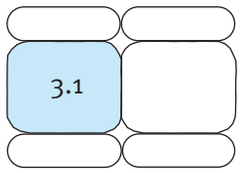
IL COMPUTER VIVENTE
CALCOLO MOLECOLARE E CELLULARE

Giancarlo Mauri

L'hardware dei calcolatori attuali è basato su circuiti elettronici le cui dimensioni si sono rapidamente ridotte, seguendo la "legge di Moore", ma questo trend si scontra ormai con limiti fisici che sembra molto difficile superare, mantenendo lo stesso paradigma di computazione. Per questo si stanno studiando e sperimentando modelli alternativi, in cui il supporto fisico per il calcolo non è più il silicio, ma sono biomolecole, cellule, batteri, con una ricerca interdisciplinare che coinvolge informatici, matematici, biologi, e anche medici, per le possibili applicazioni alla nanomedicina.



"*Biology and computer science — life and computation — are related. I am confident that at their interface great discoveries await those who seek them.*"

Leonard Adleman

1. TRA INFORMATICA E BIOLOGIA

Se il '900 è stato il secolo della fisica e dell'informatica, il 2000 promette di essere il secolo dell'informatica e della biologia (o più in generale delle scienze della vita). È infatti in atto una convergenza tra queste due discipline che vede da un lato l'informatica offrire alla biologia strumenti per gestire e analizzare la crescente massa di dati prodotti grazie a tecnologie come i sequenziatori di DNA o i *microarrays*, che permettono di verificare e confrontare il livello di espressione di migliaia di geni, dall'altro la biologia offrire all'informatica metafore e strumenti per nuovi paradigmi di calcolo, come le reti neurali, gli algoritmi genetici, ormai ben noti e sviluppati anche in senso applicativo, o il calcolo molecolare basato sul DNA, sulle cellule, sui batteri. Questo articolo si propone di introdurre le principali linee di ricerca e le prospettive del calcolo molecolare, a partire dalle prime idee di Feynman [10] e dalla loro prima concretizzazione effettiva, con l'esperimento di Adleman, ampiamente descritto nel terzo paragrafo. Una discussione sulle difficoltà presentate dall'approccio di Adleman è poi il punto di partenza per presentare gli sviluppi più recenti, che puntano a progettare automi a stati finiti costituiti da una singola molecola in grado ad esempio di rilevare lo stato di salute delle cellule in cui vengono collocati, o ad usare cellule (viventi o artificiali) o colonie di batteri per elaborare informazione in modo controllato.

2. VERSO IL CALCOLO MOLECOLARE

Alla fine degli anni '50 del secolo scorso, quando i computer occupavano ancora intere stanze, il fisico Richard Feynman già intui-

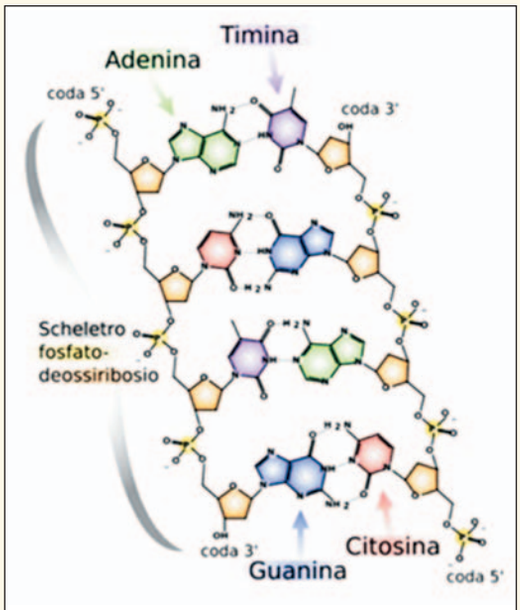
va le potenzialità dei sistemi biologici come manipolatori d'informazione su scala molecolare, e prefigurava la costruzione di computer sub-microscopici, basati su cellule viventi e complessi molecolari:
"Le cellule sono straordinariamente piccole, ma sono molto attive: trasformano varie sostanze, si spostano, si agitano e fanno altre cose meravigliose, tutte su una scala molto piccola. Ma anche, memorizzano e usano informazione. Si consideri la possibilità che anche noi riusciamo a realizzare un oggetto molto piccolo che faccia ciò che vogliamo – che possiamo costruire un dispositivo che operi a quel livello!" [10].
L'idea di un computer molecolare viene ripresa da Vaintsvaig e Liberman nel 1973 [26], e ulteriormente sviluppata da Conrad, che svolge ricerche approfondite sulla possibilità di elaborare informazioni con macromolecole, come le proteine [7].
Un significativo passo avanti, anche se solo teorico, è dovuto a Tom Head, che nel1987 definisce i **sistemi di accoppiamento** (***splicing systems***), un modello matematico che formalizza l'azione degli enzimi di restrizione sul DNA (riquadro 1) in termini di operazioni su linguaggi formali [14].
Un enzima di restrizione è una proteina che taglia una doppia elica di DNA in corrispondenza di particolari sottosequenze, specifiche per ogni enzima.
Per esempio, i due enzimi *TaqI* e *SciNI* tagliano il DNA in corrispondenza, rispettivamente, delle sottosequenze:

```
5'TCGA3'   e   5'GCGC3'
3'AGCT5'       3'CGCG5'
```

Riquadro 1 - La struttura del DNA

Il DNA è il depositario dell'informazione genetica nelle cellule viventi, ed è costituito da due filamenti che corrono parallelamente, formando una struttura spaziale a doppia elica. Ciascun filamento è una sequenza di **nucleotidi** di quattro diversi tipi, ciascuno dei quali è formato da uno zucchero (desossiribosio) cui sono legati un gruppo fosfato e una base azotata. Ogni nucleotide ha due estremità libere, indicate rispettivamente con 3' e 5' (corrispondono al terzo e al quinto dei cinque atomi di carbonio del desossiribosio) che gli consentono di legarsi ad altri nucleotidi (3' si lega con 5'), formando in tal modo la struttura portante centrale (*backbone*) del filamento. I nucleotidi differiscono l'uno dall'altro solo per la base azotata, che può essere **adenina**, **guanina**, **citosina o timina** (indicate rispettivamente con le lettere A, G, C e T). Una sequenza singola di DNA può quindi essere vista come una parola su un alfabeto di quattro lettere {*A,C,G,T*}, orientata per convenzione da 5' a 3'.
Una proprietà fondamentale dei nucleotidi è la **complementarità di Watson-Crick**: A è complementare a T, G è complementare a C, e due catene singole di DNA con orientamento opposto si legano tra loro grazie a legami a idrogeno tra basi complementari. In questo modo, formano una doppia elica stabile, con le catene di desossiribosio all'esterno e le coppie di basi legate all'interno (Figura **A**). Per esempio, la sequenza 5' –GGGGGAA–3' si legherà con 5'–TTCCCCC–3' , formando la doppia elica

```
5' — GGGGGAA — 3'
3' — CCCCCTT — 5'
```



A - *Struttura della doppia elica del DNA*

Sulle sequenze di DNA si possono eseguire diverse operazioni di modifica, attraverso l'azione di specifici enzimi. Le operazioni più interessanti per le applicazioni al calcolo sono:
Sintesi di una specifica sequenza.
Annealing (accoppiamento): due sequenze singole complementari si legano, formando una doppia elica.
Melting: una doppia elica si scinde nelle due sequenze componenti (per esempio, per riscaldamento).
Amplificazione con *PCR*: permette di produrre più copie di una sequenza data.
Separazione di sequenze sulla base della loro lunghezza.
Estrazione delle sequenze che contengono (o non contengono) una data sottosequenza.
Taglio: grazie agli enzimi di restrizione, è possibile tagliare sequenze doppie di DNA in corrispondenza di particolari siti, detti siti di restrizione; ogni enzima riconosce univocamente un sito specifico, in cui opera il taglio.
L'RNA è un'altra molecola fondamentale per i processi cellulari, e in particolare ha la funzione di rendere disponibile l'informazione codificata nel DNA per sintetizzare proteine. Rispetto al DNA, si presenta in catene singole il cui scheletro è formato da molecole di ribosio anziché di desossiribosio, cui si legano le basi A, C, G, mentre la timina (T) è sostituita dall'uracile (U).

