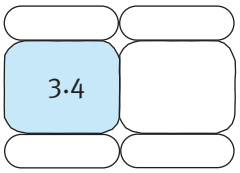
FRONTIERE DELLA RICERCA
DNA COMPUTING
IL CALCOLATORE IN PROVETTA

Vincenzo Manca

Il *DNA Computing* è un modello di calcolo proposto dodici anni fa da Leonard Adleman. In esso le molecole di DNA codificano sequenze di simboli e tipici processi biotecnologici realizzano algoritmi che elaborano tali sequenze. In questo articolo si presentano le idee fondamentali e i presupposti dei calcoli effettuati tramite il DNA. Quindi, si descrivono i principi dell'esperimento di Adleman e si indicano gli sviluppi, le tendenze e le attuali frontiere della ricerca nel settore.



1. INTRODUZIONE

L'idea iniziale del *DNA Computing* è molto semplice: "calcolare" significa passare da dati iniziali a risultati finali che soddisfino certe condizioni risolutive, ma dati e risultati sono sempre esprimibili con "parole" in un qualche linguaggio di rappresentazione dell'universo in cui si opera. Le molecole di DNA si possono assimilare a "parole doppie", costruite a partire da quattro simboli: A, T, C, G che stanno per le iniziali delle basi azotate: *Adenina, Timina, Citosina e Guanina*. Quindi, calcolare con il DNA significa sviluppare calcoli su "parole" di DNA, ovvero "stringhe" costruite sulle quattro lettere A, T, C, G. Questa metafora è stata tradotta in realtà da un esperimento ideato da Leonard Adleman [1] in cui, manipolando opportunamente una popolazione iniziale di molecole DNA che codificavano un grafo, si otteneva una popolazione finale composta da molecole di un solo tipo, che codificava la soluzione di un particolare problema definito sul grafo di partenza.
Su una molecola di DNA si può scrivere l'informazione desiderata, non appena si definisca un criterio di codifica. Per esempio, usiamo due lettere consecutive per indicare una cifra decimale secondo lo schema: 0 = AA, 1 = AT, 2 = TT, 3 =CC, 4 =CG, 5 = GG, 6 = AC, 7 = AG, 8 = TC, 9 = TG. In questo modo il numero 1270 viene codificato dalla parola ATTTAGAA. Per sommare tramite DNA i numeri 1270 e 27 basta quindi partire da una provetta in cui vi siano inizialmente molecole ATTTAGAA e TTAG, in una qualche concentrazione (diciamo una pico-mole di ciascuna in soluzione acquosa), e applicare dei procedimenti di manipolazione biomolecolare, di cui diremo più avanti, in modo che alla fine vi sia solo un tipo di molecola lunga otto bp (coppie di basi) del tipo ATTTTGAG, che corrisponde, secondo la nostra codifica, proprio alla somma 1297 dei due numeri 1270 e 27. Ovviamente tale metodo è di scarso interesse pratico, ma si presta bene a spiegare l'essenza di un qualsiasi algoritmo DNA: codificare i dati con molecole DNA poste in una provetta iniziale, applicare delle operazioni che trasformano ad ogni passo il contenuto della provetta e alla fine "leggere" il risultato in qualche tipo di molecola selezionata secondo un opportuno criterio di decodifica.

Un tale modello di calcolo è profondamente diverso dal classico modello della macchina di Turing. Infatti questa, secondo una delle sue formulazioni più comuni, elabora una struttura lineare, assimilabile ad un nastro suddiviso in caselle, in cui è posto un simbolo per casella. L'organo di controllo della macchina può alterare il contenuto di una casella e spostarsi a leggere nella casella contigua di destra o di sinistra. L'azione svolta dalla macchina, ad ogni passo, è determinata da un programma, in base allo stato della macchina e al simbolo letto. Un tale tipo di calcolo è essenzialmente sequenziale e "monogenico" per due aspetti cruciali. La struttura lineare che si elabora è unica e su di essa si interviene leggendo ed eventualmente alterando un simbolo alla volta. Inoltre, tutto il processo segue un programma centralizzato. Il paradigma sottostante al calcolo DNA è invece essenzialmente parallelo e "poligenico": la stessa sequenza è presente in centinaia di miliardi di copie e vari agenti di calcolo (enzimi) agiscono su di esse con operazioni globali (concatenazioni, tagli, appaiamenti, riconoscimento di porzioni), in gran parte indipendentemente l'uno dall'altro.

