del}) \oplus \text{SF3B1 Missense point} \triangleright \text{CBL Missense point}$$

10.3.4 Analisi Cancro al Colon Via PiCnIc

Dopo aver visto l'analisi reale dei dati della *aCML* si illustra brevemente uno studio fatto sui dati del *cancro al colon*, appunto il **Colorectal Cancer Analysis (CRC)**, soprattutto in ottica di presentazione della **Pipeline for Cancer Inference (PiCnIc)**, il cui schema generale è visualizzabile in figura 10.8¹⁵, una serie di script in *R*, associati alla già presentata libreria *TRONCO* per l'analisi di dati tumorali tramite una specifica pipeline. Dal punto di vista dei dati per il *CRC* viene usato come dataset il *COADREAD 2012 TCGA dataset*, contenente sempre dati *cross-sectional*. Tale dataset è già incluso in *TRONCO*.

¹⁵ Giulio Caravagna, Alex Graudenzi, Daniele Ramazzotti, Rebeca Sanz-Pamplona, Luca De Sano, Giancarlo Mauri, Victor Moreno, Marco Antoniotti, and Bud Mishra (2016) Algorithmic Methods to Infer the Evolutionary Trajectories in Cancer Progression. PNAS.

Figura 10.8: Schema generale di *PiCnIc*

Come detto gli *algoritmi CAPRI e CAPRESE* sono usati sia in condizioni di studio individuali (nel caso detto appunto *individual*) che di studi di molteplici pazienti (nel caso detto appunto *ensemble*).

All'interno di questa pipeline sono riconoscibili cinque passaggi che verranno approfonditi a breve. La pipeline produce *modelli grafici probabilistici* restituendo le relazioni di ordinamento temporale e dipendenza statistica tra le *alterazioni driver* che si accumulano in quello specifico sottotipo, quindi l'inferenza del modello di progressione fornisce una caratterizzazione delle tendenze di accumulo più probabili delle *alterazioni del driver* alla base di ciascun sottotipo di tumore. Si noti che la definizione dei sottotipi si può ottenere tramite studi di letteratura o tecniche software, magari basate su immagini. Con la pipeline, nel caso del *CRC*, otteniamo risultati relativi ad una popolazione di pazienti, risultati che devono comunque subire una forte validazione medica specialistica.

L'idea di massima è quindi quella di fornire all'*algoritmo CAPRI*, l'ultimo step della pipeline *PiCnIc*, un input elaborato costituito da dati *cross-sectional (epi)genomici* di singoli pazienti oncologici, stratificato in diversi sottotipi, da studiare nel complesso.

Input Data

I dati per la pipeline vengono presi dai soliti *TCGA*, *FireBrowse* etc... e sono dati relativi, principalmente, alle mutazioni e alle informazioni relative al *copy-number*.

I dati in input servono a creare la *matrice booleana*, contenente i profili mutazionali *cross-sectional*. Come già spiegato più volte tale matrice booleana indica tramite 0 e 1, rispettivamente, la presenza e assenza di:

- **single nucleotide variation (SVN)**
- **copy number alterations (CNA)**
- **fusioni**
- **varianti strutturali**

- **dati epigenetici**, come, ad esempio, dati di metilazione (ovvero l'addizione o la sostituzione di un gruppo metile su vari substrati, catalizzata da enzimi) ed espressione, ma solo se persistenti, per *l'assunzione di persistenza*

Dal punto di vista dei formati file la pipeline *PiCnIc* accetta:

- **file Mutation Annotation Format, MAF**. tali file sono file delimitati da tabulazioni che contengono annotazioni di mutazioni somatiche e/o germinali. I file MAF contenenti eventuali annotazioni di mutazione germinale sono protetti e distribuiti nella parte ad accesso controllato del *GDC Data Portal*. I file MAF contenenti solo mutazioni somatiche sono disponibili pubblicamente¹⁶
- **file GISTIC**, contenente i risultati delle *copy-number analysis*

Inoltre, entrando nei dettagli numerici del dataset *COADREAD 2012*, si hanno:

- 15995 mutazioni somatiche provenienti da 224 pazienti
- 10184 acquisizioni di alto livello (*high-level gain*) provenienti da 564 pazienti
- 8174 delezioni omozigote provenienti da 564 pazienti

Subtyping

Per evitare gli effetti confusionali delle popolazioni eterogenee, i campioni dovrebbero essere stratificati in sottotipi molecolari omogenei, si procede quindi con la fase di riconoscimento dei sottotipi.

Bisogna notare che in alcuni dataset, tipo quello relativo al *CRC*, tale lavoro era già incluso ma non è sempre così. Ignorare che il lavoro sia già stato fatto, inoltre, può comportare risultati disastrosi sulla pipeline, per questo bisogna sempre leggere la documentazione.

Nel dettaglio, per il *CRC*, si conoscono due sottotipi molecolari:

- **microsatellite instability (MSI)**
- **microsatellite stability (MSS)**

¹⁶https://docs.gdc.cancer.gov/Encyclopedia/pages/Mutation_annotation_Format/

Questa fase di riconoscimento dei sottotipi si basa prettamente su tecniche di *clustering*:

- *mutation clustering*
- *methylation clustering*
- *expression clustering*

Driver Selection

Partendo dall'idea che con il termine *evento* si intende un qualsiasi fenomeno inferibile dalla matrice booleana in input si ricercano quelli principali, gli *eventi driver*, che saranno quelli indicati nei nodi del *DAG* finale della pipeline.

Per ogni sottogruppo, deve essere selezionato un elenco di eventi driver putativi, in vari modi:

- sfruttando la conoscenza biologica a priori, tramite letteratura, oncologi, paper su *TCGA* etc...
- tramite software, tra cui *Dendrix*, *MUTEX*, *MutsigCV* etc...
- database specifici come *COSMIC*, il catalogo delle mutazioni somatiche nel cancro

In questa fase si ricercano vari fenomeni e tra i principali si identificano:

- inattivazioni dei geni di soppressione dei tumori, **Tumor Suppressor Genes (TSG)**
- amplificazioni oncogene ad alto livello
- mutazioni oncogene

Mutual Exclusivity Relations

Questa fase è opzionale.

La pipeline *PiCnIc* può testare modelli di mutua esclusività, tramite funzioni booleane con *or* o *xor*, al fine di indagare la possibile esistenza di traiettorie evolutive *fitness-equivalent*. Anche in questo caso per riconoscere tali pattern si hanno varie soluzioni:

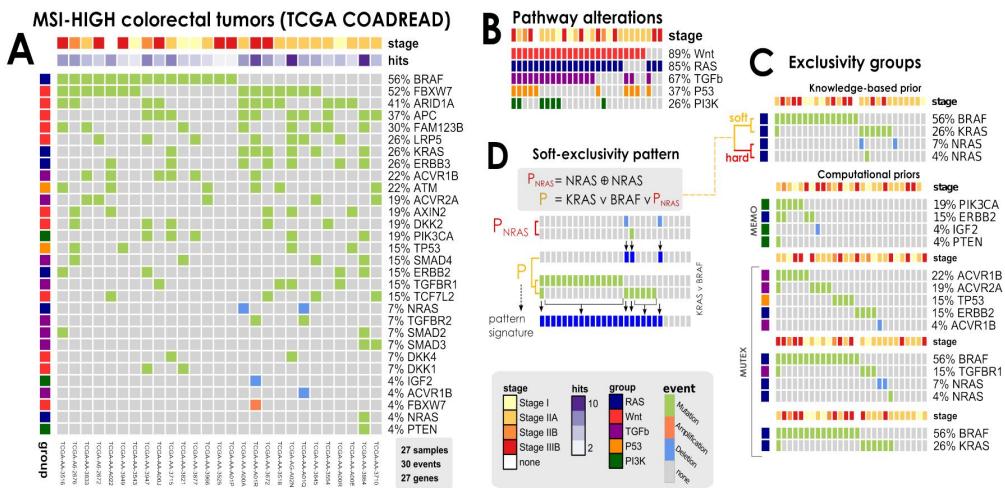


Figura 10.9: Varie stampe della funzione *oncoprint* dei vari stage. Si noti, nella figura *B* come il *pathway Wnt* e il *pathway RAS* siano i più alterati. Si noti anche, nella figura *D*, la creazione delle formule booleane.

- sfruttando la conoscenza biologica a priori, tramite letteratura, oncologi, paper su *TCGA* etc... andando ad analizzare alcuni pathway, tipo il *pathway RAS*, un dei pathway principali che è usato per trasdurre i segnali intracellulari in risposta ai mitogeni per controllare la crescita cellulare, la sopravvivenza e i programmi anti-apoptotici. Tale pathway è quindi fortemente legato alla proliferazione. Un altro pathway che viene spesso impattato è il *pathway Wnt*, legato alla trasduzione del segnale attraverso proteine che trasmettono il segnale dall'esterno della cellula, attraverso recettori di superficie, all'interno della cellula
- tramite software, tra cui *MEMO*, *MUTEX* etc...

Infine per ogni gruppo si identifica un particolare formula. In ogni caso si ricorda che i risultati ottenuti in questo passaggio (ma anche in quello precedente) sono puramente inferiti dai dati e non sono risultati meccanicistici.

Step Finale

Nello step finale si usa, come già anticipato, l'*algoritmo CAPRI*.

L'algoritmo viene fatto girare scegliendo opportunamente i termini di regolarizzazione e gli altri parametri, ovvero gli step di *bootstrap* (solitamente un centinaio), le soglie di *p-values* etc...

Alla fine della pipeline *PiCnIc* si avrà quindi un *DAG*, che ha comunque un fondamento biologico ma necessita comunque di un'interpretazione medica, dove:

- ogni nodo rappresenta un *evento driver*
- ogni arco rappresenta una relazione di ordinamento temporale
- i gruppi di esclusività sono visualizzati in “forma estesa”, ovvero in modo semplificato

Il modello finale descrive le tendenze più probabili dell'accumulo degli *eventi driver*, rispetto a quello specifico sottogruppo/sottotipo di cancro, avendo quindi che ogni relazione nel modello potrebbe essere interpretata come una relazione di vantaggio selettivo tra due *eventi driver*. In termini pratici significa che la presenza dell'alterazione *X* aumenta le probabilità che il tumore acquisisca in seguito anche la mutazione *Y*, dato l'input. Ovviamente le mutazioni successive potrebbero essere non solo *Y* ma anche molteplici.

Una rappresentazione di uno studio è visualizzabile in figura 10.10. Quindi per utilizzare effettivamente l'*algoritmo CAPRI* in modo corretto, è necessario procedere alla pre-elaborazione dei dati, utilizzando anche una serie di strumenti “pronti all’uso” per eseguire una serie di passaggi di stratificazione e filtraggio necessari e a questo serve la pipeline *PiCnIc*.

10.4 Individual Data

Finora si è parlato di *cross-sectional data*, parlando di dati *ensemble*. Si introduce ora la tematica relativa agli **individual data**, parlando di ricostruzione di modelli di progressione tumorale nel caso di singoli tumori, provenienti dallo stesso paziente. A tale fine si studiano varie strutture dati:

- alberi di filogenesi (*phylogenetic trees*)
- alberi clonali (*clonal trees*)
- alberi e grafi mutazionali (*mutational trees and graph*)

Si introduce anche un sottomodulo della libreria *TRONCO*, chiamato **TRaIT** (*Temporal oRder of Individual Tumors*)¹⁷, atto appunto a inferire alberi mutazionali dai dati individuali di tumori.

¹⁷Ramazzotti et al., Learning mutational graphs of individual tumour evolution from single-cell and multi-region sequencing data, BMC Bioinformatics 20, 210 (2019). <https://doi.org/10.1186/s12859-019-2795-4>

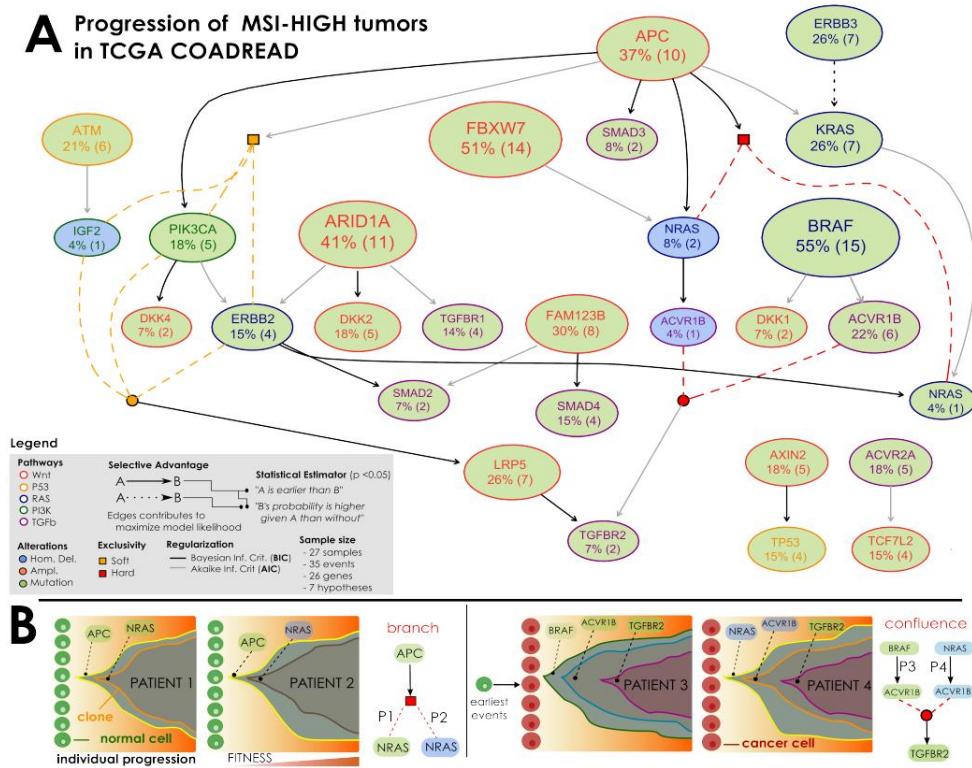


Figura 10.10: Immagine coi risultati del modello di progressione completo dello studio su *MSI-HIGH TCGA COADREAD 2012*. La codifica a colori nella figura A con il DAG, indica i diversi tipi di alterazioni, i percorsi coinvolti e gli schemi disgiuntivi. L'immagine A è ottenuta direttamente dalla pipeline *PiCnIc* mentre l'immagine B presenta grafici ottenuti in seguito manualmente, in quanto non possono essere generati automaticamente.