Se consideriamo le due sequenze doppie:

```
5'CCCCCTCGACCCCC3'
3'GGGGGAGCTGGGGG5'
e
5'AAAAAGCGCAAAAA3'
3'TTTTTCGCGTTTTT5'
```

queste verranno tagliate dagli enzimi, producendo quattro nuove sequenze, con gli estremi non allineati:

```
5'CCCCCT CGACCCCC3' 5'AAAAAG  CGCAAAAA3'
3'GGGGGAGC TGGGGG5' 3'TTTTTCGC  GTTTTT5'
```

Si noti che le due basi libere della prima sequenza (G e C) sono complementari alle basi libere dell'ultima (C e G), e lo stesso accade per la seconda e la terza sequenza. La aggiunta di ligasi, l'enzima che induce la formazione di legami tra basi complementari, a questo punto può dare come risultato la formazione di due sequenze con le parti scambiate, rispetto a quelle di partenza:

```
5' CCCCCTCGCAAAAA3'5'AAAAAGCGACCCCC3'
3' GGGGGAGCGTTTTT5'3'TTTTTCGCTGGGGG5'
```

Questo meccanismo di ricombinazione del DNA è alla base delle biotecnologie, perché consente di modificare il materiale genetico in modo controllato.
Dal punto di vista dell'informatica possiamo leggerlo in un altro modo: ogni sequenza di DNA rappresenta dati codificati con un alfabeto di quattro lettere, e l'azione di un enzima di restrizione corrisponde ad una operazione effettuata su questi dati. Partendo da queste basi, Head ha sviluppato un modello formale di computazione che, a grandi linee, funziona come segue:
1. si assegnano un insieme iniziale D_0 di parole sull'alfabeto di quattro lettere $\Sigma = \{A, T, G, C\}$, che rappresentano i dati di ingresso, e un insieme di operazioni di ricombinazione della forma a#b$c#d, dove anche a, b, c, d sono parole su Σ;
2. si applicano le regole in questo modo: se D_0 contiene le parole w_1abw_2 e v_1cdv_2, l'operazione a#b$c#d le taglia e le ricombina, producendo le parole w_1adv_2 e v_1cbw_2;
3. si ottiene così un nuovo insieme D_1 su cui si ripete il procedimento, e così via, fino ad ottenere un insieme D_k che non può più essere trasformato.
Scegliendo opportunamente i dati iniziali e le regole di trasformazione, è possibile definire sistemi di splicing equivalenti come potenza computazionale a macchine che stanno ai diversi livelli di potenza nella gerarchia dei modelli di computazione tradizionali, dagli automi a stati finiti alle macchine di Turing [14].
Fin qui abbiamo idee interessanti, e risultati teorici promettenti per quanto riguarda la possibilità di usare sequenze di DNA per codificare informazioni ed enzimi per simulare computazioni, ma manca ancora una prova di realizzabilità pratica, che arriverà solo con l'esperimento di Leonard Adleman nel 1994 [1].

3. L'ESPERIMENTO DI ADLEMAN

È noto che i problemi di elaborazione dell'informazione possono essere classificati in base alla loro "complessità computazionale", cioè alla quantità di risorse (tipicamente tempo di calcolo) necessarie per risolverli, in funzione della dimensione dei dati di input. I problemi **NP-completi** (riquadro 2) sono tra i

RIQUADRO 2 - Problemi P, NP e NP-completi

Un problema di elaborazione dati è detto di tipo **P** (*Polinomiale*) se può essere risolto con un algoritmo il cui "tempo di esecuzione" cresce al più come un polinomio di grado k fissato rispetto al numero n di bit necessari per codificare l'input. I problemi in **P** sono considerati "*feasible*", o effettivamente risolubili, nel senso che anche per dati di dimensioni notevoli il risultato può essere ottenuto in tempi ragionevoli, mentre problemi non in **P**, pur teoricamente risolubili in modo algoritmico, non lo sono in pratica, dato che per input di dimensioni realistiche trovare una soluzione esatta richiederebbe tempi di calcolo che raggiungono rapidamente i miliardi di anni (e poco importa la potenza della macchina utilizzata).
Tra i problemi che le nostre conoscenze attuali non ci consentono di collocare in **P** (algoritmi polinomiali per questi problemi non sono stati trovati finora, ma potrebbero essere trovati in futuro), hanno particolare importanza i problemi **NP**. Un problema appartiene alla classe **NP** (*Polinomiale Non-deterministico*) se esiste un algoritmo che permette in tempo polinomiale di verificare se una particolare soluzione, scelta (magari da un indovino) tra un numero di soluzioni possibili che cresce in modo esponenziale è effettivamente una soluzione.
La classe **NP** contiene molti problemi di grande interesse applicativo, come il "problema del commesso viaggiatore" o il "problema dello zaino", e rappresenta il limite della nostra capacità di risolvere in pratica problemi algoritmici. Ovviamente **P** è incluso in **NP**; resta da stabilire se vale anche il viceversa, ovvero se le due classi coincidono: questo è noto come problema **"P = NP ?"**, ed è il più importante problema aperto nell'informatica teorica.
Purtroppo oltre quarant'anni di ricerche hanno portato alla convinzione generalizzata, anche se non alla dimostrazione, che valga invece **P ≠ NP**, cioè che ci siano problemi in **NP** non risolubili in tempo polinomiale, come i problemi **NP-completi**, che sono i più difficili tra i problemi in **NP**.

più difficili e tra questi Adleman ha scelto il Problema del Cammino Hamiltoniano Diretto (DHPP), analogo al Problema del Commesso Viaggiatore (TSP):

> dato un grafo orientato G con due vertici specifici "Partenza" (P) e "Arrivo" (A), stabilire se in esso esiste un *cammino hamiltoniano*, cioè un cammino da P ad A che passa una ed una sola volta per ciascuno degli altri vertici.

Nel lavoro pubblicato su Science [1], Adleman mostra come sia possibile risolvere una piccola istanza di DHPP, rappresentata dal grafo nella figura 1, manipolando in laboratorio molecole di DNA.

