Università degli Studi di Milano - Bicocca
**Scuola di Scienze**
**Dipartimento di Informatica, Sistemistica e Comunicazione**
**Corso di laurea in Informatica**

# Inferenza di alberi tumorali tramite Particle Swarm Optimization

**Relatore:** Prof. Della Vedova Gianluca
**Correlatore:** Dott. Ciccolella Simone

**Relazione della prova finale di:**
Castro Tenemaya Adrian David
Matricola 816015

**Anno Accademico 2016-2019**

# Indice

# Premessa e ringraziamenti

Il presente lavoro è frutto del lavoro svolto come tirocinio all'interno dell'Università di Milano-Bicocca, e viene anche utilizzato come tesi finale ai fini del conseguimento della laurea in Informatica. È però necessario chiarire che il progetto in questione non sarà abbandonato nè una volta terminata la stesura di questa relazione, nè dopo il conseguimento della laurea. È mia intenzione contribuire al meglio delle mie possibilità in quello che ritengo essere uno dei campi con il quale mi sento più legato, sia a livello di interesse professionale, che a livello strettamente personale: la ricerca sul cancro. Secondo il *National Cancer Institute*, nel 2012 sono stati riportati 14.1 *milioni* di nuovi casi, e di questi, 8.2 *milioni* hanno portato alla morte [1]. I dati mostrano anche quelli che può sembrare all'apparenza una realtà discordante: il numero totale di morti per cancro è in crescita, ma il rapporto delle morti per individuo sta calando [5]. Nel 1990, 161 persone su 100.000 nel mondo sono morte a causa del cancro. Nel 2016, questo numero è calato a 134 su 100.000. Questo miglioramento è dovuto indubbiamente ad un numero molto elevato di fattori, tra cui l'aumento della qualità di vita ed un migliore sistema sanitario, ma è anche grazie alla crescita incessante della ricerca sul cancro, ed ai campi sui quali essa si appoggia. Lo sviluppo di algoritmi sempre più efficienti e performanti, e l'utilizzo di calcolatori super-veloci, ha permesso a questo settore di ricerca di ottenere dei considerevoli risultati.

Con questo progetto spero, quindi, di aver dato un contributo in questo settore, anche se in una percentuale minuscola.

Vorrei ringraziare mia mamma **Laura**, mio padre **José**, mia sorella **Valeria**, i miei fantastici nonni, e tutte le bellissime e meravigliose persone che hanno contribuito, in maniera diretta ed indiretta, a farmi appassionare all'informatica e, in questo caso, alla bioinformatica.

# Prefazione

Il presente lavoro è stato svolto sotto la guida ed il supporto di AlgoLab, laboratorio presso il dipartimento di informatica dell'Università di Milano-Bicocca, che ha lo scopo di progettare, studiare, analizzare ed implementare algoritmi efficienti per problemi computazionali. Il tirocinio è cominciato il 22 Marzo 2019, ed è stato condotto per la maggior parte in maniera autonoma, da remoto. Il problema affrontato è l'*inferenza di progressioni tumorali* su dati single-cell, al fine di determinare l'ordine e la frequenza con cui le variazioni somatiche vengono acquisite durante una progressione tumorale. Spesso ciò è basato sulla "Infinite Sites Assumption", dove le mutazioni possono solo essere acquisite, e mai perse. Lo stage si colloca nella ricerca del superamento di tale assunzione, utilizzando il modello della *filogenesi persistente*, dove ogni mutazione può essere persa al massimo una volta nell'intero albero. Più precisamente, si è investigata la tecnica *Particle Swarm Optimization*, un algoritmo di ottimizzazione di tipo euristico, ispirato al movimento degli sciami. I dati single-cell sono caratterizzati da un elevato tasso di errore e di valori mancanti: ciò rende inutilizzabili gli approcci noti in letteratura per i dati di *bulk sequencing*. In particolare, sono state analizzate quali strutture dati utilizzare per rendere l'algoritmo efficiente ed efficace, e quali operazioni considerare per inferire predizioni accurate.

# Capitolo 1

# Introduzione

## 1.1 Descrizione

Recenti sviluppi nel trattamento mirato di questo gruppo di malattie fa affidamento sull'accurata inferenza della progressione e dell'evoluzione del cancro [2], rivelandosi una . Il cancro è la seconda causa più comune di morte [5], arrivando nel 2017 a contare il 17.08% delle morti nel mondo, per un totale di 8.93 *milioni* di decessi.