In questo contesto quindi lo studio dell'evoluzione tumorale/cancerogena diventa lo studio dell'evoluzione di una singola popolazione di cellule all'interno di un tumore. Come già detto più volte il cancro sviluppa il progressivo accumulo di alterazioni genomiche ed epigenetiche, dette *mutazioni driver*, che possono essere modellate tramite modelli di filogenesi o similari, ovvero tramite *alberi* o in rari casi *grafti*. Inoltre nel contesto di questi studi non bisogna dimenticare la già citata **Intra Tumour Heterogeneity (ITH)**.

Nell'ottica di ricostruire quale processo evolutivo sia in corso per il tumore si possono ottenere quattro modelli evolutivi, raffigurabili come in figura 10.11¹⁸, per quattro tipi di evoluzione, ovvero:

1. linear evolution
2. branching evolution
3. neutral evolution
4. punctuated evolution

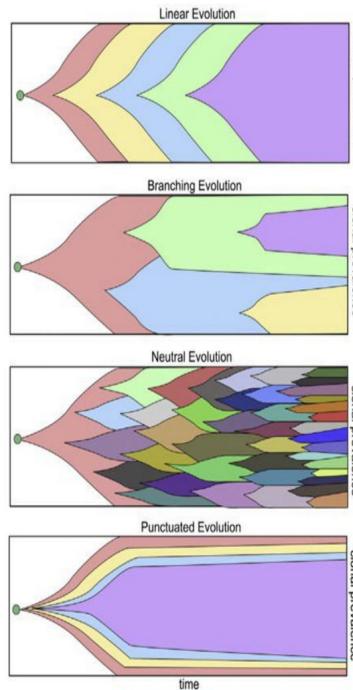


Figura 10.11: Rappresentazioni grafiche manuali dei vari modelli di evoluzione tumorale. Oggigiorno esistono tool per ottenere tali grafiche.

¹⁸Davis, A., Gao, R., Navin, N. (2017) Biochim Biophys Acta 1867(2)

In *BIMIB/DCB* inoltre sono in corso anche studi in merito a come discernere come studiare il tumore principale a fronte di eventuali metastasi.

Si ricorda che dal punto degli *individual data* si hanno due situazioni possibili, descritte da Schwartz e Schäffer¹⁹:

- partire dal **bulk sequencing** (con i vari “side-effects” relativi a questo tipo di sequenziamento, per quanto molto usato in quanto economico) e produrre una albero di filogenesi dei “pezzi” di tumori, distinguendo nei nodi i tumori principali e le metastasi e avendo come radice un tumore progenitore. Con il *bulk sequencing* si trattano dati provenienti da più regioni del tumore, dati che sono a *deep coverage*, studiando principalmente deconvoluzioni dei segnali come i **Variant Allele Frequency (VAF)** (???)
- partire dai dati di **single-cell sequencing (SCS)** (con costi molto elevati), isolando cellule nel tumore principale o nelle metastasi. Si segnala inoltre che tali sequenziamenti non hanno un altissimo *coverage*. In questo caso si ottiene un albero di filogenesi con i vari sotto-cloni di un tumore, avendo come radice una cellula ancestrale sana, ottenendo un modello filogenetico molto più preciso per costruire modelli di progressione statisticamente ben fondati di un singolo tumore. Queste analisi hanno però anche problematiche tecniche dovute al dover isolare le cellule (un isolamento anche solo parziale impedisce il successo dell’perimento, avendo risultati sfalsati). Tali problematiche sono comunque gestite sempre più con facilità. Di recente, anche in *BIMIB/DCB*, si è iniziato ad usare tecniche di ML/DL, come gli *auto-encoder*, per “pulire” i dati prodotti da *SCS* in caso di problematiche di isolamento. Un’altra problematica tecnica è dovuta al fatto che si usa il cosiddetto **whole genome amplification (WGA)**, che potrebbe comportare ulteriori problematiche. Si hanno anche errori legati prettamente ai dati, come:
 - allelic dropouts (*ADOs*)
 - falsi alleli
 - dati mancanti
 - coverage non unifome
 - doublets

¹⁹The evolution of tumour phylogenetics: principles and practice, R. Schwartz and A. A. Schäffer, Nature Review Genetics, 2017

10.4.1 Tipologie di Alberi

Vediamo quindi nel dettaglio le tipologie di output che si producono in fase di studio di dati individuali, nel dettaglio assumendoli di provenienza *SCS*. La maggior parte degli algoritmi sono ormai storici, in quanto nati nella fase di massima crescita dell'algoritmica, intorno agli anni settanta.

In tutti i casi si parte da:

- un insieme di mutazioni, che nelle immagini verranno rappresentate tramite “nuvolette” etichettate con un *id* per ogni mutazione
- un insieme di cellule che “annotano” tali mutazioni, che nelle immagini verranno rappresentati tramiti cerchi colorati

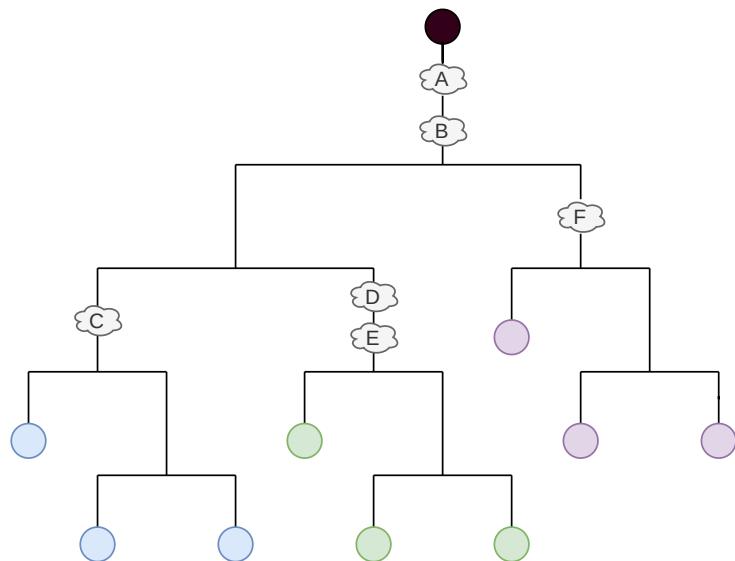
Tutti questi problemi solitamente appartengono comunque alla classe dei problemi *NP-hard*, quindi la loro gestione, dal punto di vista computazionale, non è affatto banale e deve essere sempre tenuta in considerazione durante lo sviluppo di nuove soluzioni.

Standard Phylogenetic Trees

Sono il modello standard e più studiato in letteratura. La struttura è la seguente:

- le foglie rappresentano le cellule
- gli archi sono etichettati dalle mutazioni

Si ottiene quindi un modello del tipo:



Si hanno ormai vari tool standard per il loro studio²⁰.

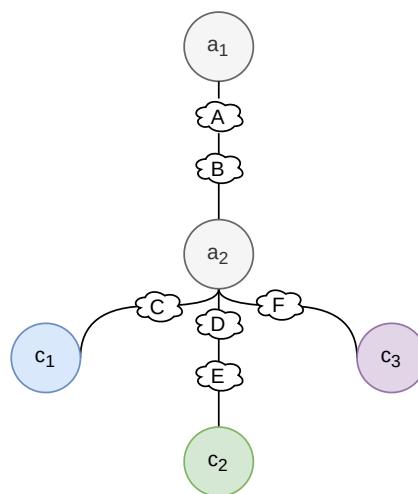
Clonal Lineage Trees

Questo particolare tipo di alberi può essere creato a partire da quelli di filogenesi standard, essendo di fatto una rappresentazione compatta degli stessi dove si hanno:

- i nodi che rappresentano i cloni, quindi sotto-popolazioni tumorali con una certa caratteristica, inferibili anche con tecniche che vanno oltre il mero algoritmo di ricostruzione dell'albero. Si rappresenta quindi la cosiddetta *clonal signature*
- gli archi sono etichettati dalle mutazioni

In questi alberi si ha un'*ordinazione basata sulla prevalenza*.

Si ottiene quindi un modello del tipo:



Si hanno vari tool dedicati a questa soluzione, tra cui:

- *Bitphylogeny*²¹, tra i più usati
- *OncoNem*²²
- *Single Cell Genotyper*²³
- *ddClone*²⁴, tra i tool più sofisticati

²⁰Davis, A., Navin, N. (2016) Genome Biology, 17(1):113

²¹Yuan et al. (2015) Genome biology 16(1), 1

²²Ross & Markowetz (2016) Genome biology 17(1), 1

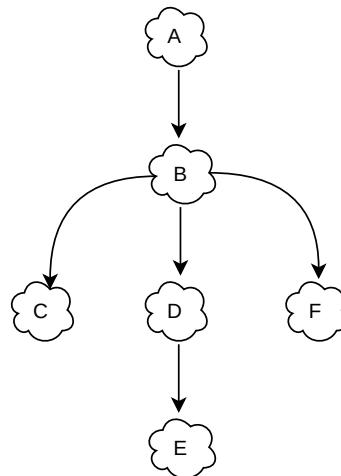
²³Roth et al. (2016) Nat met 13(7), 573-576

²⁴Salehi et al (2017) Genome biology 18:44

Mutational Trees

In questo caso considerano invece solo le mutazioni, in modo analogo a quanto visto con i *DAG* prodotti nel caso di dati *cross-sectional* usando *TRONCO*. Si hanno quindi solo le mutazioni, nei nodi, e si ha un *DAG* che rappresenta l'ordine delle mutazioni (delle varie tipologie, come le *copy-number* etc...) stesso.

Si ottiene quindi un modello del tipo:



Si hanno vari tool dedicati a questa soluzione, tra cui:

- *MUTTREE*²⁵
- *SCITE*²⁶, che è il primo di una serie di tool dedicati
- *SiFit*²⁷, che è un tool davvero molto sofisticato in quanto è il primo che non si basa sulla *Infinite Site Assumption*

Tutti questi tool si basano su determinate assunzioni tra cui quella detta **Infinite Site Assumpio (ISA)**, tranne *SiFit*. Questa assunzione ci impone che:

ogni mutazione si verifica al massimo una volta durante la storia evolutiva di un tumore e non viene mai persa.

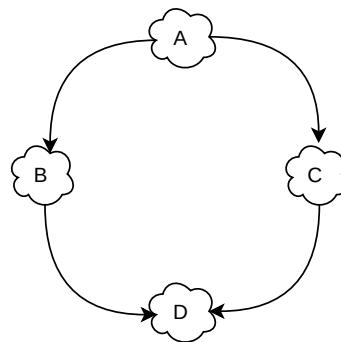
²⁵Kim, & Simon (2014), BMC bioinformatics, 15(1), 27

²⁶Kuipers et al. (2016) Genome biology, 17(1), 86

²⁷Zafar et a. (2017), Genome Biology 18:178

Essendo quindi un'assunzione, che incide soprattutto nel caso di modelli *mutational trees* piuttosto che *clonal lineage trees*, che permette di “alleggerire” la complessità computazionale dei problemi trattati.

Si hanno comunque violazioni possibili alla *ISA*, ad esempio nel caso di **evoluzione convergente**, che comporta quindi un'evoluzione a “diamante” del tipo:



10.4.2 TRaIT

Si introduce quindi, in modo spanno-metrico, il funzionamento del sottomodulo di *TRONCO* chiamato **TRaIT** (*Temporal oRder of Individual Tumors*).

Questo modulo fornisce una stima robusta dell'ordinamento mutazionale nei singoli tumori e supporta sia dati *multi-regionali bulk* che *SCS*, tramite un framework statistico unificato e non si ha nessun modello di rumore specifico per i dati.