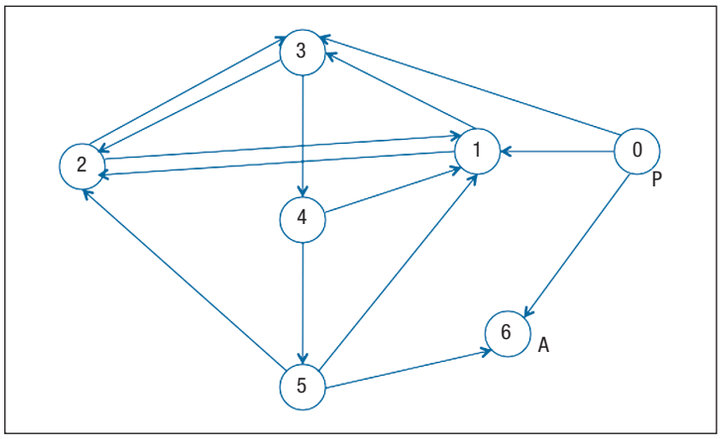


FIGURA 1
Il grafo dell'esperimento di Adleman

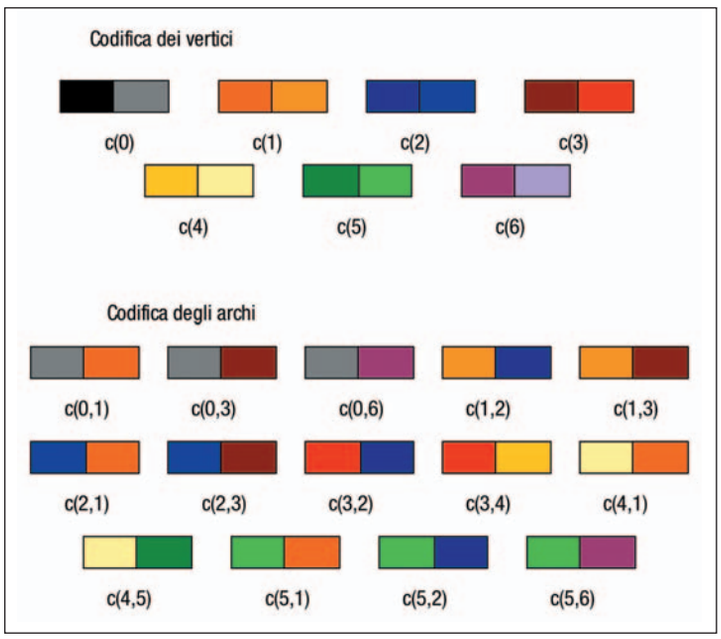


FIGURA 2
La codifica del grafo nell'esperimento di Adleman

L'idea di base è semplice: si tratta di codificare opportunamente i dati con biomolecole (DNA in questo esperimento, ma potrebbero essere anche RNA o proteine) anziché con sequenze di cifre binarie, e manipolare queste molecole usando tecniche comunemente disponibili nei laboratori di biologia molecolare, così da simulare operazioni che selezionano la soluzione del problema, se esiste.
L'algoritmo molecolare di Adlemann si sviluppa nei seguenti passi:

1. *Codifica del grafo*.
Un vertice n viene codificato con una sequenza c(n) di 20 nucleotidi, indicati con le lettere A, C, G, T, generata casualmente. Per esempio, possiamo porre:

```
c(1) = AATTGTGCGAAATTCCGGAA
c(2) = CCGTTAAGGCTATAAGGGTA
```

Un arco da n a m, che indicheremo con (n, m), viene codificato con una sequenza c(n, m) di 20 nucleotidi ottenuta concatenando la seconda metà di c(n) con la prima metà di c(m). Nel nostro esempio avremo:

```
c(1, 2) = AATTCCGGAACCGTTAAGGC
```

Nella figura 2 è illustrata graficamente la codifica del grafo di figura 1, usando colori diversi per rappresentare le codifiche dei vertici, suddivise in due metà, e le conseguenti codifiche degli archi.

2. *Generazione di percorsi sul grafo*.
50 pmol (circa 3.10^{13} molecole) di ciascuna delle sequenze codificanti gli archi vengono messe in soluzione con la stessa quantità di sequenze complementari alle codifiche dei vertici, in condizioni di reazione. In tal modo, ogni vertice "aggancia" due archi in parte complementari a ciascuna delle sue due metà, e così via, formando per ligasi molecole doppie che rappresentano cammini orientati nel grafo. Nella figura 3 sono rappresentati un cammino non hamiltoniano e uno hamiltoniano.

3. *Amplificazione dei cammini che iniziano con il nodo P e terminano nel nodo A*.
Usando la tecnica PCR (*Polimerase Chain Reaction* – vedi Nomenclatura), i cammini che iniziano con P e terminano con A vengo-

no enormemente amplificati, così da rendere trascurabile la percentuale degli altri cammini presenti in soluzione.

4. *Selezione dei cammini che passano esattamente per sette vertici.*

Questi cammini sono formati da 140 nucleotidi, e possono essere selezionati immergendo la soluzione ottenuta al passo precedente in un gel di agarosio sottoposto ad un campo elettrico. La diversa velocità di spostamento consente di isolare le molecole il cui peso molecolare corrisponde a 140 nucleotidi, eliminando le più corte e le più lunghe.

5. *Selezione dei percorsi che passano per n vertici differenti.*

Questo passo si basa sulla purificazione per affinità: con questa tecnica si comincia a selezionare le sequenze che contengono per esempio, il vertice 2, e si continua la selezione sui frammenti rimasti ripetendo il procedimento per tutti i vertici del grafo. In questo modo si è sicuri che il risultato finale sarà costituito dai soli frammenti che contengono una e una sola volta le sequenze codificanti ogni vertice.

6. *Lettura del risultato.*

Ancora con PCR, si amplificano le eventuali molecole rimaste, che codificano cammini hamiltoniani nel grafo.

4. I LIMITI DEL CALCOLO BASATO SUL DNA

La caratteristica principale del processo di calcolo sopra descritto è il suo altissimo parallelismo: la costruzione dei cammini nel grafo (passo 2), che è il passo di calcolo vero e proprio, è ottenuta attraverso un numero di reazioni chimiche dell'ordine del numero di molecole di DNA presenti, che avvengono in parallelo, come se fossero realizzate da altrettanti processori. La velocità effettiva è di circa 1.2×10^{18} operazioni al secondo, cioè oltre un milione di volte superiore a quella dei più potenti supercomputer. Altri notevoli vantaggi teorici rispetto ai dispositivi elettronici tradizionali sono l'efficienza energetica, dato che un Joule basta per $2 \cdot 10^{19}$ operazioni contro 10^9, e l'altissima densità di informazione, pari a circa 1 bit per nanometro cubo, mentre i dispositivi di memorizzazione attuali arrivano a 1 bit per 10^{10} nm^3. Ma queste sono, appunto, valutazioni teoriche, ricavate sulla base della soluzione di una istanza molto semplice di DHPP. In realtà, l'approccio di Adleman presenta molti limiti, sia teorici che pratici, e le difficoltà da superare prima di arrivare a sfruttare in pratica e su problemi significativi i principi dimostrati da Adleman sono ancora moltissime.