## 1.2 Storia

Era il 1869 quando venne isolato per la prima volta nella storia dell'umanità l'*Acido Desossiribonucleico*, anche conosciuto come *DNA*. Il pioniere di questa scoperta è Friedrich Miescher, medico e ricercatore nato in Svizzera nel 1844. Durante il processo di scoperta, Miescher aveva realizzato che nonostante avesse proprietà simili alle proteine, la nuova sostanza – il DNA – non lo era. Prima di isolare le cellule dal pus presente nelle bende chirurgiche dell'ospedale in cui lavorava, Miescher fu molto attento ad assicurarsi che il materiale che stava utilizzando fosse fresco e non contaminato. Fu solo più tardi, nel 1871, che il ricercatore iniziò a lavorare sullo sperma di salmone, una specie di pesce che affluiva numerosa durante il periodo autunnale nella città di Basel.

## 1.3 Nozioni di biologia

Al fine di poter comprendere appieno il lavoro svolto, in questa sezione verranno trattate nozioni base di biologia, partendo dalla cellula fino alla rappresentazione in modello del DNA in essa contenuta.

### 1.3.1 La cellula

Le cellule costituiscono le fondamenta di tutti gli organismi viventi. Il corpo umano è composto da trilioni di cellule. Esse danno forma al corpo, estraggono le sostanze nutritive dal cibo, convertono quelle sostanze nutritive in energia, ed hanno delle funzioni specifiche. Le cellule contengono anche il materiale ereditario del corpo, e possono fare copie di loro stesse [6]. Esse sono a loro volta costituite da diverse parti, tra le quali analizzeremo il nucleo e ciò che esso contiene, il DNA.

### 1.3.2 Il DNA

Il *DNA*, o *acido desossiribonucleico*, è il materiale erediratio negli umani e quasi tutti gli altri organismi viventi. Quasi ogni cellula presente all'interno del corpo umano ha lo stesso identico DNA. La maggior parte del DNA è situata all'interno del nucleo della cellula (dove è chiamato *DNA cellulare*), ma può trovarsi anche all'interno dei mitocondri, organi cellulari addetti alla respirazione cellulare. Le informazioni nel DNA sono conservate come un codice formato da quattro componenti chimici base (anche dette basi azotate): **adenina** (A) (Figura 1.2), **guanina** (G) (Figura 1.3), **citosina** (C) (Figura 1.4), e **timina** (T) (Figura 1.5). L'ordine, o la sequenza, di queste basi determina le informazioni disponibili per costruire e mantenere operativo un organismo.

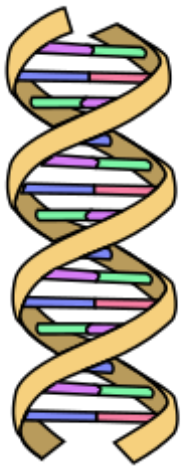

Figura 1.1: Il DNA

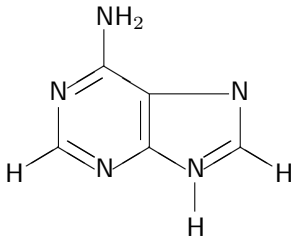


Figura 1.2: Adenina (A)

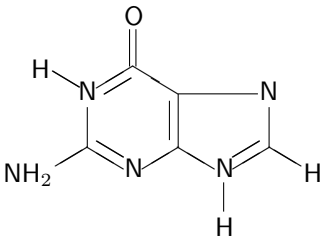


Figura 1.3: Guanina (G)

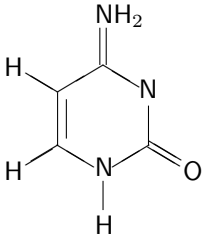


Figura 1.4: Citosina (C)

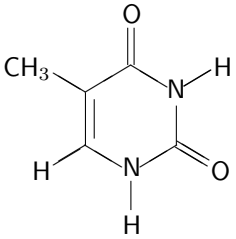


Figura 1.5: Timina (T)

Le basi del DNA si combinano tra di loro, A con T e C con G, in maniera tale da formare unità chiamate *coppie base*. Assieme ad una molecola di zucchero ed una di fosfato, le basi costituiscono quello che è definito un *nucleotide*. I nucleotidi sono disposti in due lunghi fili che formano una spirale chiamata *doppia elica*.