Anche in questo caso si ha in input la solita matrice booleana rappresentante i vari tipi di alterazione, tra cui:

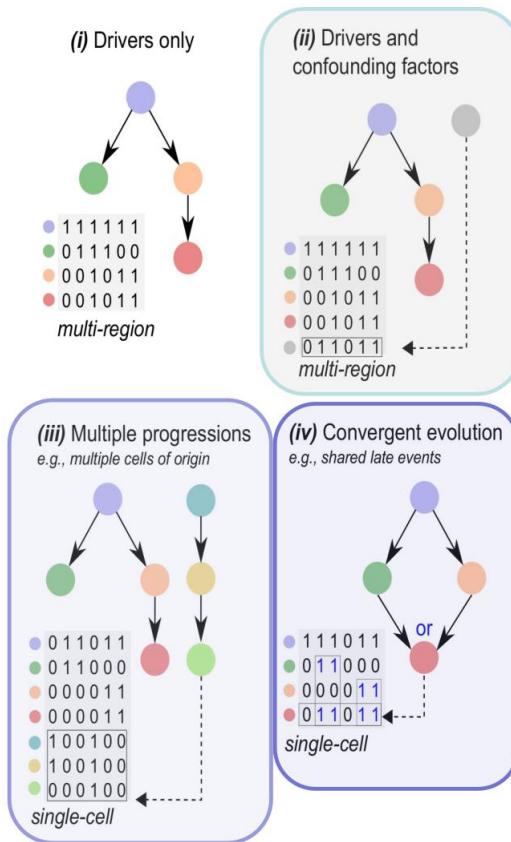
- *SNV*
- *CNA*
- *fusioni*
- ...

e si premura poi di estendere gli alberi mutazionali ai grafi mutazionali , ovvero ai *DAG*, gestendo:

- fattori di confusione
- possibili molteplici traiettorie indipendenti

- violazioni dell'*ISA*, dovute all'evoluzione convergente

Si hanno vari tipi di studi fattibili con *TRaIT*, a seconda della provenienza e dalla tipologia dei dati, riassumibili con questa immagine (anche se si segnala che la forma a “diamante” del quarto modello, a causa dell’uso dei *minimum spanning trees* è solo “ideale”):



Vediamo quindi brevemente i vari step:

1. si parte dalla una matrice booleana che memorizza la presenza di un’alterazione in un campione
2. si procede con una fase di *bootstrap non parametrico*, per ottenere un *bootstrap prima facie* praticamente tramite l’algoritmo *CAPRI* assegnando:
 - ordine temporale
 - associazione statistica

in modo da poter poi usare la teoria di Suppes. Si fa anche una fase di testing delle ipotesi, producendo un grafo diretto dove i nodi/variabili sono le varie alterazioni mentre gli archi sono pesati tramite la *mutua informazione*, eventualmente tramite la *mutua informazione puntuale* e si estraggono i modelli di output con strategie algoritmiche basate su misure teoriche dell'informazione

3. dal grafo ottenuto si rimuovono i cicli
4. a questo punto si può scegliere se mantenere o meno l'orientamento del grafo (e usando la mutua informazione che è simmetrica posso non tenere l'orientamento) e i base a tale scelta si hanno due alternative per il calcolo del *minimum spanning tree (MST)*:
 - il calcolo dell'*MST* a partire da un grafo diretto, aciclico e pesato tramite l'*algoritmo di Edmonds* o l'*algoritmo di Gabow*. Il risultato sarà un *MST diretto*, un *DAG* in pratica
 - il calcolo dell'*MST* a partire da un grafo indiretto tramite l'*algoritmo di Prim* o l'*algoritmo di Chow-Liu*. Il risultato non sarà un *MST diretto* e quindi serve una fase di postprocessing. In questa fase si introducono nuovamente le conoscenze di ordinamento temporale e crescita probabilistica, per la teoria di Suppes, che già si hanno (erano infatti solo state dimenticate nel corso del processo)

Tutti gli algoritmi per il calcolo dell'*MST* sono in tempo *polinomiale* ma i due relativi al calcolo a partire da grafi non orientati sono più efficienti.

Volendo si può anche scegliere di usare l'*algoritmo CAPRESE* durante l'uso di *TRaIT*.

In figura 10.12²⁸ e 10.13²⁹ alcuni grafici di due studi fatti con *TRaIT*, rispettivamente sul *CRC* e sul *cancro ai polmoni*.

²⁸Lu, You-Wang, et al. "Colorectal cancer genetic heterogeneity delineated by multi-region sequencing." PloS one 11.3 (2016): e0152673

²⁹Wang, Yong, et al. "Clonal evolution in breast cancer revealed by single nucleus genome sequencing." Nature 512.7513 (2014): 155

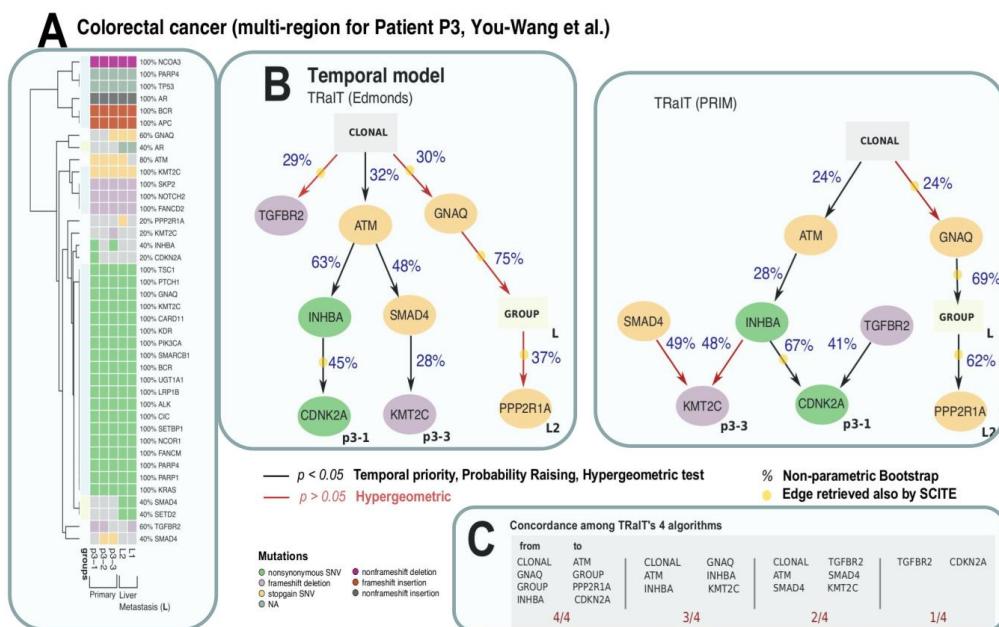


Figura 10.12: Esempio di studio sul *CRC*, partendo da dati *multi-region bulk*, dove si parte dalla solita matrice dove si hanno sulle righe le varie alterazioni e sulle colonne i vari sample del singolo paziente. S’hanno poi due *MST*, prodotti uno con l’*algoritmo di Edmonds* e uno con l’*algoritmo di Prim*. Si ha infine, nella figura C, il risultato vero dello studio, con il confronto dei quattro algoritmi di *TRaIT*.

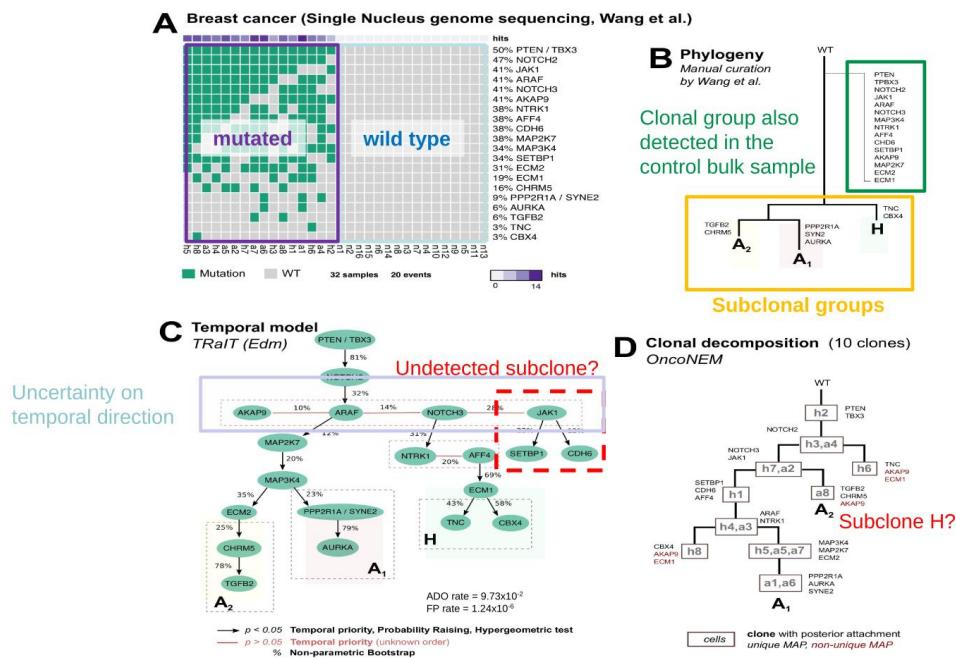


Figura 10.13: Esempio di studio sul *cancro ai polmoni*, uno dei primi fatti con dati *single-cell*. In figura *B* si ha la filogenesi curata manualmente attesa, con 3 gruppi di sotto-cloni. In figura *C* l'uso di *TRaIT* rileva i tre gruppi più uno aggiuntivo mentre in figura *D* nl'uso del tool *OncoNEM*, sviluppato a Cambridge, si dimostra insufficiente non riconoscendo tutti e tre i gruppi.

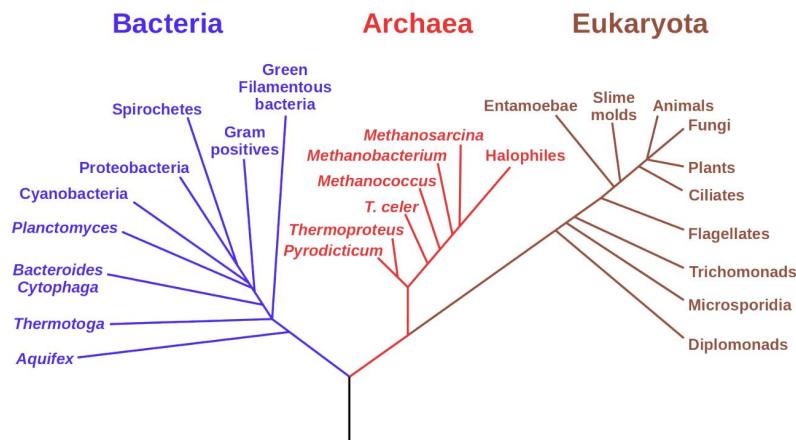


Figura 10.14: Rappresentazione dell'*albero filogenetico della vita* con i tre principali domini.

10.5 Teoria della Filogenesi

Si riflette ora sui metodi per ricostruire modelli di evoluzione cancerogena, discutendo diversi approcci che sono i principi alla base della filogenetica dei tumori. Si analizzeranno poi alcuni metodi alternativi che sono il risultato di, invece, diverse assunzioni. Anche in questo caso si tratterà principalmente quanto scritto nel paper di Schwartz e Schäffer³⁰

Una comoda rappresentazione dell'evoluzione delle specie può essere visualizzata nel cosiddetto **albero filogenetico della vita**, in figura 10.14, dove troviamo rappresentati i tre principali domini riconosciuti:

1. *batteri*
2. *archaea*
3. *eucarioti*

L'albero è stato costruito sulla base dell'*RNA ribosomiale* e si basa, dal punto di vista delle *relazioni*, sulla nozione chiave di **antenato “più vicino”**.

Nel dettaglio si può studiare come, ad esempio, *eucarioti* e *archaea* siano più “vicini” rispetto a quanto sarebbero coi *batteri*. La radice dell'albero ha una sua denominazione, ovvero **LUCA (Last Universal Common Ancestor)**.

³⁰R. Schwartz and A. A. Schäffer, “The evolution of tumour phylogenetics: principles and practice.,” *Nature Reviews Genetics*, vol. 18, no. 4, pp. 213–229, Feb. 2017.

Come ben risaputo è stato Darwin a studiare per primo l'evoluzione delle specie e fu lui a fissare in primis tre aspetti che bisogna considerare negli studi:

1. i tratti morfologici o fenotipici tra gli individui variano
2. la variazione dei tratti conferisce a ciascun individuo un'idoneità complessiva, dato l'ambiente
3. gli individui più idonei/adattati si riproducono più facilmente e possono trasmettere l'insieme dei tratti che li rendono più adattati alla loro prole

Questi aspetti caratterizzano il meccanismo che regola l'*evoluzione Darwiniana*, meccanismo che viene chiamato **selezione naturale**. Tale meccanismo è la motivazione per cui gli alberi filogenetici hanno una forma abbastanza prevedibile (*forma diadica*).

Si noti che per circa 40 anni dopo la morte di Darwin l'ereditarietà dei tratti fu una delle questioni spinose della *teoria della selezione naturale*. Con i lavori di Mendel e altri si è arrivati all'attuale teoria dell'evoluzione, detta **sintesi/teoria moderna dell'evoluzione** (o anche *neodarwinismo*), che ha come menti fondanti quelle di Haldane, Fischer, Wright e altri. Questa teoria concilia di base la genetica di Mendel e la teoria di Darwin (ma anche la forma matematica della genetica delle popolazioni e l'analisi dei dati della paleontologia). In sintesi, il neodarwinismo consiste nel considerare il gene come unità fondamentale dell'eredità e bersaglio del meccanismo evoluzionistico della selezione naturale. La sintesi neodarwiniana unifica diverse branche della biologia che in precedenza avevano pochi punti di contatto, in particolare la genetica, la citologia, la sistematica, la botanica e la paleontologia³¹.

Inoltre Crick e Watson, coi loro studi, trovarono nel *DNA* la conferma fisica del fatto che la *sintesi moderna dell'evoluzione* fosse effettivamente funzionante.

10.5.1 Algoritmi di Filogenesi

Il prossimo passo è capire come ricostruire la filogenesi che spieghi la “ramificazione” delle forme di vita, data la conoscenza sull’evoluzione delle specie. In letteratura possiamo identificare, in modo comunque molto generale, tre categorie di metodi/algoritmi per la ricostruzione di tali alberi:

³¹https://it.wikipedia.org/wiki/Sintesi_moderna_dell%27evoluzione

1. metodi basati sulla distanza
2. metodi di massima parsimonia
3. metodi di massima verosimiglianza e inferenza Bayesiana

Metodi Basati sulla Distanza

Questa prima categoria di metodi su base appunto sulla nozione di **distanza** tra gli elementi che saranno, perlomeno alcuni, i nodi dell'albero risultante. Tale *distanza* può essere caratterizzata in vari modi, ad esempio:

- funzione dei tratti osservati nelle specie
- una distanza dal punto di vista delle sequenze genetiche, come la **distanza di Hamming**

La *distanza*, inoltre, può essere una misura di **similarità** o di **dissimilarità**. Tali algoritmi (ma in realtà anche le altre categorie di algoritmi per la filogenesi) si basano su iterazioni con degli step standard in ogni iterazione. In particolare, per i *metodi basati sulla distanza*, presa in input l'attuale matrice di distanza si fanno le seguenti operazioni:

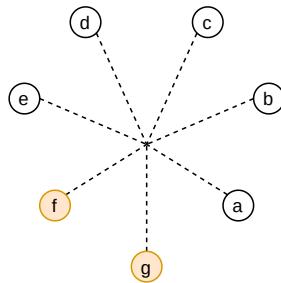
1. si calcola un nuovo punto di diramazione
2. si calcola, o precisamente o tramite una stima, la lunghezza del ramo
3. si aggiorna la matrice

Ogni singolo passaggio ovviamente varia leggermente a seconda dell'algoritmo usato, infatti i metodi basati sulla *distanza* possono a loro volta essere categorizzati in:

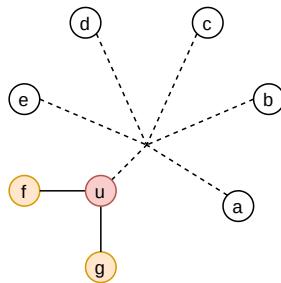
- **metodi pair-group**, basati sulla media aritmetica, pesata o meno. Tali algoritmi, tra cui abbiamo appunto **WPGMA (Weighted Pair Group Method with Arithmetic Mean)** e **UPGMA (Unweighted Pair Group Method with Arithmetic Mean)**, sono sostanzialmente algoritmi di clustering gerarchico
- **metodo neighbor joining**, un diverso metodo di clustering agglomerativo
- **metodi dei minimi quadrati (least-squares)**, come l'**algoritmo Fitch-Margoliash**

Vediamo, a titolo di esempio, uno step di esecuzione dell'*algoritmo neighbor joining*.