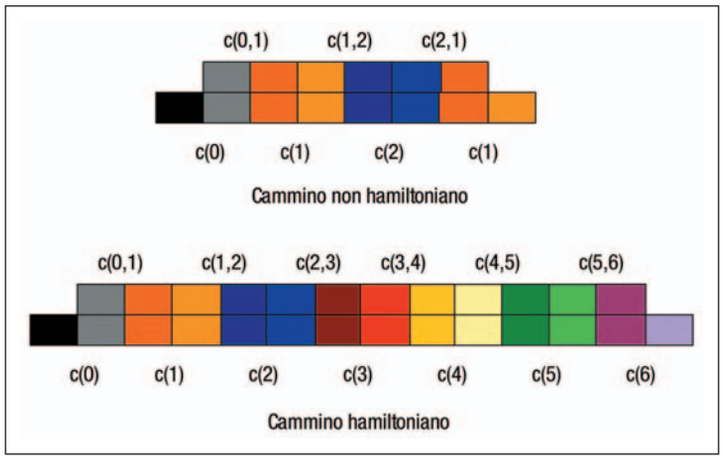


FIGURA 3
Esempi di cammini generati

I problemi teorici riguardano anzitutto la versatilità dei computer basati sul DNA e la loro capacità di trattare efficientemente un'ampia varietà di problemi computazionali. Infatti la procedura sopra descritta non corrisponde ad un modello di calcolo generale, ma è specifica per DHPP. Le domande che un informatico teorico si pone a questo punto sono due:

1. Il modello di calcolo basato sul DNA è computazionalmente completo nel senso che qualunque funzione computabile può essere calcolata attraverso manipolazioni del DNA?

2. Esiste un sistema di calcolo universale basato sul DNA, cioè un sistema che, data la codifica di una qualunque funzione computabile (in altre parole, un programma) sia in grado di calcolarla (cioè di eseguire il programma) per qualunque argomento?

Una prima generalizzazione è stata proposta da Lipton [17], che ha definito un modello formale di calcolo molecolare applicabile a tutti i problemi NP-completi. Il principale contributo di Lipton consiste in un metodo uniforme per codificare sequenze binarie arbitrarie con sequenze di DNA e in una sorta di linguaggio di programmazione per computer molecolari basato sulle operazioni ***merge*** (mettere nella stessa soluzione molecole che codificano dati diversi), ***separate*** (separare molecole che contengono o non contengono particolari sot-

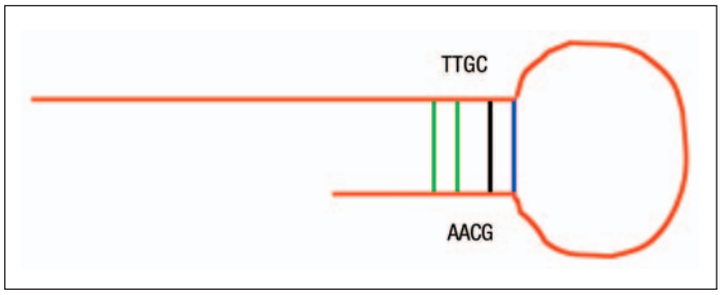


FIGURA 4

Auto-accoppiamento di una sequenza singola di DNA

tosequenze), ***detect*** (leggere l'output verificando quali molecole sono presenti nella soluzione finale). Sono poi stati proposti altri modelli teorici di calcolo basato sul DNA e ne sono state studiate le proprietà e la potenza computazionale, mostrando che diversi insiemi di bio-operazioni raggiungono la potenza delle macchine di Turing. Questi risultati (oltre ai risultati di Head [14] già citati) provano che computer molecolari "general purpose" sono in linea di principio realizzabili.

I problemi pratici riguardano soprattutto la scalabilità dell'approccio e la sua precisione. Infatti il procedimento funziona perché l'istanza di DHPP considerata è molto piccola. Per istanze di dimensioni significative per le applicazioni la situazione è ben diversa. Per esempio, si è stimato che per un grafo con 200 nodi, per rappresentare tutti i possibili cammini occorrerebbe un numero di molecole di DNA la cui massa supererebbe quella della terra: la crescita esponenziale non perdona!

Inoltre, le reazioni chimiche che realizzano la computazione, oltre ad essere sensibili all'ambiente (temperatura, acidità), presentano un certo margine di errore che, se pur ridotto, deve essere preso in considerazione. Visti i numeri in gioco ($2^{13} - 2^{14}$ molecole già per istanze molto semplici), una percentuale di un errore su un milione si traduce in un numero totale di errori tra i dieci e i cento milioni! I rischi di errore aumentano all'aumentare della lunghezza delle sequenze di codifica, perché sequenze molto lunghe potrebbero contenere sottosequenze tra loro complementari, che si legano tra loro rendendo così inutilizzabile la molecola ai fini del calcolo (è l'equivalente del *default* di un processore), come nell'esempio riportato nella figura 4.

Per questo motivo è importante realizzare librerie di sequenze di DNA progettate in modo da ridurre questo rischio, al cui interno scegliere le sequenze di codifica [9].

Infine, anche se il passo di calcolo vero e proprio, il passo 2, avviene in tempo brevissimo grazie al parallelismo, tutte le operazioni collaterali, per la codifica dei dati, il loro inserimento (che significa mettere molecole di DNA in soluzione in una provetta), la lettura dei risultati sono molto lente, con tempi dell'ordine della settimana, e non sembra si possano significativamente migliorare, anche se sono stati fatti tentativi per automatizzare le computazioni molecolari usando, per esempio, la microfluidica.

Sempre per superare queste difficoltà, si è sperimentato l'uso di molecole alternative al DNA, come l'RNA (acido ribonucleico) o le proteine, che hanno maggior flessibilità strutturale e funzionale, oppure si sono fatti esperimenti col DNA bloccato su una superficie anziché in soluzione, senza però arrivare a risultati decisivi.

Una descrizione dettagliata dell'esperimento di Adleman è stata pubblicata su Mondo Digitale nel 2006 [18]. Per una trattazione approfondita del calcolo basato sul DNA ci si può riferire al libro di Amos [2] o alla rassegna di Lila Kari [15].

5. OLTRE L'ESPERIMENTO DI ADLEMAN

Ovviamente la ricerca successiva ha portato a notevoli miglioramenti rispetto all'esperimento di Adleman, che hanno consentito di trattare con successo problemi più sofisticati, come una istanza del problema 3-SAT con 20 variabili, o la rottura del sistema crittografico DES. Tuttavia è opinione comune tra i ricercatori che non sono i problemi NP-completi il terreno su cui il calcolo basato sul DNA può competere con la tecnologia basata sul silicio: "È stupido tentare di predire il futuro della tecnologia, ma può essere che l'applicazione ideale del calcolo basato sul DNA non sia la soluzione di grandi problemi NP completi" [20].

Ciò non significa che sia tutto da buttare. Al contrario, pur con tutti i suoi limiti, l'esperimento di Adleman ha rappresentato un fortissimo stimolo per la ricerca sul calcolo moleco-

lare, ed ha aperto prospettive interessanti soprattutto in connessione con le nanoscienze, fornendo un punto di vista computazionale su molti loro aspetti, tra cui l'autoassemblaggio delle nanostrutture, e con la biologia e la medicina, con potenziali applicazioni al progetto di farmaci intelligenti. Gli orizzonti della ricerca sul calcolo molecolare si sono così enormemente allargati, dando origine ad una nuova disciplina a cavallo tra matematica, informatica, biologia e nanoscienze che punta a capire il ruolo dell'informazione e della sua manipolazione nei processi naturali: ruolo che, riprendendo una idea avanzata da Zuse e Fredkin nel 1960, potrebbe essere ancor più fondamentale di quello della materia e dell'energia. La tesi di Zuse e Fredkin sosteneva che l'intero universo è un qualche tipo di dispositivo computazionale, una sorta di gigantesco automa cellulare che continuamente aggiorna le sue regole; più recentemente, si è ipotizzato che l'universo sia un computer quantistico che calcola se stesso e il suo comportamento.
In questi scenari, l'attenzione non è tanto sul calcolo, quanto sulla capacità di controllo a livello molecolare. La **programmazione molecolare** si propone di definire una metodologia di programmazione sistematica per controllare le reazioni chimiche tra biomolecole, in modo da poter costruire nanostrutture biochimiche progettate a tavolino.