Un'importante proprietà del DNA è che si può replicare, o fare copie di se stesso. Uno qualunque dei due fili di DNA [7] può essere utilizzato nel processo di duplicazione per ottenere una copia identica del DNA di partenza. Questa è una fase cruciale nella divisione di una cellula, poiché la nuova copia di essa deve avere lo stesso identico DNA della cellula di origine.

### 1.3.3 Cancro e tumore

La fase di divisione di una cellula è cruciale. Si stima che durante la replicazione, solo una base su $10^9$ [3] sia errata. Vari fattori possono inoltre influenzare questa delicata fase,

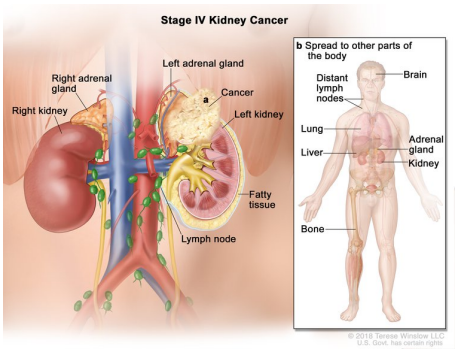


Figura 1.6: Cancro al rene

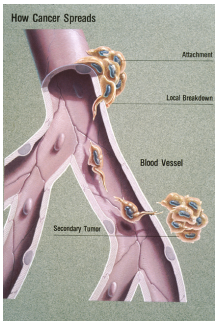


Figura 1.7: Diffusione del cancro

come l'esposizione ad agenti chimici ed irradiazione. Questi errori molto spesso sono corretti in vari modi, ma quando questo non basta, possono essere la causa scatenante che porta una cellula a diventare *cancerogena*. In generale, una cellula è cancerogena quando inizia a moltiplicarsi senza controllo. Quando questo processo avviene in un tessuto solido come un organo (Figura 1.6), muscolo od ossa, prende il nome di *tumore*. Ci sono due tipi di tumore: *maligno* (cancerogeno) e *benigno* (non cancerogeno). I primi possono invadere i tessuti circostanti nel corpo, e mentre crescono, alcune cellule possono viaggiare attraverso il sangue (Figura 1.7) o altri mezzi a formare delle *metastasi*, dei tumori secondari [8], mentre gli ultimi conservano le caratteristiche del tessuto di origine e non hanno la tendenza di invadere gliorgani circostanti. Un tumore benigno non è quindi un cancro, ma solo una massa che può raggiungere dimensioni considerevoli, ma non si diffonde in altre parti del corpo.

## 1.4 Modelli di sostituzione

In filogenetica il DNA può essere rappresentato come una sequenza di simboli, utilizzando le basi (sottosezione 1.3.2) corrispondenti alle posizioni degli allineamenti come caratteri.

AGTCCAGGACAT GGCATTCAATCA

Figura 1.8: Esempi di sequenze di DNA

La Figura 1.8 rappresenta un esempio di *modello di sostituzione*, un modello che in biologia descrive il processo per cui una sequenza di simboli cambia in un'altra, modificandone i tratti che rappresenta. In cladistica[1] viene utilizzato per rappresentare delle caratteristiche presenti, utilizzando il carattere "1", o assenti, utilizzando il carattere "0", in una specie.

10011 01110

Figura 1.9: Esempio di modello di sostituzione in cladistica

L'esempio in Figura 1.9 può ipoteticamente rappresentare due specie: la prima può digerire i latticini, non depone uova, è una creatura a sangue freddo, vola e sa nuotare; la seconda non può digerire i latticini, può deporre uova, è una creatura a sangue caldo, vola e non sa nuotare. Lo stesso ragionamento può essere utilizzato per rappresentare le *mutazioni* presenti all'interno di una cellula nel caso di cellule tumorali (Figura 1.10).

[1]Metodo di classificazione degli esseri viventi che si basa sul grado di parentela, ovvero sulla distanza nel tempo dell'ultimo progenitore comune

| BBS4 | CAMSAP1 | DOCK3 | EPHA10 | EYA4 | HIPK4 | HIST1H2AG | INTS8 | MAL2 | MYOM3 | OAZ3 | PPIG | PTPRQ | RGS11 | RYR3 | SERPINF2 | SMOC1 | TTN | TUFT1 | ZNF540 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |

Figura 1.10: Dataset di cellule tumorali e delle relative mutazioni

## 1.5 Richiami di ottimizzazione matematica

In questa sezione verranno trattate inizialmente nozioni base di ottimizzazione matematica, in particolare sulla ricerca locale dell'ottimo, per poi introdurre due tecniche di ottimizzazione che verranno utilizzate nel Capitolo 2.