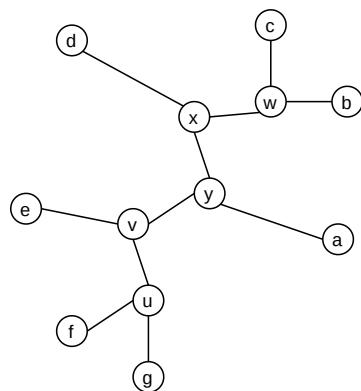
Supponiamo di avere sette eventi, così disposti (gli archi tratteggiati significano che non sono definitivi):



e si supponga di voler unire i nodi f e g , creando il nuovo nodo u (i due rami nuovi, indicati con la linea piena):



I nuovi archi vengono quindi memorizzati nella struttura e si aggiorna la matrice delle distanze, aggiungendo il nodo u , e calcolando le distanze tra esso e i nodi a, b, c, d e e , non considerando f e g . Andando avanti con altri step come quello appena visto si potrebbe, ad esempio, arrivare a qualcosa del tipo:



Metodi di Massima Parsimonia

L'idea dietro di **metodi di massima parsimonia** è quella di costruire l'albero filogenetico con il minimo numero di *step evolutivi*, *step* che si sa a priori non essere equiprobabili, pur raggiungendo un albero finale che sia consistente coi dati in ingresso. Questi metodi quindi procedono associando uno *score* ad ogni *step* e costruendo poi l'albero con minimo peso.

Alcuni di questi metodi, tra cui l'*algoritmo Sankoff-Morel-Cedergren*, usano un mix di procedure che costruiscono un albero di massima parsimonia insieme al **Multiple Sequence Alignment (MSA)** delle sequenze in input, che siano di *DNA*, *RNA* o anche *sequenze proteiche*. L'algoritmo utilizza quindi lo *score* massimo di parsimonia e un termine extra che favorisce le sequenze che presentano un grado di *omologia*. Alla fine la costruzione complessiva dell'albero di massima parsimonia e del *MSA* viene fatto mediante **algoritmi approssimanti di programmazione dinamica** che possono quindi anche non portare alla soluzione ottima ma comunque ad una molto prossima e sicuramente accettabile.

Metodi di Massima Verosimiglianza e Inferenza Bayesiana

Nel contesto degli alberi di filogenesi la **massima verosimiglianza/maximum likelihood** viene usata per inferire uno *score* di probabilità da associare all'albero stesso. L'uso della massima verosimiglianza massimizza la probabilità che gli alberi abbiano un numero ridotto di nodi interni, analogamente a quanto accadrebbe con i metodi di massima parsimonia, ma questi sono metodi più "flessibili".

Questi metodi sono implementati mediamente tramite **algoritmi di programmazione dinamica**, nonostante i problemi trattati siano di classe *NP-hard*.

In alternativa si hanno i metodi basati sull'**inferenza Bayesiana**, che campionano da una certa distribuzione di alberi filogenetici consistenti coi dati. Il punto critico è appunto la scelta di questa distribuzione è uno dei primi tool atti a risolvere questa problematica fu *BitPhylogeny*, che fu uno dei primi a fornire una buona *prior* allo studio dell'evoluzione tumorale.

Dal punto di vista implementativo i metodi Bayesiani sono solitamente implementati partendo da uno schema **Markov Chain Monte Carlo (MCMC)** dove, trattandosi essenzialmente di un metodo di ottimizzazione a scelta casuale, la scelta dell'elemento successivo, ovvero del prossimo albero, da prendere in considerazione diventa un altro *grado di libertà*. Anche la scelta di questo elemento successivo, nel contesto dell'evoluzione tumorale, viene facilitata dall'uso di software come il già citato *BitPhylogeny* o come *SCITE*.

Il problema principale con i metodi Bayesiani in piena regola è che l'uso di *MCMC* può essere molto costoso dal punto di vista computazionale. Inoltre, la convergenza ad un campione buono da parte di questi metodi dipende dalle scelte di distribuzione a priori e dalle mosse fatte durante la procedura *MCMC*. Infine, la procedura di selezione del modello richiede spesso un corretto modello di errore o rumore dell'input, aumentando ulteriormente i costi di calcolo.

10.5.2 Software per Filogenesi

Si propone quindi una lista di software usati per lo studio di alberi filogenetici:

- **BitPhylogeny**, un tool basato su metodi Bayesiani, con *MCMC*, per dati *bulk* regionali. In output si ha un *clonal lineage tree*
- **SCITE**, un tool basato su metodi Bayesiani, con *MCMC*, per dati *single-cell*. In output si ha un *standard phylogenetic tree*
- **TRONCO**, un tool basato sulla causalità probabilistica, come indicato nella teoria di Suppes. In output si ha un *mutational tree* e un *DAG*
- **CALDER**, un nuovo tool per l'analisi longitudinali di dati *bulk* con *Variant Allele Frequency (VAF) deconvolution*
- **LACE**, un nuovo tool per l'analisi longitudinali di dati *single-cell*

10.6 Problemi Aperti

Bisogna ora trattare delle varie problematiche tutt'ora aperte nell'ambito della ricostruzione dell'evoluzione tumorale:

1. una problematica riguarda la struttura stessa dei dati in input. È ancora molto difficile infatti integrare lo sorgenti di dati in merito, ad esempio, *SVN*, *CNA*, misure epigenetiche etc..., considerando anche che tali dati sono ormai molto ricchi di informazioni
2. una problematica riguarda l'assenza di un modello preciso vero e proprio per l'evoluzione tumorale. Questo problema si ripercuote soprattutto dei modelli Bayesiani basati su *MCMC*.
Ci sono alcuni fenomeni biologici recentemente scoperti, ad esempio i *Cromotripsi* (un fenomeno genetico, causato da catastrofe cromosomica, nel quale si vengono a creare in un colpo solo

da poche unità a varie centinaia di mutazioni; questo fenomeno è responsabile del rapido sviluppo di varie forme di cancro), che dovrebbero essere presi in considerazione per fornire una corretta funzione di *score* per la maggior parte degli algoritmi di ricostruzione degli alberi filogenetici

3. una problematica riguarda la possibile discrepanza tra gli algoritmi di costruzione della filogenesi, sia per *standard phylogenetic tree* che per *clonal lineage tree*, e i possibili modelli di evoluzione del cancro.

TRONCO cerca di garantire che i *DAG* delle precedenze mutazionali possano essere ricostruiti, ma comunque questo aspetto ha ancora margine di miglioramento

4. una problematica riguarda l'interazione tra la costruzione degli studi per i quali vengono ricostruiti i modelli di progressione in quanto molte volte, l'applicazione di uno o più metodi di ricostruzione della progressione dipende molto dall'effettivo disegno sperimentale.

È necessario molto lavoro per applicare sistematicamente una serie di strumenti a una serie di tumori al fine di trovare punti in comune tra le progressioni risultanti e ottenere informazioni sul miglioramento degli strumenti e degli algoritmi stessi

10.7 Analisi Longitudinale

Quanto trattato in seguito è ritrovabile nei seguenti articoli:

- D. Ramazzotti et al., “Longitudinal cancer evolution from single cells.”, biorXiv, <https://doi.org/10.1101/2020.01.14.906453>, March 2020
- D. Ramazzotti et al., “VERSO: a comprehensive framework for the inference of robust phylogenies and the quantification of intra-host genomic diversity of viral samples”, Patterns, 2021, <https://doi.org/10.1016/j.patter.2021.100212>
- M. A. Myers, G. Satas, B. J. Raphael, “CALDER: Inferring Phylogenetic Trees from Longitudinal Tumor Samples”, Cell Systems 8, 514–522, June 26, 2019 (che però studia *dati bulk*)

Si introduce quindi l'**analisi longitudinale dell'evoluzione del cancro**, anche in questo caso introducendo un tool sviluppato all'interno di *BIMIB/DCB*, vedendo anche alcuni dei nuovi algoritmi/tecnologie a tema, studiando la combinazione della **teoria della ricostruzione della filogenesi** e dei **dati single-cell time-course**, parlando quindi di **longitudinal dataset**. Fino ad ora si sono studiati dati provenienti da momenti temporali ovviamente diversi ma di cui non si conosceva l'ordine e quindi non si poteva sfruttare in alcun modo la conoscenza temporale, avendo, ad esempio, pazienti in *stage* diversi. Non si aveva nei dati un *tag temporale*. Con i *dati longitudinali* si ha invece l'annotazione del tempo, avendo circa quattro/cinque *timestamp*, essendo comunque tali dati costosi da produrre.

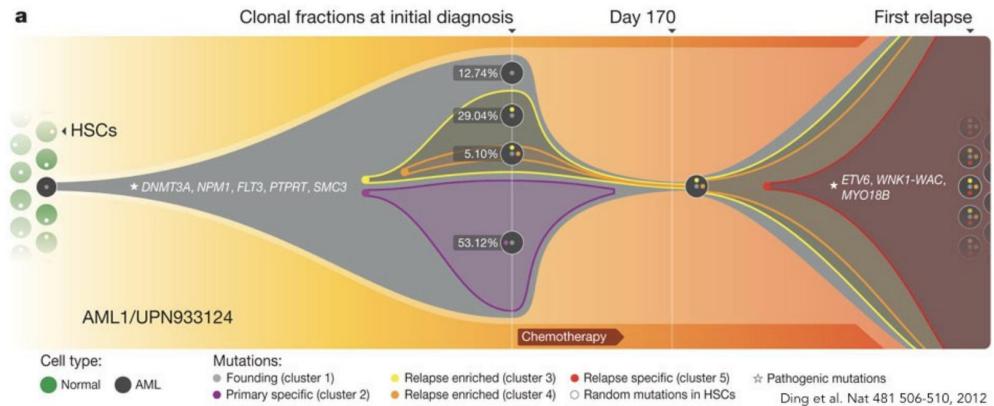
Quando si è parlato di *dati bulk* si studiavano in primis le *Variant Allele Frequencies*, studiando vari sample. Con *dati Single-cell* si studiano in primis le *RNAseq*, più economiche di fare un sequenziamento completo (costo che è il problema principale delle analisi single-cell).

L'obiettivo diventa quindi la ricostruzione di filogenesi per lo studio dell'evoluzione tumorale sfruttando le informazioni temporali annotate nei dati. Tali annotazioni derivano però da esperimenti difficili da fare in primis per un discorso economico (discorso magari destinato a migliorare nel tempo). A causa di questo i tool sviluppati vengono solitamente validati tramite simulazioni estese, al fine di supportare con dei risultati i vari algoritmi, che sono normalmente **algoritmi stocastici**. A tal proposito il gruppo di *BIMIB/DCB* studia gli *organoids*.

Lo scopo dell'*analisi longitudinale*, in termini più pratici, è quindi quello di inferire un modello generativo che rappresenti un fenomeno evolutivo (che può essere sia lo studio di un singolo tumore ma anche della diffusione di un virus) che sia il più consistente possibile coi dati. L'inferenza avviene quindi a partire da:

- *dati single-cell longitudinali*, con multipli sample a diversi time-point (ad esempio prima e dopo una certa terapia)
- *impostazioni sperimentali eterogenee (tendenzialmente un problema)*, avendo sample di diversa grandezza con un diverso numero di singole cellule ai vari timepoint, avendo diversi livelli di rumore ad ogni esperimento effettuato (si usano quindi metodi robusti per questo discorso come *SmartSeq* o *10x*) etc...
- *tipi di dato diversi* (ottenuti tramite sequenziamento *NGS*), avendo magari *RNAseq* (più economiche), tramite *varianti calling*, *DNAseq* (più costose) etc...

Si ottiene quindi qualcosa del tipo:



Avendo che l'analisi permette quindi di:

- analizzare la variazione della composizione clonale nel tempo
- aumentare il “potere statistico” dell’inferenza, avendo più data point ordinati temporalmente
- valutare l’impatto della terapia e la resistenza ai farmaci emergente, avendo che alcuni cloni possono comparire, altri espandersi e altri sparire

Gli esperimenti producono dati anche prima delle chemioterapie, permettendo uno studio più approfondito.

10.7.1 LACE

In *BIMIB/DCB* si è quindi sviluppato un framework algoritmico per l’inferenza di modelli longitudinali dell’evoluzione del cancro. Tale framework è chiamato **LACE** (*Longitudinal Analysis of Cancer Evolution*) e si basa su:

- **Boolean Matrix Factorization (BMF)**, che è una forma particolare del **Non Negative Matrix Factorization**. Tali algoritmi sono abbastanza vecchi ma ancora popolari (e sono usati per vincere la cosiddetta *Netflix Preferences Competition*, per consigliare i prossimi contenuti video). Si ha comunque l’assunzione di *filogenesi perfetta*
- una **funzione obiettivo di massima verosimiglianza pesata** innovativa

- le **Monte Carlo Markov Chain (MCMC)** per esplorare le soluzioni possibili (come fanno la maggior parte dei tool allo stato dell'arte)

Lo stato dell'arte a questo punto prevede:

- per *dati single-cell a singolo timepoint* i seguenti tool:
 - **SCITE**³²
 - **SiCloneFit**³³
 - **TRaIT**³⁴

Dove si inferiscono alberi mutazionali.