6. NANOMACCHINE BASATE SUL DNA

Il DNA, grazie alla specificità delle interazioni tra nucleotidi complementari, è il materiale ideale per queste costruzioni, e le interazioni tra sequenze corte di DNA possono essere controllate in modo affidabile attraverso la progettazione della sequenza di basi. Nell'ultimo decennio, gli esperimenti di costruzione di nanostrutture e nanomacchine basate sul DNA sono stati numerosissimi; una interessante rassegna è riportata in [5].
Uno dei più attivi ricercatori in questo settore è Ned Seeman, che nel suo laboratorio ha sintetizzato straordinarie strutture di DNA autoassemblate come cubi, ottaedri troncati, cristalli bi- e tri-dimensionali di DNA [25]. Altri ricercatori hanno realizzato triangoli di Sierpinski [23], o nanostrutture complesse in grado di eseguire calcoli come il conteggio di stringhe binarie o lo XOR bit a bit [19]. Un modo per sfruttare questro controllo architetturale straordinariamente preciso è l'uso di modelli di DNA autoassemblati per posizionare molecole funzionali. Si possono così realizzare circuiti elettronici molecolari, dispositivi ottici e reti enzimatiche.
Una immediata estensione di queste ricerche è il passaggio da strutture di DNA statiche a macchine, o "nanobots", in grado di muoversi. Il DNA è meno adatto per la costruzione di strutture attive rispetto alle proteine o all'RNA, che hanno una maggiore versatilità strutturale e catalitica. La maggior ricchezza di interazioni e legami che si possono instaurare in una molecola di RNA rispetto al DNA, in cui il legame nettamente prevalente è quello dovuto alla complementarità di Watson e Crick, permette la formazione di complesse strutture bi- e tri-dimensionali. Attualmente i biochimici non sono in grado di controllare queste interazioni fino ad ottenere strutture tridimensionali programmate. Si usa quindi il DNA perché la semplicità della sua struttura e delle interazioni ci permette di controllarne meglio l'assemblaggio.
Le nanostrutture attive più semplici sono interruttori o molecole che possono essere fatte passare dall'una all'altra di due diverse conformazioni. Le "pinzette" realizzate dal gruppo di Yurke sono un interessante esempio di nanostruttura di questo tipo. Sono costituite da due "braccia" di DNA in doppia sequenza, unite da una "cerniera" flessibile in sequenza singola. La pinzetta può assumere una configurazione chiusa ed una aperta, e il passaggio dall'una all'altra viene ottenuto attraverso due diverse molecole di DNA, corrispondenti rispettivamente ai comandi "apri" e "chiudi".
Altri esperimenti hanno riguardato la costruzione di circuiti logici basati sul DNA, o di ribozimi (sono molecole di RNA in grado di catalizzare reazioni che modificano altre molecole, o anche il ribozima stesso) che possono essere usati per eseguire operazioni logiche e semplici computazioni [22], oppure motori molecolari e "nanobots" che si muovono lungo percorsi autoassemblati, e si procurano l'energia necessaria catalizzando reazioni chimiche [5].
In queste strutture, il passaggio da una conformazione all'altra viene indotto da cambiamenti di temperatura o di acidità, o dal legame con

una molecola di segnalazione, in genere una sequenza singola di DNA. Un possibile utilizzo è quindi come sensori molecolari, in grado di misurare grado di acidità, temperatura, presenza o assenza di piccole molecole o di proteine direttamente all'interno della cellula, dato che il DNA è un supporto computazionale ovviamente biocompatibile. Questo apre, anche se non a breve termine, una nicchia applicativa legata alla medicina che non può essere certo coperta da chip al silicio, con la possibilità di eseguire diagnosi, di controllare dispositivi terapeutici, di trasportare farmaci, di attivare un processo che porti alla morte programmata della cellula nel caso questa risulti irrimediabilmente danneggiata.

Lavorando in questa direzione, il gruppo di Ehud Shapiro ha realizzato un automa a stati finiti costituito da biomolecole [5]. Grazie ad una serie di regole di transizione, realizzate attraverso enzimi che tagliano e ricombinano il DNA, la macchina cambia il proprio stato interno in funzione dello stato corrente, rappresentato da una molecola di DNA, e dell'input, rappresentato dalla presenza o meno di particolari molecole di mRNA, fino a raggiungere uno stato finale, in corrispondenza del quale rilascia un output, pure rappresentato da una doppia elica di DNA. L'mRNA (RNA messaggero) è la molecola di collegamento tra un gene, codificato nel DNA, e la proteina corrispondente: in particolari condizioni, il gene "si esprime", producendo l'mRNA, che a sua volta esce dal nucleo e viene "tradotto" in proteina dai ribosomi, sulla base di un codice preciso che fa corrispondere ad ogni sequenza di tre nucleotidi uno dei venti aminoacidi che costituiscono le proteine. Malattie come i tumori dipendono da alterazioni di questo meccanismo che portano a produrre proteine "sbagliate", o a non produrre proteine essenziali. Un automa come quello sopra descritto potrebbe rilevare e segnalare all'esterno queste alterazioni, e anche cercare di correggerle. L'obiettivo a lungo termine è quindi la realizzazione di macchine molecolari da usare come "sentinelle" all'interno dell'organismo, in grado di riconoscere cellule o tessuti danneggiati, e all'occorrenza di rilasciare molecole per ripararli. Si aprono così importanti prospettive in termini di terapie innovative e personalizzate e sintesi di farmaci "intelligenti".

7. CALCOLARE CON LE CELLULE

Quanto sono realistiche queste prospettive? Per rispondere a questa domanda dobbiamo tornare all'inizio, alla visione di Feynman: la cellula come sistema di elaborazione dell'informazione. Il punto è che gli esperimenti sopra ricordati sono stati effettuati *in vitro*, quindi in un ambiente assolutamente controllato e in presenza solo delle molecole coinvolte nell'esperimento. Una volta inserite in una cellula, le nanomacchine si troverebbero in un ambiente molto più complesso, in cui avvengono in parallelo migliaia di processi biochimici che interferiscono e si influenzano l'un l'altro all'interno di complesse reti regolatorie. Occorre allora capire come funzionano queste reti regolatorie e trovare strumenti per imbrigliarle e piegarle ai nostri scopi, sia di calcolo che di controllo: il **calcolo cellulare** [12] ha l'obiettivo di "imbrigliare" i processi che avvengono nelle e tra le cellule viventi. Qui l'informatica si interseca da un lato con la **biologia sistemica** [11], che cerca di comprendere gli organismi biologici come reti di interazioni, dall'altro con la **biologia sintetica** [3], che ha come obiettivo l'ingegnerizzazione e la costruzione di sistemi biologici artificiali.

Una prima linea di ricerca che emerge da queste considerazioni riguarda lo studio dei meccanismi cellulari per astrarne e formalizzarne gli aspetti significativi dal punto di vista algoritmico e di controllo dei processi concorrenti: il numero di processi concorrenti che una cellula è in grado di gestire al suo interno è di gran lunga superiore a quelli gestiti da un qualunque sistema operativo!