In una tale prospettiva, la classica differenza *software/hardware* risulta difficilmente riconducibile allo schema tipico della macchina a programma di von Neumann. In altre parole, per dare solo una prima idea iniziale, una "programmazione DNA" può assumere la forma dello pseudo-codice dato nel riquadro di p. 30, ove, in relazione ad un problema combinatorio, è descritta una procedura per manipolare delle molecole iniziali di DNA, eseguendo certe operazioni su provette. Il funzionamento di un tale paradigma prevede quindi di agire piuttosto che su una singola sequenza, su "popolazioni" di sequenze, elaborate in parallelo, e portando avanti miliardi di calcoli singoli. Ciascun calcolo esplora una possibilità risolutiva. Alla fine, se una soluzione viene determinata, questa viene selezionata secondo opportuni criteri e quindi "letta". Volendo fare un esempio, è come se, per realizzare un certo compito si costruissero tanti piccoli automi, di diversi tipi, ciascuno con una specifica funzionalità, e poi si mettessero in un ambiente dove questi incontrano varie occorrenze del problema da risolvere. Gli automi, in base alle loro caratteristiche, provano varie forme di assemblaggio e di coordinazione reciproca per aggredire "il nemico" da sconfiggere. Chi prima riesce nell'impresa "suona un campanello" e quindi esibisce la strategia vincente. In tal senso non si programma la soluzione, ma piuttosto dei comportamenti da cui la soluzione può emergere.

Negli attuali algoritmi DNA l'esecuzione di certe manipolazioni di base è per lo più devoluta all'operatore umano. Tuttavia, un aspetto sempre più emergente nella ricerca nel settore tende a sviluppare metodi in cui opportune strutture molecolari codificano programmi eseguiti da opportuni nano-agenti. In tal caso, si ha una sorta di distinzione tra *hardware* e *software*, anche se entrambi i ruoli sono giocati da opportune molecole. Del resto, semplificando alquanto, una tale differenza è quella che in natura distingue i due due tipi di biopolimeri che realizzano i processi vitali fondamentali: i geni (opportune sequenze di DNA) codificano la costruzione delle proteine (opportune sequenze di amminoacidi) e queste, a loro volta, realizzano le funzioni primarie degli organismi. Capire l'algoritmica di vari processi naturali, ed elaborare nuovi algoritmi che possano "girare" su questa sorta di "bio-ware", è una sfida completamente nuova in cui l'informatica può giuocare, a vari livelli, un ruolo insostituibile, che impone ripensamenti e nuove elaborazioni concettuali: la scoperta di algoritmi su strutture di dati non convenzionali, l'analisi di codifiche, la simulazione e la gestione computazionale di esperimenti per la realizzazione *in vitro* di calcoli molecolari.

Nel resto dell'articolo presenteremo i principi e gli strumenti sottostanti all'esperimento di Adleman, indicando gli antefatti e le prospettive che questo esperimento ha dischiuso. Quindi, concluderemo con un breve cenno alle attuali tendenze e alle linee di frontiera del *DNA Computing*.

2. RICHIAMI SUL DNA

2.1. La struttura bilineare del DNA

Il DNA (*Acido DeossiriboNucleico*) è, come noto, il componente costitutivo del materiale genetico degli organismi viventi. Le molecole di DNA sono sequenze di coppie di **nucleotidi**. Ogni nucleotide, a sua volta, risulta dalla sintesi di tre tipi di molecole: uno zucchero **Z**, un