Purtroppo, in questi casi:

- non si hanno informazioni temporali
- si ha una bassa confidenza statistica, soprattutto se in presenza di forte rumore
- si hanno eventualmente implementazioni *ad hoc* per gestire multipli timepoint
- si ottengono modelli meno espressivi, senza variazioni sulla prevalenza dei cloni, sulla sparizione degli stessi etc..., avendo che il dataset è solo una sorta di *snapshot*

Si assume quindi di non avere dati longitudinali

- per *dati bulk a multipli timepoint* si ha **CALDER**³⁵, costruendo *clonal tree* partendo da misure di *Variant Allele Frequencies*. Con questa soluzione si ha la perdita di “alta risoluzione”, avendo:
 - dati a “bassa risoluzione” da segnali misti di sotto popolazioni tumorali
 - limitazione delle fasi di deconvoluzione per inferire architetture clonali
 - impossibilità a collegare il genotipo single-cell al fenotipo

³²Tree inference for single-cell data, Jahn, K. et. al – Genome Biology (2016)

³³Zafar, H. et. al – Genome research (2019)

³⁴Ramazzotti, D. et. al – BMC Bioinformatics (2019)

³⁵CALDER: Inferring phylogenetic trees from longitudinal tumor samples. Myers, M. A. et. Al - Cell systems (2019).

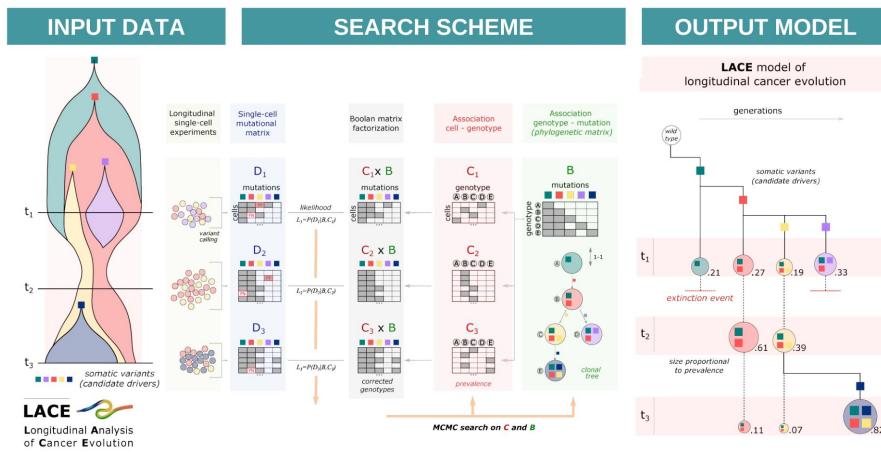


Figura 10.15: Schema generale di *LACE*. Si noti che nella fase di *input data*, il *fishplot* con le varianti somatiche è un disegno fatto a mano e non ottenuto dal codice.

Si sta qui lavorando comunque su dati longitudinali, ottenuti anche in modo economico tramite il *bulk sequencing*

LACE, di cui possiamo vedere uno schema generale in figura 10.15, prende in input *dati somatici longitudinali* da vari esperimenti single-cell. Si producono quindi delle matrici con le mutazioni, che vengono fattorizzate, ottenendo quindi un'associazione tra cellule e genotipi (che sono anch'essi matrici), ottenendo una matrice che codifica di per sé la filogenesi per un certo timepoint. Alla fine si ottiene il modello finale cercando di verificare la coerenza/consistenza tra i vari modelli ai vari timepoint, ottenendo il modello finale. Nel modello finale ci si concentra su alcuni aspetti, ad esempio sono visibili gli eventi di estinzione di un clone e si ha che in ogni nodo si ha conformazioni sul clone prevalente.

Approfondendo il discorso relativo ai dati in input a *LACE* si ha che si tratta di read *RNAseq* che vengono allineate. Nel dettaglio, provenendo le read da studi *single-cell*, si parla più propriamente di **scRNAseq**. A questo punto le **SNV (single nucleotide variation)** calcolate tramite **Genome Analysis Toolkit (GATK)**, costruendo una matrice booleana $n \times m$, dove n è il numero di singole cellule studiate mentre m è il numero di posizioni del genoma con varianti, dove si ha:

- 1 se la mutazione è presente con un alto livello di confidenza in base al *depth coverage* del sequenziamento
- 0 se la mutazione è assente

- NA (visto che l'implementazione è in *R*) se non si hanno abbastanza informazioni a causa di un *basso coverage*

Dopo aver ottenuto l'input e averlo manipolato per ottenere la matrice booleana si passa allo schema di ricerca, tramite *MCMC*, per la fattorizzazione della matrice. Dopo ogni step di *MCMC* si ha, dal punto di vista puramente astratto, un nuovo albero di filogenesi, infatti la fattorizzazione corrisponde alla ricerca dell'albero finale nello spazio degli alberi di filogenesi possibili. Si ha quindi un ciclo:

1. dal dataset D in input si propone una matrice che viene studiata come se fosse un *clonal tree* che soddisfa l'ipotesi di filogenesi perfetta. Si associa ad ogni cellula di D , chiamando ogni cellula D_i , ad un clone nel *clonal tree*. A partire da questo albero si ottiene una matrice B , dove le colonne sono le mutazioni e le righe i genotipi. In pratica si sta ricreando un nuovo dataset
2. si associa ogni di D ad un clone in B , ottenendo le *cell attachment matrixes* C_i , che insieme formano il dataset C
3. si esegue il prodotto matriciale ($D_1 = C_1 \times B$, $D_2 = C_2 \times B$ etc...) per riottenere un nuovo D
4. si calcola la verosimiglianza di osservare D dato B e C
5. eventualmente si ripete da capo usando come input D

Si ripete il ciclo per massimizzare la **log-verosimiglianza pesata (weighted log-likelihood)** (le varie L_i), includendo tutti i vari timepoint:

$$w_1 \ln(L_1) + w_2 \ln(L_2) + \dots$$

avendo che di default i pesi w_i sono calcolati per essere inversamente proporzionali alla dimensione del campione al fine di confrontare i valori di verosimiglianza attraverso i punti temporali. In pratica dopo ogni iterazione associo un punteggio all'albero ottenuto. Oltre alla *filogenesi perfetta* posso avere altri vincoli imponibili.

La fattorizzazione è computazionalmente pesante ma non così tanto come l'intera esplorazione dello spazio degli alberi di filogenesi.

Bisogna quindi calcolare la verosimiglianza ad ogni timepoint t . Ad esempio, per $t = 1$, avrei, avendo $G_{i,j}$ elemento di $G_1 = C_1 \times B$ e $d_{i,j}$ un elemento di D_1 :

$$L_1 = P(D_1|B, C_1) = \prod_{i=1}^n \prod_{j=1}^m P(d_{i,j}|G_{i,j})$$

avendo quindi:

$$P(d_{i,j}|G_{i,j}) = \begin{cases} \alpha & \text{se } d_{i,j} = 1 \wedge G_{i,j} = 0 \\ 1 - \alpha & \text{se } d_{i,j} = 1 \wedge G_{i,j} = 1 \\ \beta & \text{se } d_{i,j} = 0 \wedge G_{i,j} = 1 \\ 1 - \beta & \text{se } d_{i,j} = 0 \wedge G_{i,j} = 0 \\ 1 & \text{se } d_{i,j} = NA \end{cases}$$

I vari α , β , $1 - \alpha$, $1 - \beta$ e 1 sono i genotipi, avendo che *alpha* e β sono specifici per ogni timepoint. I valori $d_{i,j}$ sono quelli osservati mentre i valori $G_{i,j}$ sono quelli attesi, avendo quindi come *matrice di confusione*:

	Si	no
si	$1 - \alpha$	β
no	α	$1 - \beta$

Bisogna quindi verificare la correttezza di quanto calcolato ad ogni step tramite la verosimiglianza.

Lo schema di ricerca quindi cerca di massimizzare la *log-verosimiglianza pesata*, al fine di poter considerare multipli timepoint t . In questo contesto ogni peso w_i , con i che indica i timepoint, può essere usato per gestire dei dataset con un numero diverso di cellule e di livelli di rumore. Si ha infatti, con $ncell_i$ numero di cellule al tempo i :

$$w_i = 1 - \frac{ncell_i}{\sum_{i=1}^t ncell_i}$$

Avendo infine il “punteggio” complessivo per una certa iterazione di *MCMC* è la *log-verosimiglianza pesata*:

$$W = \sum_{i=1}^t w_i \ln(L_i)$$

Bisogna ora capire come si esplora lo spazio delle filogenesi possibili.

Ad ogni step si eseguono una serie di operazioni standard al fine di modificare a dovere l’albero filogenetico attualmente in analisi. La procedura *MCMC* in uso include due mosse ergodiche che, partendo dall’albero corrente, garantiscono l’ipotesi di *filogenesi perfetta* nella modifica dello stesso per la costruzione del nuovo albero:

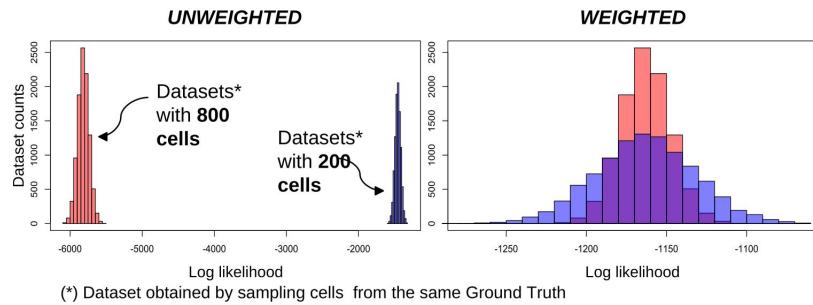
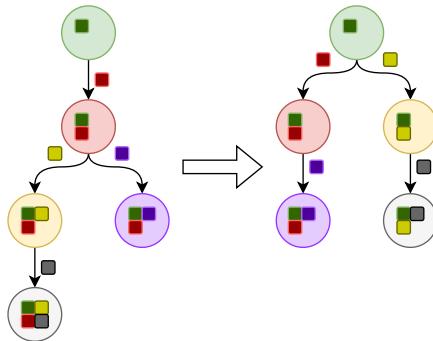
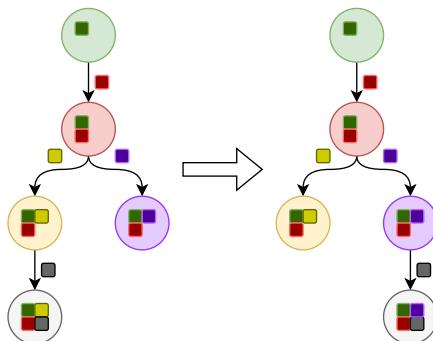


Figura 10.16: Grafici di confronto tra lo studio della *log-verosimiglianza pesata* nel caso non pesato e nel caso pesato. Si nota che in pratica con il caso pesato si ottiene una sorta di normalizzazione.

1. **Prune and Reattach**, letteralmente staccando e riattaccando i nodi in modo diverso, ad esempio:



2. **Node relabeling**, ovvero etichettando i nodi in modo diverso, ad esempio:

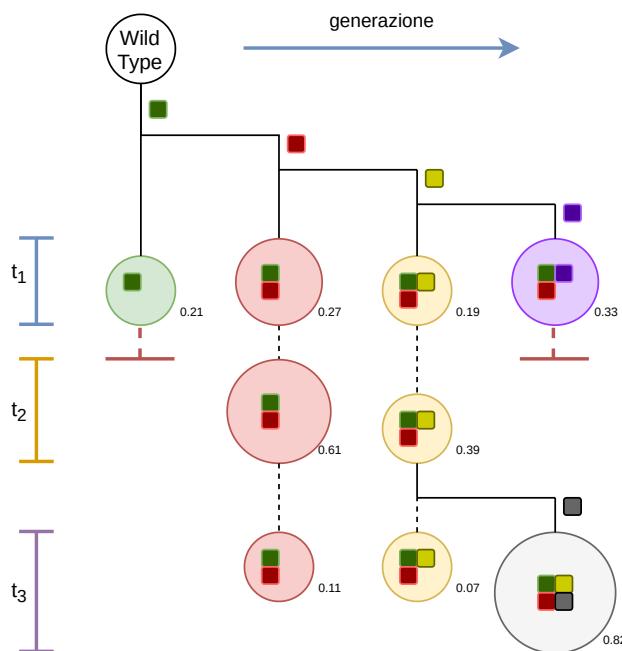


Quindi per ogni configurazione proposta B si cerca la massima verosimiglianza C tramite una ricerca esaustiva, anche se questo risulta essere dispendioso.

so dal punto di vista computazionale. Per fare questo si usa l'**algoritmo Metropolis–Hastings**, che è un metodo *MCMC* per ottenere una sequenza di campioni casuali da una distribuzione di probabilità per la quale è difficile il campionamento diretto. Tale algoritmo è usato per campionare a partire da distribuzioni multidimensionali, specialmente quando il numero di dimensioni è alto. La mossa “candidata” viene accettata se:

- la verosimiglianza pesata migliora
- la verosimiglianza pesata è peggiore con probabilità ρ , e tale ρ è accettabile secondo un certo vincolo scelto a priori

Questa strategia garantisce la convergenza ad un numero infinito di iterazioni. Come output si ha quindi un **albero longitudinale con nodi pesati**, come ad esempio, indicato coi quadratini le varianti somatiche:



Nell'esempio notiamo come in rosso siano indicati gli eventi di estinzione, con le linee piene le *relazioni di parentela*, che sono interne ad un singolo timepoint, mentre con le linee tratteggiate le *relazioni di persistenza*, che sono tra timepoint diversi. I nodi sono (normalmente meglio che nella figura) di grandezza proporzionale alla loro prevalenza, indicata comunque in basso a destra di ogni nodo. Vengono indicati anche i vari timepoint mentre la generazione si legge da sinistra a destra.