I **sistemi a membrane** [21], proposti da George Paun nel 1998, sono modelli di calcolo parallelo e distribuito definiti a partire dalla struttura e dal funzionamento delle cellule viventi, e dal modo in cui queste si organizzano in tessuti o in strutture di ordine superiore. Una cellula è costituita anzitutto da una membrana, che la separa dall'ambiente esterno, che contiene molecole di vario genere ed altre membrane, e così via. Questa struttura gerarchica a compartimenti è la principale caratteristica delle cellule che viene astratta nei sistemi a membrane. Per quanto riguarda gli aspetti dinamici, di evoluzione delle cellule, all'interno di ogni membrana avvengono reazioni chimiche che ne

modificano il contenuto; inoltre alcune molecole possono attraversare le membrane, muovendosi verso l'esterno o verso l'interno, realizzando in tal modo una forma di comunicazione dell'informazione, e si possono formare nuove membrane, mentre membrane esistenti possono dissolversi o duplicarsi per gemmazione.
Astraendo questa struttura, definiamo un sistema a membrane generico come una struttura gerarchica di **regioni** o compartimenti, delimitati da membrane individuate univocamente da una etichetta, e contenuti in una membrana esterna, chiamata **pelle**, che separa il sistema dall'ambiente esterno. Ogni membrana contiene un multiinsieme di **oggetti** che evolvono in base a **regole di evoluzione** applicate non deterministicamente e con massimo parallelismo. Gli oggetti possono essere anche spostati da una regione alla regione immediatamente più esterna (out) o ad una delle regioni interne (in). Le regole hanno quindi la forma:

$c \rightarrow (d, here)$

che ci dice che l'oggetto c viene trasformato nell'oggetto d che rimane nella stessa membrana, o, con significato facilmente ricavabile,

$ab \rightarrow (cd, here)$
$c \rightarrow (d, out)$
$c \rightarrow (d, in)$

La figura 5 visualizza un esempio di sistema a membrane.
L'applicazione delle regole di evoluzione determina la transizione del sistema da uno stato all'altro. Una **computazione** parte da una configurazione di input assegnata e procede eseguendo ad ogni passo tutte le regole che è possibile eseguire (massimo parallelismo) in ogni membrana. Il risultato della computazione, quando questa si arresta perché non è più possibile applicare nessuna regola, può essere letto in una **membrana di output** assegnata.
I sistemi a membrane sono stati ampiamente studiati negli ultimi anni, e ne sono state definite diverse varianti; ad esempio, sono stati proposti diversi meccanismi formali che riflettono il modo selettivo in cui le membrane biologiche permettono alle molecole di attraversarle. Per molte di queste varianti si è dimostrata l'equivalenza computazionale con le macchine di Turing. Inoltre, come nell'esperimento di Adleman, si può sfruttare il parallelismo per accelerare le computazioni. In particolare, se si aggiungono regole che permettono di dividere o duplicare le membrane, è facile dimostrare che si possono risolvere problemi NP-completi in tempo lineare, naturalmente al prezzo di una crescita esponenziale del numero di membrane.
La situazione non è molto diversa da quella del calcolo basato sul DNA prima dell'esperimento di Adleman: abbiamo le idee e i risultati teorici, ma manca la prova della realizzabilità pra-

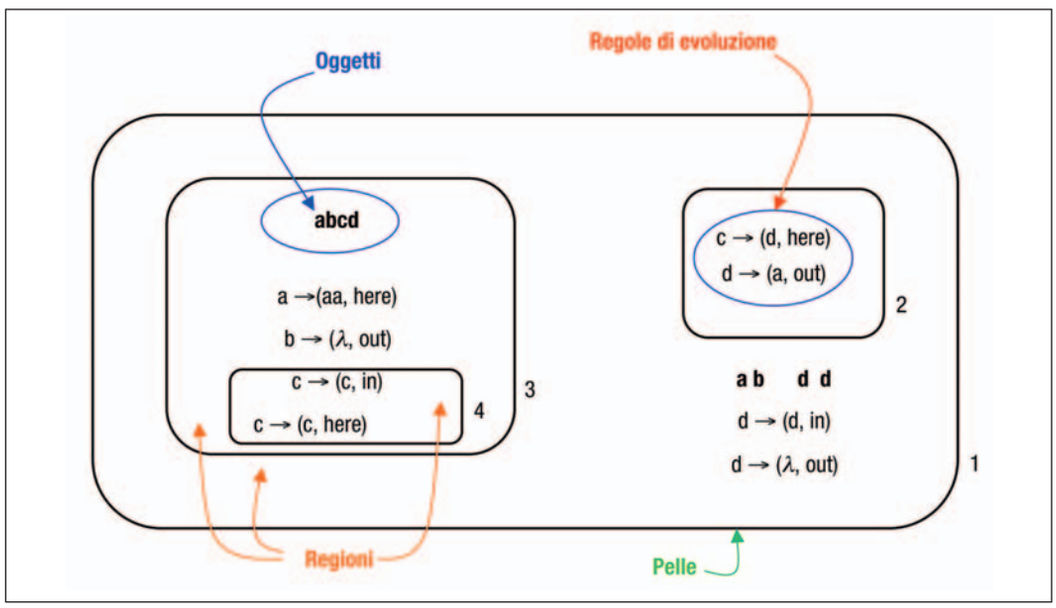


FIGURA 5
Un esempio di sistema a membrane

tica di queste idee. Per questo, si stanno seguendo diverse strade per realizzare computazioni "in vivo", progettando cellule sintetiche o cercando di utilizzare colonie di batteri come substrati computazionali, estendendo il loro comportamento attraverso l'inserimento di circuiti logici biochimici che controllino i processi intra- ed inter-cellulari.