### 1.5.1 Hill climbing

In matematica, *hill climbing* è un algoritmo di ricerca dell'ottimo, migliorando la soluzione ripetutamente fino a quando non si raggiunge un criterio di ottimalità. L'idea è quella di partire da una soluzione sub-ottimale, che per analogia viene paragonato al partire alla base della collina, per poi migliorare la soluzione ottimale, che viene comparato allo scalare la collina, fino al raggiuntimento di una condizione, cioè raggiungere la cima della collina. In maniera generale, si può modellare nella forma descritta in Algoritmo 1.

**Algoritmo 1:** Hill Climbing

```
1  inizializzazione
2  while non raggiunta condizione di ottimalità do
3  |  seleziona e applica nuova operazione
4  |  if nuovo stato è ottimo then
5  |  |  termina
6  |  end
7  |  if nuovo stato è migliore del precedente then
8  |  |  stato = nuovo stato
9  |  end
10 end
```

Esistono numerose variazioni di questo algoritmo, ma le più conosciute sono *simple hill climbing*, *steepest hill climbing* e *stocasting hill climbing*, applicate a seconda delle proprietà del problema in questione. Un esempio di funzione da ottimizzare, in questo caso massimizzare, può essere come quella rappresentata in Figura 1.11, dove esiste un

solo ottimo locale ($f(x, y) = 0$) cioè la funzione è monomodale[2]. In questo caso, il *simple hill climbing* e lo *steepest hill climbing* ottengono sempre il risultato migliore.

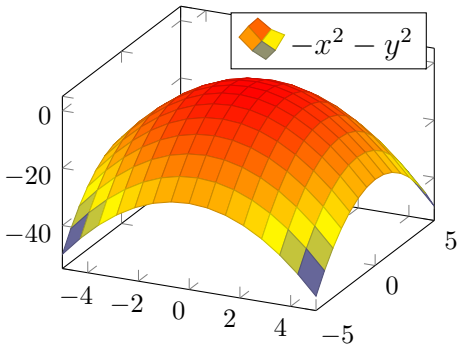


Figura 1.11: Paraboloide, monomodale

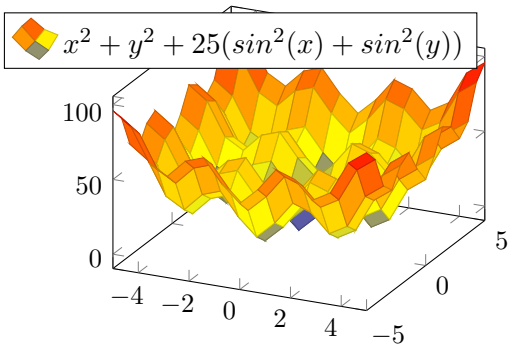


Figura 1.12: "Eggcrate", plurimodale

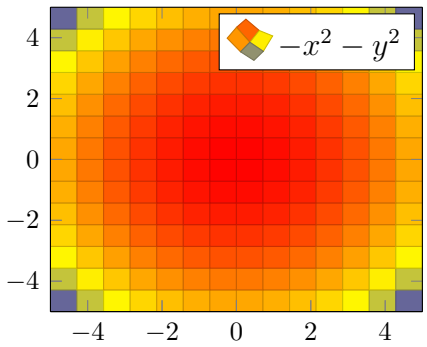


Figura 1.13: Paraboloide, monomodale

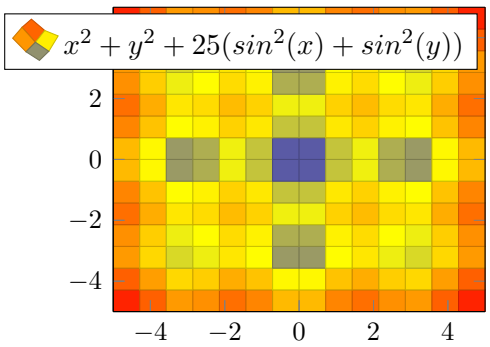


Figura 1.14: "Eggcrate", plurimodale

D'altro canto, funzioni plurimodali[3] come "Eggcrate" rappresentato in Figura 1.12, dove le tecniche citate precedentemente falliscono miseramente. Entrano quindi in gioco algoritmi di ricerca locale che ammettono, con un certo grado di libertà, di accettare un risultato peggiore di quello attuale, nella speranza di non rimanere intrappolati in un ottimo locale.