A parire dall'albero poi si associa, tramite i vari timepoint, il rispettivo *fishplot* nonché vari studi sullo stato cellulare, sulla diffusione, sulle fasi del ciclo cellulare etc... (su slide 13, pagina 21, immagine di uno studio reale, su *BRAF mutated-melanoma*, con 4 timepoint e multipli esperimenti per ogni timepoint, e il dataset ottenuto tramite *Patient Derived Xenograph (PDX)*).

Tramite *LACE* quindi si possono identificare cloni e subcloni con differenti sensibilità alle terapie. Tra le cose particolari, scoperte sempre nel caso del *BRAF mutated-melanoma*, si è notato un comportamento inaspettato del meccanismo del ciclo cellulare in diversi (sub)cloni dopo il trattamento. Solo (sub)clone del gene *PRAME* manteneva un certo numero di cellule nella cosiddetta *fase S*, implicando che queste cellule potevano essere bersaglio di un trattamento molto specifico.

Quindi si ha che *LACE* consente una mappatura tra l'evoluzione clonale e le proprietà fenotipiche delle singole cellule e questi risultati suggeriscono che le analisi longitudinali di singole cellule sono efficaci nel sezionare i meccanismi con cui le cellule tumorali reagiscono al trattamento.

10.7.2 VERSO

Vediamo quindi un'estensione del framework *LACE* ad un contesto diverso, quello delle diffusioni virali, tramite **VERSO** (*Viral Evolution Reconstruction*)³⁶.

LACE può essere usato, in generale, per ricostruire *processi di accumulazione* e quindi, con qualche modifica, si è ottenuto appunto *VERSO*, il cui schema è visualizzabile in figura 10.17, per lo studio dell'evoluzione del virus *SARS-CoV-2*, ricostruendone la filogenesi.

Nel dettaglio si cercava la filogenesi delle cosiddette *minor mutations*, studiando le sotto-varianti e le sotto-discendenze delle varie mutazioni. Questo è possibile in quanto il genoma dei virus è molto piccolo e quindi può essere studiato nella sua interezza, inoltre si vuole cercare di inferire dal modello e dai dati cosa può accadere nel futuro del virus stesso, come muterà etc...

Uno dei problemi è che i dati pubblicati erano dei *consensus data*, con l'intero set di read usato per ottenere il consenso. Tale set di read può essere usato anche per studiare le piccole varianti, anche se non incluse nel consenso, e tali varianti possono essere il segno di una nuova mutazione del virus stesso che emergerà nel futuro (nel dettaglio alcune parti della *variante omicron* erano state predette da *VERSO*).

³⁶VERSO: a comprehensive framework for the inference of robust phylogenies and the quantification of intra-host genomic diversity of viral samples, Ramazzotti, D. et al., bioRxiv October 2020; <https://doi.org/10.1101/2020.04.22.044404>.

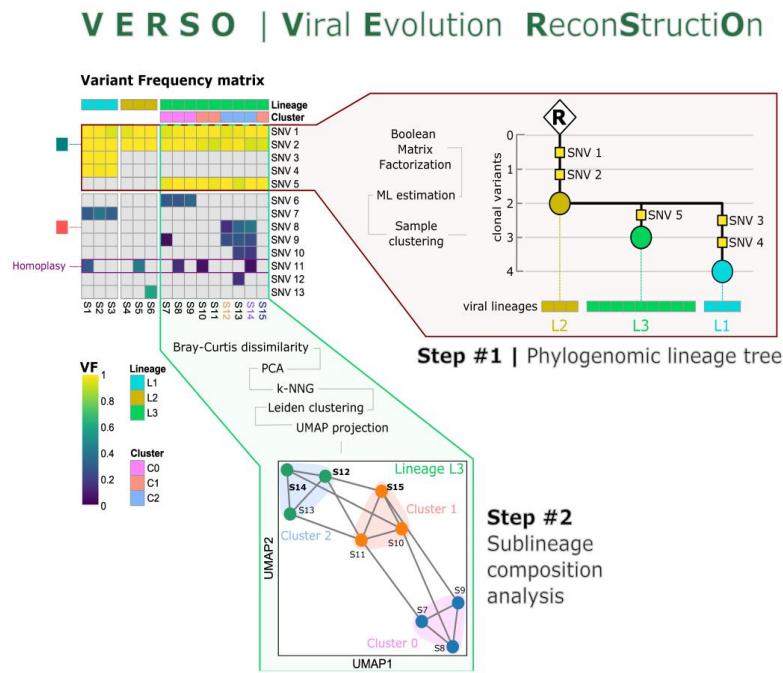


Figura 10.17: Schema generale di VERSO

Si parte dalla *variant frequency matrix* per ottenere un *phylogeomic lineage tree*, dove si hanno discendenza/lineage che possono contenere sublineage con sotto-varianti. A questo punto si fa una *sublineage composition analysis*, tramite clustering, facendo anche un mapping delle sublineage con i dati geografici, ad un tempo stabilito, per rafforzare i risultati dello studio.

Con questo metodo si ha quindi una migliore inferenza filogenetica in presenza di informazioni rumorose e limitazioni di campionamento. L'applicazione a 3690 campioni virali (via sequenziamento dell'amplicone, un frammento di DNA o RNA che è la fonte e/o il prodotto di reazioni di amplificazione o di replicazione) e a 2766 campioni RNA-seq ha migliorato la stima dell'evoluzione e della diffusione del virus.

L'uso, inoltre, di modelli di co-occorrenza di varianti minori ha prodotto percorsi di infezione potenzialmente non rilevati, convalidati da dati di tracciamento dei contatti (*contact-tracing*) per quanto riguarda i dati geografici. L'analisi approfondita del panorama mutazionale del virus conferma un aumento statisticamente significativo della diversità genomica nel tempo con un numero di varianti che possono transitare da uno stato minore a uno stato clonale nella popolazione, nonché diverse *omoplasie*, ovvero la condivisione da parte di due organismi di un carattere comune non ereditato da un ante-

nato comune alle due specie.

Si hanno quindi risultati che suggeriscono che le analisi longitudinali single-cell sono efficaci nel sezionare i meccanismi con cui le cellule tumorali reagiscono al trattamento, infine infatti, vengono derivati modelli di Farmacodinamica e Farmacocinetica tenendo conto dei modelli di progressione e dei modelli longitudinali, al fine di valutare meglio gli effetti di un farmaco (e servono dati ben precisi per non avere modelli troppo generici).

Il grande problema resta però, soprattutto nel caso di studi virali, quello di predire comportamenti futuri partendo dagli snapshot attuali.

10.7.3 Problemi Aperti

Ci sono comunque anche dei *problemī aperti*, che ovviamente riguardano sia *LACE* che *VERSO*:

- si ha il vincolo dell'*assunzione di filogenesi perfetta*, con il conseguente “rilassamento” della **infinite Site Assumption (ISA)**, a causa, ad esempio, di *back mutations* ed *evoluzioni convergenti*
- ci sono spazi di miglioramento per quanto riguarda l’inferenza Bayesiana:
 - le conoscenze a priori potrebbero includere ulteriori conoscenze biologiche, ad esempio in merito alla conformazione delle proteine atte al *binding* etc...
 - i vincoli temporali esplicativi includono la funzione di verosimiglianza. Si vorrebbero quindi usare meno vincoli temporali
 - si ha una fase di clustering a posteriori di eventi mutazionali per eseguire la selezione delle caratteristiche. Si ha infatti che:

$$P(B, C|D) \propto P(D|C, B)P(B, C)$$

Capitolo 11

Single-Cell Data Preprocessing

Si fa ora un'introduzione alla produzione e alla gestione dei dati *single-cell*, parlando di *sequenziamento*, *CNA* e di *pipeline* per la loro analisi.

11.1 Single-Cell Sequencing

Una delle prime cose interessanti riguarda la produzione di *scRNAseq* per la quantificazione dei livelli di espressione genica in ogni cellula. L'output del sequenziamento però è molto rumoroso e “sparso” a causa di problematiche tecniche nel sequenziamento stesso. Il primo problema è quindi quello di identificare i trascritti di *mRNA* ma solo una piccola frazione di tutte le trascrizioni in una cellula viene effettivamente identificata e questo si traduce in dati mancanti che bisogna considerare. Questo problema di dati mancanti è detto *dropout*. Inoltre, durante la **PCR** (*Polymerase chain reaction*), si procede con l'amplificazione del materiale genico e questo comporta molte copie dello stesso trascritto, portando ad un *bias di amplificazione*. Questo problema si corregge tramite **UMI** (*Unique Molecular Identifiers*), una tecnica biotecnologica, in cui si attaccano sequenze univoche a ciascun trascritto prima dell'amplificazione (in pratica si ha un *tag* per ogni trascritto prima dell'amplificazione).

Si ottiene quindi, dopo il sequenziamento, una matrice, dove le colonne sono le cellule e le righe i geni, che contiene in ogni posizione un conteggio di un certo gene in una certa cellula. Ovviamente alcuni di questi valori sono conteggi veri, altri no a causa del *rumore*. Potrei avere anche conteggi nulli a causa del *dropout*.

Parlando delle tecniche di sequenziamento per *scRNAseq* si hanno principalmente due soluzioni:

1. **full-length sequencing**, dove si sequenzia il trascritto intero. Si ha una maggiore efficienza di acquisizione e profondità di sequenziamento, avendo circa il 70% di conteggi nulli. Solitamente un esperimento di questo tipo coinvolge circa cento cellule
2. **3' sequencing**, dove si sequenzia la fine del trascritto. Si ha una minore efficienza di acquisizione e profondità di sequenziamento, avendo circa il 90% di conteggi nulli. Solitamente un esperimento di questo tipo coinvolge circa un milione cellule (e questo è un vantaggio) e per portarlo a termine si fa uso della *UMI* per correggere il *bias di amplificazione*

Bisogna capire come gestire il *rumore*, facendo **denoising** dei dati prodotti con *scRNAseq*. Negli ultimi due anni sono stati proposti molti metodi computazionali con l'obiettivo di recuperare le informazioni perse a causa del rumore e queste si possono catalogare in quattro gruppi, in base alle loro ipotesi e all'approccio impiegato per eseguire il *denoising*:

1. **data-smoothing**
2. **neural networks**
3. **matrix-theory**
4. **model-based**

11.1.1 KNN-Smoothing

Uno dei principali metodi per il *denoising*, appartenente alla categoria *data-smoothing*, è quello detto **KNN-smoothing**. Per questo metodo si hanno due assunzioni:

- cellule simili mostrano profili di espressione simili
- i conteggi osservati per ogni gene i e per ogni cellula j seguono una *distribuzione di Poisson*, ovvero, chiamando U_{ij} il conteggio osservato, si ha per calcolare l'efficienza di acquisizione:

$$U_{ij} \sim \text{Poisson}(\lambda_{ij}e_j)$$

dove $\lambda_{ij}e_j$ è l'espressione reale

L'idea chiave è quindi quella di aggregare k variabili indipendenti di Poisson, che risultano in variabili secondo la *distribuzione di Poisson*, con il **Signal-to-Noise-Ratio (SNR)**, una metrica per descrivere la qualità della misurazione in presenza di rumore, aumentato di un fattore \sqrt{k} . Si ha quindi, con μ valore atteso e σ rumore dovuto alle tecniche usate in fase di sequenziamento:

$$SNR = \frac{\mu}{\sigma}$$

11.1.2 Deep Count Autoencoder

Un'altro metodo molto usato, appartenente alla categoria delle *neural networks*, è quello detto **Deep Count Autoencoder (DCA)**.

Questo metodo usa appunto un *autoencoder*, un tipo di rete neurale per l'*apprendimento non supervisionato*, dove la fase di *encoding* è validata e ridefinita tentando di riottenere l'input a partire dall'*encoding* stesso. Gli *autoencoder* apprendono quindi una rappresentazione di un insieme dati, l'*encoding* appunto, tipicamente per la *riduzione di dimensionalità*, trainando la rete per “ignorare” dati poco significativi, ovvero il *rumore*. Partendo da un input x si effettua l'*encoding*, sfottendo h . A partire da h si effettua la fase di *deencoding*, ottenendo o (che si vuole il più simile possibile all'input x). Si hanno ovviamente molte varianti degli *autoencoder*.

Si assume in questo caso che ogni misurazione dipende da una componente di *dropout* e da una *distribuzione binomiale negativa*, ovvero una distribuzione di probabilità discreta che modella il numero di successi in una sequenza di prove Bernoulliane indipendenti e identicamente distribuite prima che si verifichi un numero specificato (non casuale) di fallimenti (indicato normalmente con r). Si ha quindi:

$$x_{ij} \sim ZINB(\mu_{ij}, \vartheta_{ij}, \pi_{ij})$$

dove *ZINB* sta per **zero-inflated negative binomial**:

$$ZINB(\mu, \vartheta, \pi) = \text{dropout component} + \text{negative binomial component}$$

Inoltre si usano *autoencoder* con una particolare *loss function*, ovvero la *log-verosimiglianza negativa di ZINB*, avendo così la rimozione del *rumore*. L'output finale del modello, il cui schema è visualizzabile in figura 11.1¹ è quindi una versione senza rumore dell'input.

¹Eraslan, Gökcen, et al. "Single-cell RNA-seq denoising using a deep count autoencoder." *Nature communications* 10.1 (2019): 390.