gruppo fosforico **P** (un fosfato PO_4), e una base azotata **B**, che può essere di 4 tipi diversi. Lo zucchero chiamato ***deossiribosio*** ha 5 atomi di Carbonio usualmente numerati con 1´, 2´, 3´, 4´, 5´, ed è ottenuto da un "pentosio" $C_5 (H_2 O)_5$, detto Ribosio, eliminando un Ossigeno legato al Carbonio della posizione 2´. Il Ribosio a sua volta è presente nelle molecole di RNA, che forniscono "copie di lavoro" di quelle di DNA. Le molecole Z del DNA sono usualmente allineate in due filamenti paralleli formando una sorta di "binario". In ciascun filamento ogni molecola Z è legata alle altre molecole Z dello stesso filamento per mezzo del fosfato PO_4 con un legame chimico forte detto "fosfodiestrerico". Il complesso Z + B dicesi ***nucleoside***. Il nucleotide è quindi ottenuto aggiungendo al nucleoside il gruppo fosforico, ottenendo P + Z + B, ove l'ossidrile OH in 5´ è sostituito da un ossigeno di PO_4, (Figura 1). In tale binario il ruolo di "traversine" è giuocato dalle coppie di basi appaiate con legami "deboli" a Idrogeno: A – T, T – A, C – G, G – C (tra A e T si stabiliscono due legami a Idrogeno, mentre tra C e G ve ne sono tre). Infatti, vale la regola di appiamento di Chargaff secondo cui una base A si può appaiare solo con una T e viceversa, mentre una base C si può appaiare solo con una G e viceversa. Nei filamenti vi è un orientamento intrinseco dovuto alla natura orientata degli zuccheri Z in quanto pentosi (un pentagono è una forma naturalmente orientata). Il legame di un nucleotide con il successivo avviene lungo la direzione 5´ – 3´ in un filamento mentre avviene nella direzione 3´ – 5´ nel filamento appaiato. La disposizione che segue indica lo shema di un "binario DNA" a quattro "traversine", in cui si evidenziano gli orientamenti opposti dei due filamenti appaiati.

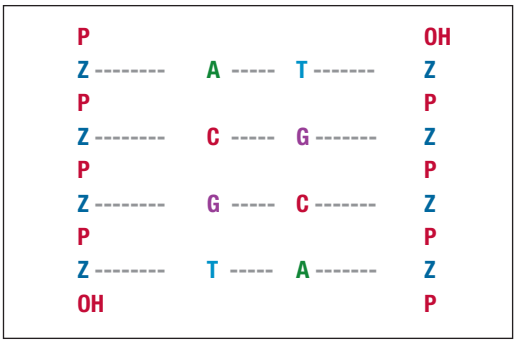


FIGURA 1
Un "binario DNA" a quattro "traversine

Dato che una lettera determina univocamente la sua appaiata, ciascuna "traversina del binario" è del tutto individuata da una sola lettera e quindi, da un punto di vista informativo, la struttura di sopra è completamente individuata dalla stringa **ACGT**.

2.2. La doppia elica DNA

Il doppio filamento di cui abbiamo detto ha uno svolgimento nello spazio secondo la tipica forma a doppia elica. Tale forma è intrinsecamente legata alla natura bilineare del DNA. Infatti, consideriamo in termini del tutto generali come si possa organizzare una doppia struttura di elementi che si susseguono su due file appaiate. Per visualizzare la cosa pensiamo ad un ballo a "pariglia" di dame e cavalieri, in cui ogni cavaliere si congiunge per mano ai danzatori della sua fila, mantenendo il viso sempre di fronte ad una stessa dama della fila opposta (e viceversa). I tre punti che determinano la disposizione di danza sono le posizioni della mano destra, della sinistra e del viso. Siccome le basi azotate che sono purine si appaiano a pirimidine (riquadro), per avere una completa analogia con l'appaiamento delle basi azotate, imponiamo che ogni cavaliere (Purina) si appaia ad una dama (Pirimidina), e che danzatori "alti" (a 3 legami H) e "bassi" (a 2 legami H) si appaino con danzatori di altezza simile. In modo analogo, per ciascun nucleotide si determinano tre punti fondamentali nella realizzazione della forma bilineare: i due punti lungo la linea

Elementi strutturali del DNA

Basi azotate: Timina, Adenina, Citosina, Guanina: T, A, C, G (A, G Purine;T, C **Pirimidine**).

Deossiribosio = Z = $C_5 H_{10} O_4$ (Pentosio con 5 atomi di Carbonio numerati: 1´, 2´, 3´, 4´ 5´).

Molecola DNA = Sequenza di coppie di Nucleotidi a basi appaiate (A- -T , C- - -G).

Nucleoside = Z(B) = Deossiribosio Z + Base Azotata B (legata al Carbonio di Z in 1´)

Z(B) = OH – (5´) – Z(B) – (3´) – OH (sono evidenziati i gruppi OH legati al Carbonio di Z in 3´ e 5´).

Nucleotide = Nucleoside + Fosfato = Z(B) + PO_4 = PO_4 – (5´) – Z(B) – (3´) – OH.