### 1.5.2 Simulated Annealing

Per cambiare e migliorare delle caratteristiche di un solido, in metallurgia viene utilizzata la tecnica della *ricottura* (in inglese "anneal"), dove i solidi come l'acciaio, bronzo o alluminio, vengono portati ad altissime temperature, per essere poi raffreddati ad una certa velocità chiamata *cooling rate*, che determina le caratteristiche finali del metallo. Alle alte temperature, gli atomi si muovono molto velocemente e rompono le strutture cristalline che avevano formato precedentemente. Mano a mano che la temperatura cala, gli atomi rallentano, e si recristallizzano.

Analogamente, la tecnica matematica del *simulated annealing* è un algoritmo di ricerca che utilizza la temperatura per riuscire a scappare da eventuali ottimi locali. Quando la temperatura è alta, l'algoritmo è meno propenso ad accettare nuovi soluzioni, anche se migliori. Mano a mano che la temperatura diminuisce, la probabilità che si accetti una

[2]Una funzione monomodale è una funzione con un solo ottimo locale, che corrisponde anche all'ottimo globale della funzione stessa

[3]Una funzione plurimodale è una funzione con più di un ottimo locale. Non è detto che esista un unico ottimo globale

soluzione migliore aumenta, fino al raggiuntimento della condizione di ottimalità, cioè quando la temperatura non può più scendere, ed il risultato non può che essere un ottimo locale.

**Algoritmo 2:** Simulated Annealing

```
1  best ← random()
2  T ← 1.0
3  T_min ← 0.0001
4  cooling_rate ← 0.9
5  while T > T_min do
6      new_best = neighbour(best)
7      ap ← acceptance_probability(best, new_best, T)
8      if ap > random() then
9          best ← new_best
10     T = T * cooling_rate
```

Questa funzione però molto spesso fallisce, specie se si cerca di ottimizzare una funzione plurimodale come in Figura 1.12.

### 1.5.3 Particle Swarm Optimization

Un'altra tecnica di ricerca dell'ottimo è quella del *particle swarm optimization*. Questo algoritmo iterativo nasce inizialmente come simulazione del comportamente sociale di stormi di uccelli che si sincronizzano in volo, o un branco di pesci alla ricerca di cibo [4]: membri individuali del branco possono trarre vantaggio dalle scoperte ed esperienze passate di tutti gli altri membri durante la ricerca del cibo, un vantaggio che può rivelarsi decisivo per superare la competizione. Questa è un'ipotesi fondamentale per poter definire il *particle swarm optimization*.

Nell'algoritmo, il branco di pesci viene sfruttato come analogia per lo *swarm* (rappresentato in Figura 1.15), che indica l'insieme degli elementi appartenenti ad una popolazione, questi indicati come *particelle* dello swarm.

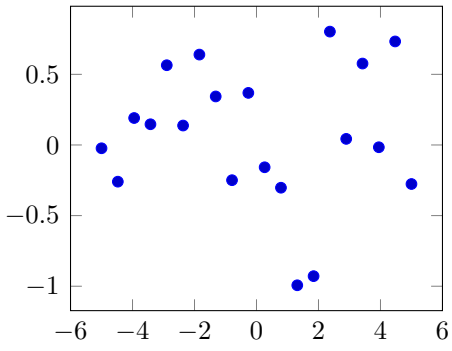


Figura 1.15: Esempio di swarm

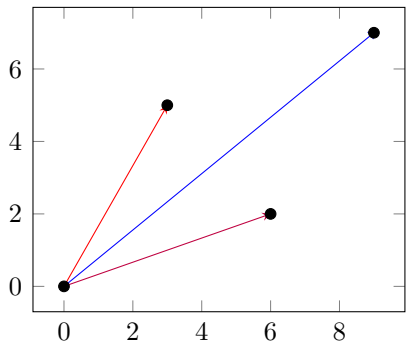


Figura 1.16: Somma vettoriale tra $p_i$ e $g$

Ad ogni iterazione, la posizione di ogni particella viene aggiornata per ogni dimensione del problema, basandosi sulla migliore posizione della particella $p_i$ e sulla migliore posizione globale dello swarm $g$ (in Figura 1.16 si vede un esempio di questa combinazione, dove la posizione finale del vettore è data dalla combinazione di due valori). Questa combinazione dei due valori migliori risolve il problema di rimanere intrappolati in un

ottimo locale che presentano gli algoritmi visti nella sottosezione 1.5.1. Con il proseguire dell'algoritmo sarà possibile osservare che le singole particelle mano a mano convergono verso un ottimo locale, o di più, se la funzione da ottimizzare è multimodale (come in Figura 1.12). È possibile inoltre introdurre la *velocità* come parte dell'algoritmo, al fine di influire su quanto velocemente una particella si muove verso la soluzione, che in algebra vettoriale si traduce nella modifica del modulo del vettore di spostamento.