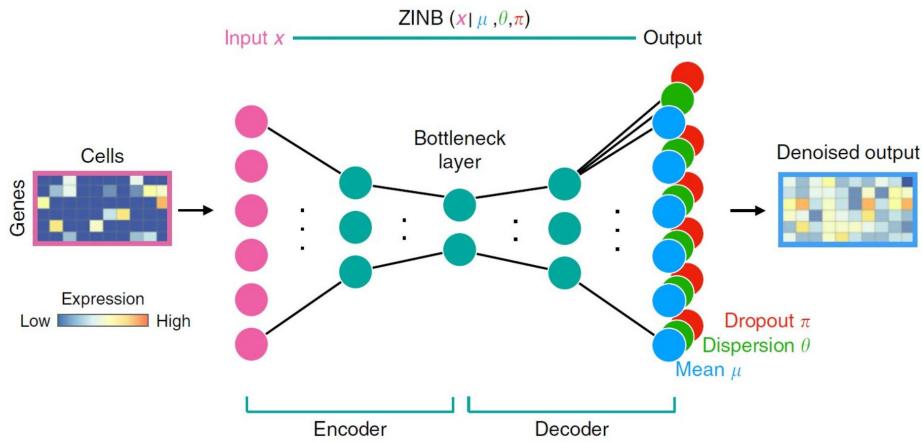


Figura 11.1: Schema del funzionamento di *DCA*. In input si ha la matrice con indicati i livelli di espressione genica. Si ha poi l’*autoencoder* e infine l’output senza rumore.

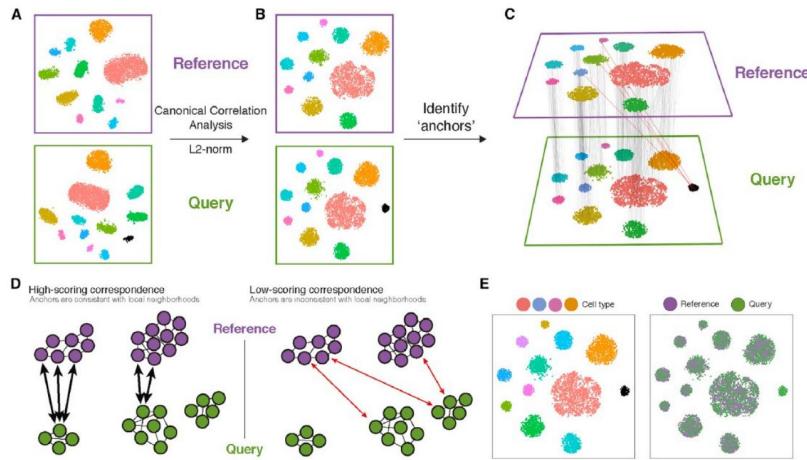
11.1.3 Batch Effect

Abbiamo visto quindi alcune tecniche per “pulire” un dataset ma il problema è che i dati provengono da diverse misurazioni, anche in un setup semplice che non considera i timepoint. Un sequenziamento ad un paio di ore di distanza da un altro, a causa di effetti esterni, può portare a risultati con livelli di dropout diversi e con un rumore diverso (magari anche solo, anche se è un caso estremamente limite, per come sono state trasportate le cellule da una stanza all’altra). Si creano quindi diversi **batch** (*lotti*) di misure, parlando quindi di **batch effect** e di conseguenza di **batch correction**, tramite tecniche computazionali per rimuovere il rumore. Se fino ad ora si è parlato di rumore all’interno di una singola misurazione ora si parla di rumore tra più misurazioni.

Una delle tecniche usate è detta **UMAP**, che offre molti vantaggi rispetto alla più storica **t-SNE**, che permette di fare i plot di un *batch* rispetto agli altri.

Uno dei tool più usati in questo contesto è **Seurat**, il cui schema generale è visualizzabile in figura 11.2². Per funzionare parte da alcune reference, si controllano, per una certa query, alcune correlazioni canoniche con il reference e si identificano i cosiddetti **anchors** tra i due dataset. Ci sono infatti alcuni “punti” all’interno del dataset che corrispondono alle cellule. Si cerca un mapping tra le cellule della query, in certe posizioni, e quelle del reference,

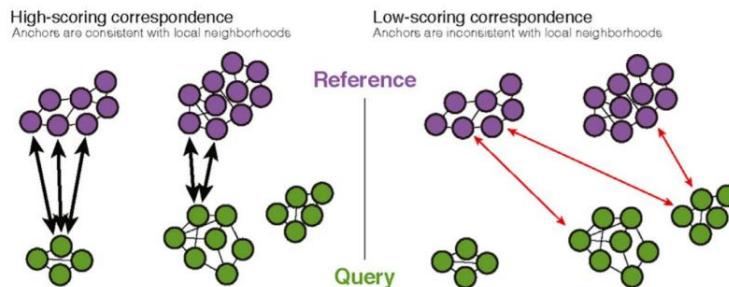
²Stuart, Tim, et al. "Comprehensive integration of single-cell data." Cell 177.7 (2019): 1888-1902.

Figura 11.2: Schema del funzionamento di *Seurat*.

che sono comunque generalmente in altre posizioni. Come visualizzabile in figura 11.2, nella sezione *C*, questo mapping si fa tra i due piani ottenuti coi punteggi ricavati in precedenza tramite *UMAP*. Si hanno quindi queste *anchors cell* e da queste si costruiscono le corrispondenze, in modo abbastanza complesso, facendo poi un mapping tra query e reference sullo stesso piano, al fine di ottenere sottogruppi completamente separati. Le *anchor* quindi rappresentano due cellule, una da ciascun dataset (query e reference), che si prevede provengano da uno stato biologico comune e sono rappresentate formalmente come un a tupla del tipo:

$$(c_i^1, d_j^2)$$

Le *anchor* vengono associate ad un punteggio di score calcolato tramite l'overlap condiviso con i vicini tra le *anchor* e le cellule di query:



Questo score viene usato per calcolare una matrice di pesi W , che definisce la forza dell'associazione tra ogni cellula di query e ogni *anchor*. I pesi si

basano sia sulla distanza tra ciascuna cellula di query e l'*anchor* sia sul punteggio dell'*anchor*.

Si hanno quindi tre step se astraiamo molto l'algoritmo:

1. si calcola la differenza tra ogni coppia di cellule *anchor* a :

$$B = Y[, a] - X[, a]$$

con X matrice della query (???) e Y matrice delle espressioni originale

2. si calcola la matrice di trasformazione C usando B e la matrice di pesi W trasposta:

$$C = BW^T$$

3. si sottrae C a Y per ottenere la matrice con il punteggio combinato:

$$Y^{int} = Y - C$$

11.1.4 Copy Number Alterations

Una delle altre cose da fare parlando di analisi *single-cell* è quella di identificare le **copy number alterations (CNA)**. Anche questa è un'operazione abbastanza complessa e si risolve con vari tool ormai standard.

Bisogna prima però capire meglio di cosa si parla nominando le *CNA*. Durante la replicazione del DNA, le regioni del genoma possono essere copiate più volte a causa di errori o instabilità. Le variazioni nelle **copy number (CN)** possono interessare:

- intere “braccia”
- poche basi, parlando in questo caso di *focali/focal*

Nelle cellule diploidi si hanno generalmente due *CN*, essendoci due copie di ogni gene. Le alterazioni includono *gain* o *loss*. Le variazioni non somatiche di *CN*, ovvero le alterazioni, identificano le sotto-popolazioni. Si ha che la distribuzione delle *read depth*, ovvero il numero di volte in cui una particolare base è rappresentata all'interno di tutte le read del sequenziamento, sul genoma senza alterazioni mostra dei “salti”. Regioni con diversi *CN* sono separate dai cosiddetti **breakpoint**. I *breakpoint* permettono di raccogliere insiemi/bin di read vicine in regioni genomiche con lo stesso stato di *CN*. Si procede quindi identificando i *breakpoint* come limiti dei bin dove la *read depth* cambia tra sottoinsiemi di cellule. Per farlo:

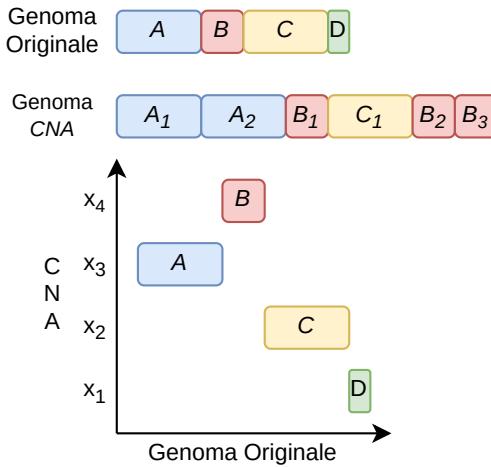


Figura 11.3: Schema astratto del funzionamento delle *CNA*, con le varie copie delle regioni (tranne *D* che viene persa). Si noti che *C* non subisce alterazioni

- si comincia con un controllo della qualità dei dati *scDNAseq*, con la fase di *quality check*. La prima cosa da fare è essere certi di osservare solo regioni con un'alta *mappability*, che è una funzione che dice quante read possono mappare in una certa regione, superiore al 90%. Le altre regioni sono scartate.

Si procede poi con la *GC correction*, avendo che il conteggio di read per cella e per “finestra” che presenta una diversa frazione di *GC content*, ovvero la percentuale di Guanina e Citosina, viene ridotto. Si ottengono quindi regioni con circa lo stesso *GC content* per evitare problemi.

Si ha inoltre una pulizia del rumore. Le cellule con alta variabilità all'interno di un bin di read coverage vengono scartate, al fine di avere:

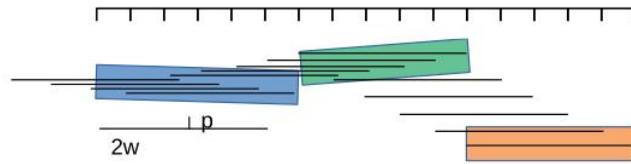
Depth Independent Median Absolute deviation of Pairwise Differences $>\sim 1$

- si modella la distribuzione delle read (di basi si parte dicendo che è una distribuzione uniforme a meno che non si sappia che alcune regioni hanno una distribuzione maggiore di read). Per ogni cellula j i conteggi delle read nel bin p , che chiamo z_{jp} , sono modellati con un binomiale negativo per tenere conto della sovradispersione. Si ha quindi:

$$P(N = z_{jp}) \sim NB(\lambda, \nu)$$

Si ha quindi il cosiddetto **0th CN model**, dove per un dato bin alla posizione p , si fissa una dimensione della finestra $2w$, e si considerano i bin da $p-w+1$ a $p+w$ e quindi si osserva l'evidenza

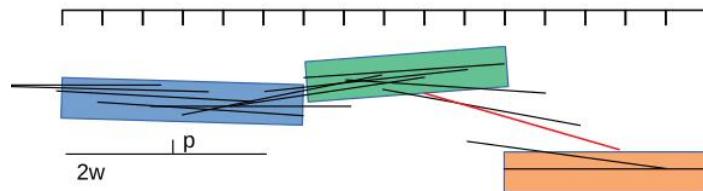
di un breakpoint. Se non vi è alcuna modifica del *CN* dopo il bin p , i conteggi previsti sono uguali tra tutti i bin nella finestra. Si ha quindi il *0th CN model*:



Il tutto viene fatto calcolando la *log-verosimiglianza* (**formula non spiegata a lezione**):

$$l(z_{j,p}; \lambda, \nu) = \left(\sum_{i=p-w+1}^{p+w} z_{i,p} \right) [\log(\lambda) - \log(\lambda + \nu)] - 2w\nu \log(\lambda + \nu)$$

- per ogni cellula si confronta ogni bin di un modello *CN costante*, ovvero l'ipotesi “stabile”, con un modello *CN step changing*, ovvero con l'ipotesi di un modello “non stabile”. Parlando modello *CN costante* per consentire il rumore che fornisce conteggi sbilanciati su entrambi i lati di p anche senza breakpoint, si utilizza e adatta un modello lineare per i conteggi attesi, usando un'ipotesi nulla costante:



Avendo (**formule non spiegata a lezione**):

$$l(z, \beta, \beta, \nu) = \sum_{i=p-w+1}^{p+\infty} z_{i,\beta} [\log(\lambda_i) - \log(\lambda_i + \nu)] - 2\omega\nu \log(\lambda_i + \nu)$$

con:

$$\lambda_i = \alpha + \beta(i - p)$$

Inoltre per evitare di adattare modiche reali al CN con il modello lineare, la pendenza è limitata così come la variazione relativa attraverso p :

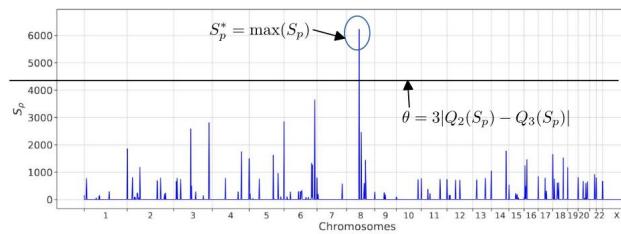
$$\frac{\lambda^* - \alpha^*}{w} = \beta^* < \frac{1}{4}$$

In pratica si limitano i cambiamenti. Se si osserva una pendenza come quella rossa nell'ultima figura si ha un breakpoint.

Su slide altri dettagli su questa parte non trattai in aula

- si procede combinando il segnale su tutte le cellule
- i bin con il segnale più forte rispetto a una soglia di rumore sono classificati come breakpoint.