Concatenazione di nucleotidi ove -- indica il legame fosfodiesterico: PO_4 – (5´) – Z(B) – (3´) – OH + PO_4 – (5´) – Z(B) – (3´) – OH = PO_4 – (5´) – Z(B) – (3´)-- PO_4 – (5´) – Z(B) – (3´) – OH.

Bilinearità = Appaiamento di due filamenti.

Complementarietà = T e A si appaiano con 2 legami H; C e G si appaiano con 3 legami H.

Antiparallelismo = Due filamenti appaiati sono orientati in senso opposto (5´ – 3´) e (3´ – 5´).

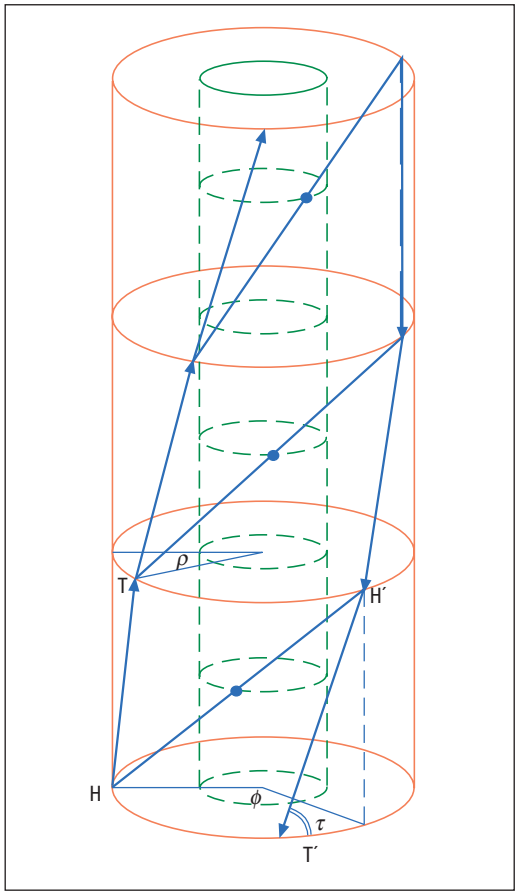


FIGURA 2
Lo sviluppo a doppia elica di una struttura lineare doppia

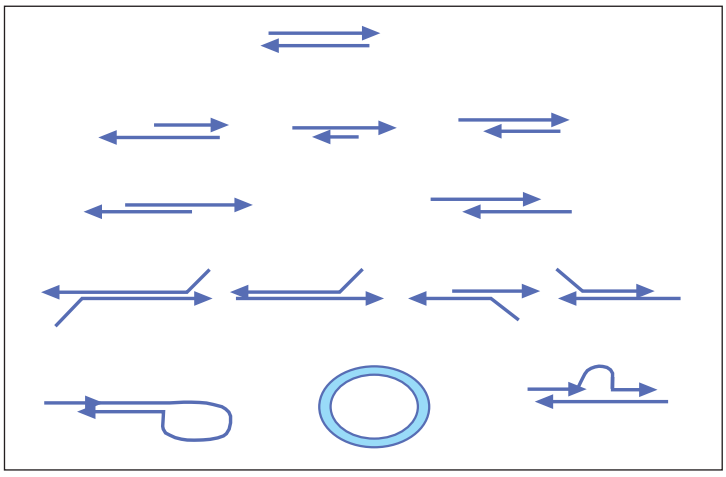


FIGURA 3
Tipi basilari di forme bilineari

di concatenazione fosfodiesterica, diciamoli Testa e Coda, e il punto intermedio tra la propria testa e quella del nucleotide ad esso appaiato. In definitiva, il modulo compositivo che rende completamente conto della doppia struttura lineare si riduce in termini semplificati ad un triangolo, e le coppie allineate di triangoli si possono pensare come sviluppate entro il cilindro di figura 2, ove, per semplificare ulteriormente, si assume che le dimensioni dei triangoli siano sempre le stesse. Nella figura 2 i simboli H e T indicano la testa e la coda di ciascun elemento compositivo, mentre H´ e T´ sono la testa e la coda dell'elemento appaiato. Si verifica che, se l'angolo formato dalla direzione di concatenazione con quella di appaiamento è acuto (come richiesto da una maggiore ottimizzazione spaziale), allora i due filamenti devono necessariamente essere orientati in modo opposto [14]. Inoltre, per evitare che i legami di concatenazione e di appaiamento di entrambi i nucleotidi siano sullo stesso piano (cosa impossibile in un ambiente fluido) deve esservi una rotazione tra i piani dei due triangoli relativi, e quindi la doppia struttura deve svolgersi nello spazio. I tre angoli ρ, τ, ϕ, indicati in figura 2, insieme al raggio del cilindro, identificano completamente lo schema della doppia elica sviluppata intorno al cilindro [14, 19]. La struttura polimerica (etimolog. "di molte parti") doppia ha un profondo significato computazionale. Infatti, sull'appaiamento di porzioni complementari sono basate le principali operazioni tra filamenti di DNA, ed in particolare la duplicazione, che è un'operazione fondamentale per I meccanismi ereditari della vita, e che, come vedremo, consente la realizzazione di procedimenti efficienti per produrre un numero esponenziale di copie di una data molecola (per mezzo di un algoritmo su doppie sequenze che produce, in *n* passi, 2^n copie di molecole uguali ad una singola molecola data). La *bilinearità,* la *complementarietà* e l'*antiparallelismo* determinano una ricchezza di possibili forme DNA del tutto inimmaginabile. Un recente campo in rapida evoluzione del *DNA computing* si