8. CALCOLARE CON I BATTERI

I batteri sono organismi unicellulari piuttosto semplici, e costituiscono un buon punto di partenza per studiare il funzionamento della cellula dal punto di vista dell'elaborazione dell'informazione.
Lo studio dei Ciliati, un gruppo di circa 10.000 diverse specie di batteri che sono tra i più antichi organismi noti, ha portato a scoprire che il processo di decodifica dei geni che essi adottano presenta molte analogie con l'algoritmo di Adleman, e in alcuni passaggi usa meccanismi molto simili ai puntatori nei linguaggi di programmazione.
I ciliati hanno due tipi di nucleo: un macronucleo attivo (soma), che governa lo sviluppo cellulare, e un micronucleo funzionalmente inerte, che contribuisce solo alla riproduzione sessuale. Il micronucleo contiene una versione "criptata" del DNA del macronucleo, ottenuta "ingarbugliando" frammenti dei geni funzionali del macronucleo con operazioni di permutazione, inversione e inclusione di sequenze non codificanti. Nel processo di riproduzione sessuale i ciliati si scambiano informazione genetica e formano nuovi micronuclei, che usano in tempo reale per assemblare i macronuclei che servono per la loro sopravvivenza. Questo processo di riorganizzazione e assemblaggio richiede di riconoscere ed eliminare i frammenti non codificanti di DNA, e quindi di porre nell'ordine corretto i frammenti codificanti, ed è affascinante sia dal punto di vista biologico che da quello computazionale. Dal punto di vista computazionale, la struttura dati implicita nel DNA del micronucleo può essere definita come una lista a puntatori, e la ricostruzione del DNA del macronucleo corrisponde a tutti gli effetti a risolvere un problema NP-completo. Dal punto di vista della biologia, gli sforzi comuni di informatici e biologi hanno portato ad una ipotesi plausibile (già parzialmente sostenuta da alcuni dati sperimentali) sul "bioware" che implementa il processo dell'assemblaggio dei geni nei ciliati, basato sul nuovo concetto di ricombinazione guidata da un modello [22].
Un secondo approccio per realizzare dispositivi di calcolo sfruttando i batteri è suggerito dalla biologia sistemica, che vede il genoma di un organismo come una rete di componenti computazionali interagenti, simili a porte logiche. Questo punto di vista, oltre a fornire ai biologi un'analogia utile a comprendere meglio i meccanismi della vita, ha ispirato ricerche nella direzione della reingegnerizzazione della circuiteria genetica e metabolica allo scopo di eseguire calcoli secondo programmi definiti dall'uomo.
Consideriamo ad esempio il cosiddetto "operone Lac", costituito da tre geni con un promotore comune che nell'Escherichia coli producono un gruppo di proteine che consentono di metabolizzare lo zucchero lattosio. Per ragioni di efficienza, queste proteine devono essere prodotte solo se nell'ambiente è presente lattosio, altrimenti si avrebbe una inutile dissipazione delle risorse della cellula. Inoltre, se è presente anche glucosio, è preferibile consumarlo prima del lattosio, perché è più facile da metabolizzare. La cellula contiene quindi una rete regolatoria che consente di attivare l'operone Lac solo se c'è lattosio (L) e non c'è glucosio (G): di fatto, abbiamo una porta logica che calcola la funzione booleana (L AND (NOT G)), poiché dà in output 1 (Lac attivo) se e solo se L = 1 (lattosio presente) e G = 0 (glucosio assente).
La comprensione di questo meccanismo apre la via all'implementazione di circuiti logici *in vivo*. Anzitutto, si seleziona un insieme di geni che rappresentano le porte logiche, assicurando che le dipendenze tra geni riflettano in modo corretto la connettività del circuito. Questi geni sono poi inseriti - ad esempio - in un batterio, usando gli strumenti standard della biologia molecolare. Questa fase di inserimento è la più costosa, in termini di tempo, dell'intero processo sperimentale, in quanto l'inserimento anche di un solo gene può essere problematico. Tuttavia, una volta che l'intero insieme di geni è presente in una colonia di batteri abbiamo a disposizione una quantità illimitata di "hardware biologico". Gli ingressi del circuito

sono rappresentati dalla presenza o assenza nell'ambiente dei vari composti da cui dipende l'espressione dei geni che rappresentano le porte logiche del primo livello. In seguito, lo sviluppo della cellula e i complessi processi di regolazione sottostanti simulano il circuito, senza bisogno di ulteriori interventi umani.

Nell'ultimo decennio sono stati riportati in letteratura diversi esperimenti di laboratorio con l'obiettivo di realizzare *in vivo* porte logiche AND e OR o circuiti più complessi. Per esempio, in [11] Gardner, Cantor e Collins descrivono la realizzazione di un interruttore genetico costituito da una rete regolatoria di geni bistabile sintetica inserita in Escherichia Coli. La transizione da uno degli stati stabili all'altro è indotta chimicamente o termicamente, con una soglia di transizione pressoché ideale.

Questo ha applicazioni ovvie alla realizzazione di dispositivi di memoria molecolari. Sempre in E. Coli, Elowitz e Leibler hanno realizzato una rete sintetica di regolatori genici che implementa un oscillatore. La rete avvia periodicamente la sintesi di una proteina con fluorescenza verde che permette di leggere lo stato della rete in ogni singola cellula. Le oscillazioni risultanti, con periodi tipici dell'ordine delle ore, sono più lente del ciclo di divisione cellulare, per cui lo stato dell'oscillatore viene trasmesso di generazione in generazione.

Un approccio diverso, seguito per esempio da Y. Sakakibara [24], punta a sviluppare un automa a stati finiti programmabile e autonomo, nel senso che, una volta inserito in E. coli, funziona sfruttando il bio-hardware del batterio, senza richiedere ulteriori interventi umani. Sia l'input che la funzione di transizione di stato dell'automa sono codificati con plasmidi, molecole di DNA circolari molto stabili e maneggevoli.

Gli esperimenti sopra citati si basano su processi che avvengono all'interno di una singola cellula, ma anche i processi intercellulari, in particolare i meccanismi di comunicazione tra cellule batteriche, possono essere utilizzati come substrato di calcolo [27].

Quando milioni di batteri agiscono collettivamente come gruppo (come nel caso di colonie batteriche), come risultato del modo in cui le singole cellule rispondono a quelle che si trovano nelle vicinanze e alle condizioni ambientali (in genere differenti per ogni batterio individuale, anche se sono relativamente vicini l'uno all'altro) si produce una grande varietà di schemi diversi nel modo in cui le cellule si raggruppano nelle diverse posizioni. Questi schemi sono il risultato emergente delle interazioni locali e delle condizioni ambientali, e possono presentare aspetti funzionali; per esempio, in condizioni di stress (come la presenza nell'ambiente di sostanze tossiche), lo schema prodotto sarà finalizzato a proteggere il maggior numero possibile di individui, minimizzando il numero di cellule senza vicini, e a promuovere la ricerca da parte della colonia di un ambiente meno dannoso, con l'estensione di "braccia" per esplorare l'ambiente in varie direzioni. Questo sviluppo spontaneo di schemi funzionali, come puro risultato della combinazione di condizioni ambientali e di interazioni locali tra individui, è una forma di computazione molto comune e utile in natura, ma attualmente poco studiata e poco sfruttata in termini computazionali.

Può essere possibile imbrigliare le sofisticate capacità di generazione di schemi guidati dall'ambiente dei batteri allo scopo di fornire applicazioni concrete (per esempio, alla "vernice intelligente" di un aereo o ai nodi di una rete di sensori) in cui l'input ambientale determina l'emergere di una organizzazione spazio-temporale adattiva e funzionale (negli esempi citati, potrebe essere, rispettivamente, uno schema superficiale che stabilizzi rispetto a turbolenze, o uno schema di connessione che ottimizzi l'assorbimento di potenza).

Per chiudere, torniamo al punto da cui siamo partiti, la soluzione di DHPP con DNA. In un recentissimo lavoro [4], viene descritta la programmazione di una colonia batterica con un circuito genetico che consente ai batteri di valutare tutti i cammini posssibili in un grafo diretto per verificare l'esistenza di un cammino Hamiltoniano. Il grafo considerato ha tre vertici, e viene codificato con frammenti di DNA che vengono inseriti in *Escherichia coli*. I vertici del grafo sono rappresentati da porzioni di due distinti geni che codificano due proteine fluorescenti, rispettivamente verde e rossa. I batteri che individuano un cammino Hamiltoniano "comunicano" il successo producendo colonie di colore giallo.

9. CONCLUSIONI

L'iniziale aspettativa, alimentata dal successo dell'esperimento di Adleman, di poter risolvere problemi combinatori complessi (NP completi) sfruttando la densità di codifica dell'informazione e il parallelismo massiccio con cui si può operare sulle molecole di DNA si è dimostrata poco realistica, per i motivi discussi nel paragrafo 4. Tuttavia le ricerche che da quell'esperimento e da quell' aspettativa sono scaturite hanno aperto scenari e prospettive ancor più interessanti, sia teorici che applicativi, a partire dalla messa in evidenza, di grande importanza anche sul piano filosofico/epistemologico, del fatto che l'informazione gioca nei processi biologici fondamentali un ruolo pari a quello di materia ed energia.
Capire la logica che sottostà all'elaborazione dell'informazione nelle cellule e nei tessuti (questo è anche il titolo di una importante serie di conferenze scientifiche su questi argomenti) diventa quindi un obiettivo scientifico e di avanzamento della conoscenza fondamentale, che richiede la collaborazione interdisciplinare di biotecnologie, biologia, informatica, nanotecnologie, medicina, e intorno al quale ogni disciplina può articolare linee di ricerca sia teoriche che applicative di proprio interesse.
Forse, o almeno non a breve termine, i computer molecolari non sostituiranno i computer elettronici, probabilmente si arriverà ad una coesistenza, su nicchie applicative diverse. In ogni caso, anche se che non si realizzassero applicazioni significative, la ricerca in questo settore ha già rivoluzionato i rapporti tra biologia, matematica, scienze dell'informazione e ingegneria, e ha costretto a ripensare il significato di "calcolare".