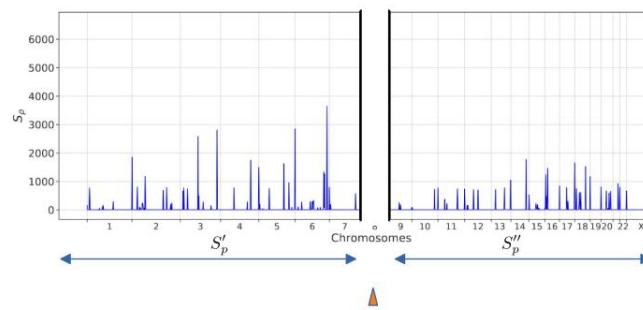
Si cercano dei *picchi* in ogni cromosoma:



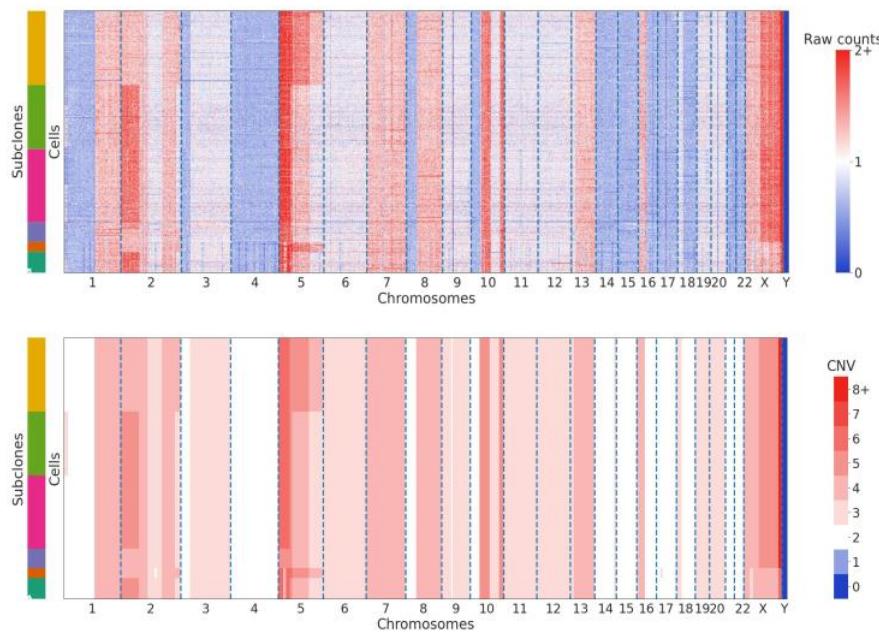
e si procede usando la seguente misura (**formula non spiegata a lezione**):

$$S_p = \log [1 - P_p(k^*)|ev]$$

Si cerca quindi il massimo valore S_p^* e se questo è maggiore di una certa soglia ϑ si ha che p è un breakpoint e quindi si possono cancellare i vicini di p . A questo punto S_p viene diviso in due sottoparti in p : S'_p e S''_p , ripetendo quanto fatto per ogni sottoparte fino a che ogni $S_p < \vartheta$, per ogni S_p ottenuto con le divisioni in sottoparti:



Infine si procede facendo *clustering* e identificando i subcloni. Ogni profilo di conteggio di read single-cell per bin viene raccolto in un profilo di regione delimitato da breakpoint e il conteggio delle read per ogni profilo di regione viene clusterizzate. Per ogni cromosoma si ha quindi il numero di cellule che ha un numero di read che “supporta” il breakpoint. I centroidi del cluster rappresentano i profili *CN*:



Questo non è comunque l'unico modo di identificare i *breakpoint* e le *CNA*.

11.2 Pipeline

Possiamo quindi descrivere la **pipeline** per l'analisi di dati *single-cell*. Essa conta di quattro step:

1. la fase di *quality check*, dove si rimuovono le osservazioni di bassa qualità
2. la fase di *denoising* e *rimozione dei batch effect* (*di cui si è già discusso in realtà nella sezione precedente*)
3. la fase di *data integration* e *clustering*
4. la fase di *analisi delle differenze d'espressione*

11.2.1 Quality Check

Prima di procedere con l’analisi dei dati, la matrice di conteggio prodotta dopo un esperimento *scRNAseq* deve subire una fase di *preprocessing* per rimuovere le osservazioni di bassa qualità, ovvero:

- **doublets (doppietti)**, ovvero quando due cellule sono isolate insieme
- **empty droplets (gocce vuote)**, ovvero gli elementi della matrice che hanno un numero estremamente basso di conteggi, se non nullo. Sono le gocce che non contengono cellule o ne contengono pochissime

Invece di utilizzare un cutoff sulla distribuzione dei conteggi *UMI*, è possibile identificare i doublets utilizzando i loro profili RNA, infatti esso si verifica quando due cellule vengono catturate nella stessa goccia d’olio. Questo fattore di confusione cresce con il carico di incapsulamento delle gocce. La formazione di doublet è mitigata riducendo la concentrazione delle cellule di input molto al di sotto del possibile teorico, limitando però così il throughput dell’esperimento *scRNAseq*.

Dopo il filtraggio genico e cellulare dei conteggi “grezzi”, viene generata una percentuale pN di doublets artificiali selezionando casualmente due cellule e calcolando la media dei loro profili. Questo viene fatto solitamente per controllo. Ad esempio, presi due droplet b e c tali che:

$$\vec{g}_b = \{g_{b1}, g_{b2}, \dots, g_{bM}\}$$

$$\vec{g}_c = \{g_{c1}, g_{c2}, \dots, g_{cM}\}$$

si ottiene il droplet artificiale a tale che:

$$\vec{g}_a = \frac{\vec{g}_b + \vec{g}_c}{2}$$

Le cellule artificiali e reali vengono fuse insieme per ulteriori trasformazioni, come la *log normalization*, correzioni per *cicli cellulari* (eliminando le cellule che sono nello stato di riproduzione), correzioni per *batch effect*, e riduzioni dimensionali.

Per ogni cellula reale nel campione C la proporzione di quelle artificiali, $pANN$, nel “vicinato, di grandezza pK , viene quindi calcolato.

I doublets sono identificati tramite una fase di ranking e di studio delle soglie sui valori $pANN$, in accordo con il valore atteso della cardinalità dei

doublets, ovvero $nExp^3$. Il punto chiave è comunque quello di gestire tutti i problemi derivanti dai doublets.

Un altro punto chiave del *quality check* è capire, di tutti i conteggi in una cellula, quanti sono associati ai trascritti localizzati nei *mitochondri*, che sono gli organelli, a doppia membrana, addetti alla respirazione cellulare, costituiti da sacchette contenenti enzimi respiratori. I mitocondri hanno il loro materiale genetico. Quando una cellula va incontro ad apoptosi o lisi, i suoi trascritti citoplasmatici vengono rilasciati, mentre i trascritti mitocondriali possono rimanere all'interno della doppia membrana dei suoi mitocondri, pertanto, un basso numero di conteggi e un'alta frazione di conte mitocondriali sono insieme indicatori di una cellula stressata o addirittura morta e quindi queste cellule dovrebbero essere rimosse.

11.2.2 Data Integration & Clustering

Avendo già visto la fase di *denoising* e di *batch effect removal* nella sezione precedente si passa direttamente alla terza fase della pipeline, quella di **data integration & clustering**.

Data Integration

Parlando di *data integration* si ha che l'identificazione delle sottopopolazioni cellulari in base alle differenze clonali consente ulteriori analisi a valle che non sono facilmente individuabili dalla semplice clusterizzazione dell'espressione genica. Quindi partendo dal presupposto che in una cellula esista una *relazione lineare* tra l'espressione genica *log-normalizzata* e il corrispondente *CN*, ne segue che *scRNA* e *scDNA (CN)*, campionati indipendentemente, possono essere integrati per arricchire i dati. La relazione lineare risulta sensata quando si ha una duplicazione in un gene che si aspetta duplichì la quantità di trascritto (??), quindi se ho tre *CN* ma ho già tre copie del gene allora ci si aspetta di avere tre volte il numero di trascritti. È quindi una buona assunzione.

Utilizzando profili *CN* determinati da approcci alternativi (come visto nella precedente sezione), ciascun profilo di espressione genica può essere associato a un profilo *CN*. Quindi se si hanno geni e cellule a partire da un esperimento di *scRNAseq* posso arricchire tale informazione, ad esempio, con gene, cloni e *CN* ottenuti da un esperimento di *scDNAseq*⁴.

³Per maggiori dettagli: McGinnis, C. S., cels., 8(4), 329–337 (2019). doi: 10.1016/j.cels.2019.03.003

⁴Per approfondire: Campbell K. R., Genome Biol., 20(1), 1–12 (2019). doi: 10.1186/s13059-019-1645-z

Dato un modello binomiale negativo per la distribuzione delle read nella cellula n al gene g :

$$P(y_{ng}) \sim NB(\mu, p)$$

si ha che la media dei conteggi RNA per un profilo CN $z_n = c$ è modellata come il seguente valore atteso (con di base la relazione “una copia implica un’unità di espressione”):

$$\mathbb{E}(y_{ng}|z_n = c) = K \cdot s_n \cdot \mu_g \cdot f(\lambda_{n,g}) \cdot e^{X_N \cdots \beta_g}$$

dove:

- s_n è la read depth della cellula
- μ_g è l’espressione per copia
- $\lambda_{n,g}$ è il CN
- $X_N \cdots \beta_g$ è l’espressione residua

Si ha inoltre che la variabilità residua dell’espressione genica non spiegata semplicemente dal CNA è rappresentata dal rumore strutturato (quindi rumore inserito nei calcoli). **Su slide, lezione 14, approfondimento del discorso non trattato a lezione.**

Fissato il numero di variabili latenti k e la funzione di dosaggio f , l’inferenza dei vari parametri (**indicati su slide ma non approfonfiti**) viene eseguita utilizzando il cosiddetto metodo *mean field variational Bayes*. In pratica si combinano le read e le CNA per inferire i cloni. Inoltre bisogna testare l’approccio Bayesiano per inferire la distribuzione delle variazioni, ovvero i cloni inferiti, per mezzo di cloni “test”, i cosiddetti *ground truth clones*.

Clustering

Attualmente, lo stato dell’arte per il clustering *scRNaseq* è un algoritmo progettato per il rilevamento di gruppi di nodi su grafi, ovvero l’**algoritmo Leiden**⁵, che crea un **grafo KNN** per *single-cell* e quindi rileva gruppi di nodi densamente connesse. Un **grafo KNN** è un grafo in cui due vertici p e q sono collegati da un arco, se la distanza tra p e q è tra le k -esime distanze più piccole da p ad altri elementi. L’idea è quindi identificare sottogruppi nel grafo che sono densamente connessi, usando quindi una misura per capire la distanza tra un nodo e un altro. L’idea dell’algoritmo è di spostare i vari nodi per capire come sono clusterizzabili i nodi.

⁵V.A. Traag et al., 2019 <https://doi.org/10.1038/s41598-019-41695-z>

Nell'algoritmo si usa il concetto di *modularity* come misura per ottimizzare la computazione per distinguere gruppi di nodi in reti complesse, infatti valori di *modularity* alti indicano connessioni dense all'interno di un gruppo e connessioni sparse tra nodi appartenenti a gruppi diversi. L'ottimizzazione della *modularity* è *NP-hard* e quindi l'algoritmo usa un'euristica:

1. sposta ogni nodo e poi calcola il guadagno in *modularity* che accadrebbe se un nodo venisse spostato dal suo gruppo attuale ad un altro. Accetta solo la mossa con il guadagno positivo più alto e se nessun guadagno positivo, lascia il nodo nel gruppo orinale
2. ricalcola i vari gruppi dopo le modifiche fatte
3. ricostruisce la rete
4. ricomincia fino a che non si hanno più cambiamenti, ovvero fino a quando il punteggio globale resta costante o quasi

Tale algoritmo è ad esempio incluso, insieme a molti altri, nella libreria *Scanpy* per *Python*.

11.2.3 Differential Expression Analysis

Dati diversi gruppi di cellule, è possibile identificare quali geni caratterizzano ciascun gruppo (identificato a sua volta tramite l'*algoritmo Leiden* eseguito sui risultati prodotti da *UMAP* a partire dal dataset con le varie single-cell, ovvero le associazioni tramite *tag* tra cellule e livelli di espressione), identificando ad esempio i geni differenzialmente espressi. Si parte sempre da una lista di geni scelti a priori.

Si usa poi il **test di Wilcoxon**, un test di ipotesi statistica non parametrico utilizzato per testare la posizione di un insieme di campioni o per confrontare le posizioni di due popolazioni utilizzando un insieme di campioni appaiati, che viene in questo caso usato o per verificare se un gene è distribuito in modo differenziale tra due gruppi. Questo test è il più usato.

Negli ultimi due anni sono stati proposti diversi metodi per tenere conto della distribuzione dei geni in un gruppo di single-cell. Solitamente si ottengono istogrammi, per ogni gene, come quello in figura 11.4, dove a sinistra si ha una barra per le cellule che non esprimono nulla mentre per il resto si hanno le espressione per il conteggio log-normalizzato.

I dati sono assunti già puliti in questa fase.

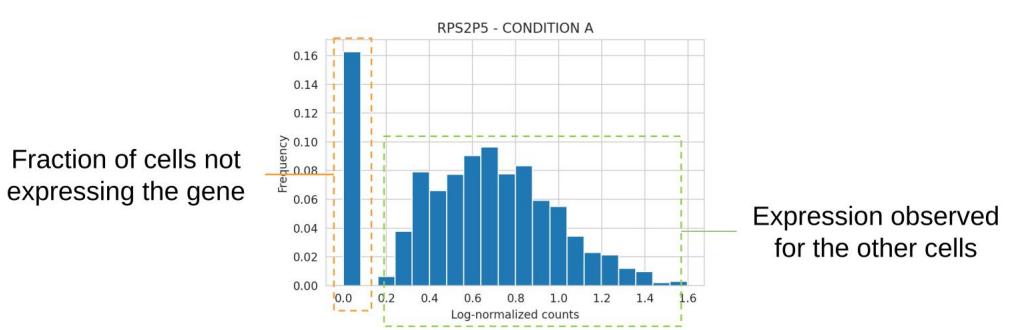


Figura 11.4: Esempio di risultato dopo il *test di Wilcoxon*

Capitolo 12

Control Theory in Computational Biology

Questa coppia di lezioni/seminario non è stata rimaneggiata negli appunti in quanto le slide, recuperabili su Moodle, sono assolutamente esaustive.