basa sullo studio di combinazioni inedite di molecole che, sfruttando tali principi costitutivi, possano produrre molecole autoaggreganti composte da "pezzi" elementari e con caratteristiche spaziali di interesse per applicazioni in vari settori delle nanotecnologie. Tale campo di indagine che va sotto il nome di *DNA Self-Assembly* ha a sua volta aspetti di interesse computazionale, in quanto si dimostra che la stessa nozione di calcolo puo essere sviluppata in termini di generazione di forme secondo moduli di assemblaggio predefiniti [22, 24]. Per dare solo una semplice idea della ricchezza combinatoria delle strutture bilineari, riportiamo nella figura 3 le varie forme bilinea-

ri di base che si possono venire a creare (il parallelismo indica complementarità e il non parallelismo non complementarità). Le ultime tre in fondo sono le cosidette forme, *hairpin*, circolari e *heteroduplex* (negli *heteroduplex* si formano delle lacune di appaiamento all'interno di due filamenti appaiati, mentre gli *hairpin* sono realizzati da uno stesso filamento in cui una parte ibridizza con un altra sua parte). Tali forme possono combinarsi producendo una varietà enorme di possibilità, inoltre esistono almeno tre forme diverse di avvolgimenti elicoidali, vi sono casi di intrecci a più di due filamenti, vi sono avvolgimenti esterni, intorno ad assi trasversali a quello dell'elica e meccanismi di compattamento per ottimizzare l'occupazione di spazio quando il DNA non è in fase di elaborazione.
Le molecole di DNA hanno un intrinseco carattere *informazionale*, ed in effetti il loro ruolo biologico è quello di memorizzare i "programmi" che dirigono il funzionamento degli organismi viventi, dai più semplici ai più complessi. Quindi non stupisce che il DNA possa essere considerato come il supporto naturale per processi di calcolo. Ma in questo caso come manipolarlo per svolgere calcoli matematici? E soprattutto, quale è il vantaggio di un tale uso? Da un punto di vista tecnico, le stringhe non sono altro che sequenze di simboli tra le quali è definita un'operazione di *concatenazione*, la stessa che a partire dalle parole "capi" e "tombolo" produce la parola "capitombolo". Anche i calcoli che si svolgono all'interno di un computer sono in ultima analisi riconducibili a manipolazioni di stringhe (dei due simboli 0, 1). Tuttavia, la realtà fisica di tali stringhe è completamente diversa e le operazioni che su di esse si eseguono sono di tutt'altra natura. In definitiva, calcolare con il DNA richiede un ripensamento della stessa nozione di calcolo, ricostruendo in termini nuovi i metodi risolutivi degli algoritmi tradizionali. Questo è di per sè il primo grande interesse del *DNA Computing* nel quadro della ricerca di nuovi modelli di calcolo.