Biologia molecolare – Nomenclatura essenziale

DNA (Acido Desossiribonucleico) – Molecola che codifica l'informazione genetica.

RNA (Acido Ribonucleico) – Molecola che rende disponibili le informazioni codificate nel DNA per la produzione di proteine. Si presenta in diverse forme, tra cui:

mRNA (RNA messaggero) – copia l'informazione genetica da un tratto di DNA, attraverso il processo di *trascrizione* e la trasporta sui *ribosomi*.

tRNA (RNA di trasporto) – ha la funzione di interprete per la *traduzione* dell'mRNA nella proteina corrispondente essendo in grado di associare ad ogni codone (tripletta di basi consecutive) l'aminoacido corrispondente.

Traduzione – Processo attraverso il quale viene sintetizzata una proteina, utilizzando l'informazione genetica contenuta in un filamento di mRNA. I nucleotidi che formano l'mRNA vengono letti a gruppi di tre (*codoni*), a ciascuno dei quail corrisponde un preciso aminoacido da inserire nella proteina (Gli aminoacidi distinti sono venti).

Ribosomi (al singolare **ribosoma**) – Organelli contenuti nella cellula, la cui funzione è quella di sintetizzare le proteine leggendo le informazioni contenute in un filamento di mRNA e traducendole con l'aiuto del tRNA.

Ribozima (termine composto da acido **ribo**nucleico ed en**zima**), anche noto come **enzima a RNA** o **RNA catalitico** – è una molecola di RNA in grado di catalizzare una reazione chimica.

Reazione a catena della polimerasi (*Polymerase Chain Reaction, PCR*) – Tecnica di biologia molecolare che consente la moltiplicazione (*amplificazione*) di frammenti di DNA dei quali si conoscano le sequenze nucleotidiche iniziali e terminali. L'amplificazione mediante PCR consente di ottenere in vitro molto rapidamente la quantità di materiale genetico necessaria per le successive applicazioni.

Bibliografia

[1] Adleman L.: Molecular computation of solutions to combinatorial problems. *Science*, Vol. 266, 1994, p. 1021-1024.

[2] Amos M.: Theoretical and experimental DNA computation. *Springer*, 2005.

[3] Andrianantoandro E., Basu S., Karig D., Weiss, R.: Synthetic biology: new engineering rules for an emerging discipline. *Molecular Systems Biology*, Vol. 2, 2006, p. 1–14.

[4] Bath J., Turberfield A.: DNA nanomachines. *Nature Nanotechnology*, Vol. 2, May 2007, p. 275-284.

[5] Baumgardner J., et al.: Solving a Hamiltonian Path Problem with a bacterial computer. *Journal of Biological Engineering*, Vol. 3, 2009.

[6] Benenson Y., Gil B., Be-Dor U., Adar R., Shapiro E.:

An autonomous molecular computer for logical control of gene expression. *Nature*, Vol. 429, 2004, p. 423–429.

[7] Conrad M.: On design principles for a molecular computer. *Comm. ACM*, Vol. 28, n. 5, 1985, p. 464-480.

[8] Ehrenfeucht A., Harju T., Petre I., Prescott D., Rozenberg G.: *Computation in Living Cells: Gene Assembly in Ciliates*. Springer, 2004.

[9] Ferretti C., Mauri G.: *Word design for DNA computing: a survey*. Proc. 9-th Int. Meeting on DNA Based Computers (J. Chen, J. Reif ed's.), Lect. Not. Comp. Sci. 2943, Springer-Verlag, 2003, p. 37-46.

[10] Feynman R.P.: *There's plenty of room at the bottom*. In D. Gilbert, editor, Miniaturization, Reinhold, 1961, p. 282-296.

[11] Gardner T., Cantor R., Collins J.: Construction of a genetic toggle switch in Escherichia Coli. *Nature*, Vol. 403, 2000, p. 339-342.

[12] Kitano H.: *Foundations of Systems Biology*. MIT Press, Cambridge/MA, 2001.

[13] Frisco P.: *Computing with Cells*. Oxford University Press, 2009.

[14] Head T.: Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors. *Bull. Math. Biology*, Vol. 49, 1987, p. 737-759.

[15] Kari L.: DNA computing—the arrival of biological mathematics. *The Math. Intelligencer*, Vol. 19, n. 2, 1997, p. 9–22.

[16] Landweber L., Kari L.: The evolution of cellular computing: Nature's solution to a computational problem. *Biosystems*, Vol. 52, n. 1/3, 1999, p. 3-13.

[17] Lipton R.J.: DNA solution of hard computational problems. *Science*, Vol. 268, 1995, p. 542-545.

[18] Manca V.: Frontiere della ricerca - DNA computing, il calcolatore in provetta. *Mondo Digitale* n. 4, 2006.

[19] Mao C., LaBean T.H., Reif J.T., Seeman N.C.: Logical computation using algorithmic self-assembly of DNA triple-crossover molecules. *Nature*, Vol. 407, 2000, p. 493-496.

[20] Ogihara M., Ray A.: DNA computing on a chip. *Nature*, Vol. 403, 2000, p. 143–144.

[21] Paun G.: *Membrane Computing: An Introduction*. Springer, 2002.

[22] Reif J., LaBean T.: Autonomous programmable biomolecular devices using self-assembled DNA nanostructures. *Commun. ACM*, Vol. 50, n. 9, 2007, p. 46-53.

[23] Rothemund P., Papadakis N., Winfree E.: Algorithmic self-assembly of DNA Sierpinski triangles. *PLoS Biology*, Vol. 2, n. 12, dec. 2004.

[24] Sakakibara Y., Nakagawa H., Sakamoto K.: *Development of an in vivo computer based on Escherichia coli*. Proceedings of 11-th International Meeting on DNA Based Computers, London, Ontario, 2005, p. 68-77.

[25] Seeman N.: Nanotechnology and the double helix. *Scientific American Reports*, Vol. 17, n. 3, 2007, p. 30-39.

[26] Vaintsvaig M.N., Liberman E.A.: Formal description of cell molecular computer. *Biofizika*, Vol. 18, 1973, p. 939-942.

[27] Weiss R.: Cellular *Computation and Communications using Engineered Genetic Regulatory Networks*. PhD thesis, Massachusetts Institute of Technology, 2001.

GIANCARLO MAURI è docente di informatica presso l'Università di Milano-Bicocca. I suoi interessi di ricerca riguardano la bioinformatica, la modellazione e simulazione di sistemi biologici, i modelli e sistemi di calcolo basati sul DNA (DNA computing) o ispirati alla struttura e al funzionamento delle cellule (sistemi a membrane e calcolo cellulare).
E-mail: mauri@disco.unimib.it