**Algoritmo 3:** Particle Swarm Optimization

```
1  n = numero particelle
2  d = numero dimensioni dello spazio di ricerca
3  for i ← 1 to n do
4      x_i ~ U(b_low, b_up) ▷ Inizializzo ogni particella con un valore random
         nel mio spazio di ricerca, delimitato da un lower bound b_low ed
         un b_up
5      p_i ← x_i ▷ Inizializzo la posizione migliore della particella alla
         sua posizione iniziale
6      if f(p_i) > f(g) then
7          g ← p_i ▷ Aggiorno la posizione migliore globale
8      v_i ~ U(−|b_up − b_low|, |b_up − b_low|) ▷ Inizializzo la velocità iniziale
         della particella
9  while criterio di terminazione non soddisfatto do
10     for i ← 1 to n do
11         for d ← 1 to m do
12             r_p, r_g ~ U(0, 1) ▷ Parametri di casualità
13             v_{i,d} ← ωv_{i,d} + φ_p r_p(p_{i,d} − x_{i,d}) + φ_g r_g(g_d − x_{i,d}) ▷ Aggiorno la
                 velocità della particella
14             x_i ← x_i + v_i ▷ Aggiorno la posizione della particella
15             if f(x_i) > f(p_i) then
16                 p_i ← x_i ▷ Aggiorno la posizione migliore della particella
17                 if f(p_i) > f(g) then
18                     g ← p_i ▷ Aggiorno la posizione migliore dello swarm
```

# Capitolo 2

# Stato dell'arte

## 2.1 Introduzione

Con l'avvento delle tecnologie per il sequenziamento del DNA partendo da singole cellule (SCS), iniziano ad essere disponibili dati di alta qualità. Queste tecnologie forniscono il sequenziamento di dati da singole cellule, permettendo quindi di ricostruire l'albero filogenetico di una cellula. È però da tenere in considerazione l'alto tasso di errore associato a questo tipo di dati, innalzando di conseguenza il grado di difficoltà del processo di ricostruzione della filogenesi. In questo capitolo, analizzeremo le tecnologie già presenti che hanno affrontato questa sfida.

## 2.2 SiFit: inferring tumor trees from single-cell sequencing data under finite-sites models, 2017 [9]

### 2.2.1 Problema e soluzione

asdf

# Bibliografia


[1] *Cancer Statistics*. URL: https://www.cancer.gov/about-cancer/understanding/statistics.

[2] Simone Ciccolella et al. «Inferring Cancer Progression from Single-cell Sequencing while Allowing Mutation Losses». In: *bioRxiv* (2018). DOI: 10.1101/268243.

[3] Cooper GM. *The Cell: A Molecular Approach. 2nd edition - DNA Replication*. URL: https://www.ncbi.nlm.nih.gov/books/NBK9940/.

[4] J. Kennedy e R. Eberhart. «Particle swarm optimization». In: vol. 4. Nov. 1995, 1942–1948 vol.4. DOI: 10.1109/ICNN.1995.488968.

[5] Max Roser e Hannah Ritchie. *OurWorldInData - Cancer*. Lug. 2015. URL: https://ourworldindata.org/cancer.

[6] *What is a cell? - Genetics Home Reference - NIH*. URL: https://ghr.nlm.nih.gov/primer/basics/cell.

[7] *What is DNA? - Genetics Home Reference - NIH*. URL: https://ghr.nlm.nih.gov/primer/basics/dna.

[8] *What is the Difference Between Cancer and a Tumor? | Dana-Farber Cancer Institute*. URL: https://blog.dana-farber.org/insight/2018/05/difference-cancer-tumor/.

[9] Hamim Zafar et al. «SiFit: inferring tumor trees from single-cell sequencing data under finite-sites models». In: *Genome biology* 18.1 (2017), p. 178. DOI: 10.1186/s13059-017-1311-2.