3. CALCOLI NATURALI

Un calcolo si svolge sempre con il supporto di un qualche sistema fisico. Tale sistema viene configurato in modo da codificare i dati da elaborare, partendo da un suo stato "iniziale", quindi lo si fa evolvere con una serie di passi, o agendo sullo stato di ogni passo con un "comando" che ne induce una transizione, oppure secondo un "programma" interno al sistema che induce una sequenza di transizioni a partire da uno stato iniziale. Quando il sistema giunge ad uno stato finale (secondo un certo criterio di terminazione), allora si decodifica tale stato estraendo da esso i risultati del calcolo. Se il sistema di supporto del calcolo è un sistema naturale, o ispirato a qualche sistema della natura, si parla di "calcolo naturale".
La ricerca di modelli di calcolo diversi da quelli sviluppati a partire dagli anni '30 del Novecento, secondo il paradigma di Turing-von Neumann, ha seguito vari percorsi ed è stata sollecitata da vari problemi. Già negli anni 40' e '50 varie metafore biologiche avevano ispirato strutture che si rivelarono di importanza centrale in vari contesti informatici (Reti neurali di McCulloch e Pitts nel 1943, Automi a stati finiti di Kleene nel 1956).
Il successivo sviluppo tecnologico del modello di von Neumann, per quanto poderoso, ha indicato dei limiti fisici intrinseci a cui la tecnologia odierna si sta sempre più avvicinando. Inoltre, la teoria matematica della complessità ha dimostrato che una grande quantità di problemi interessanti rimangono al di là delle possibilità del calcolo tradizionale.
Nei procedimenti generativi delle grammatiche di Chomsky, introdotte negli anni '60, la produzione di certe forme avveniva secondo meccanismi di controllo decentralizzato ove, tra tutte le possibili parole generate, quelle in cui le regole venivano applicate in modo "corretto" riuscivano a "terminalizzare", mentre le altre producevano forme "immature" che venivano scartate quando si "racccoglievano" i risultati [21]. In genere, nei processi di elaborazione di *stringhe* (sequenze a struttura concatenativa) si evidenziavano metodi in cui l'informazione fluiva secondo strategie "evolutive" che non erano quelle tipiche dei "calcoli a programma".
Alla fine degli anni '60, nel quadro della teoria dei linguaggi formali Lindenmayer sviluppò dei sistemi, poi detti L sistemi, che usando metodi di manipolazione di stringhe, tipici della teoria dei linguaggi formali, permisero di derivare gli stadi di sviluppo di organismi bio-

logici quali alghe (l'alga rossa fu il primo organismo modellato con tali sistemi) [21].
Gli algoritmi genetici sviluppati negli anni '70 [12], gli automi cellulari sviluppati negli anni '80 [25], a partire da studi iniziati da von Neumann, e le reti booleane, introdotte per modellare la reti genetiche, furono altri modelli a forte "ispirazione naturale".
Nel 1987 Tom Head introdusse un'operazione su stringhe, detta di *splicing*, che formalizzava il meccanismo di ricombinazione genetica e furono dimostrate interessanti relazioni con importanti classi di linguaggi formali [11, 16]. In particolare lo *splicing* individua un meccanismo di trasformazione combinatoria di stringhe completamete diverso dal rimpiazzamento tipico delle grammatiche di Chomsky e degli automi di riconoscimento (rappresentati come strutture di manipolazione di stringhe). Due stringhe, diciamole α, β, si ricombinano per *splicing* $u_1\#u_2\$u_3\#u_4$ (è il modo standard di indicare le regole di *splicing*, e purtroppo in biologia molecolare il termine *splicing* indica un fenomeno diverso) se, possono essere fattorizzate, per opportune stringhe x, y, w, z, rispettivamente in $\alpha = xu_1\,u_2\,y$, $\beta = wu_3\,u_4\,z$. In tal caso la regola $u_1\#u_2\$u_3\#u_4$ produce le nuove stringhe $xu_1\,u_4\,z$, $wu_3\,u_2\,y$. Tale operazione ha un evidente rilevanza biologica e per visualizzarla in modo semplice basti pensare alle chimere tipiche della fantasia mitologica (sirene, centauri, minotauri, grifoni, capricorni) in cui partendo da due animali, ricombinadone le parti si ottengono nuove forme animali. Le coppie stringhe (u_1, u_2) e (u_3, u_4) sono i cosidetti "siti di riconoscimento" e individuano i punti in cui tagliare le stringhe da ricombinare. Un risultato notevole stabilisce che a partire da un insieme finito di stringhe e usando un insieme finito di regole di splicing si possono ottenere linguaggi formali in una classe strettamente inclusa nella classe dei linguaggi regolari (lemma di subregolarità [20, 16]). Tuttavia, aggiungendo delle estensioni del tutto naturali, tale meccanismo diventa *computazionalmente universale*, ovvero permette di ottenere sistemi con la stessa potenza di calcolo delle machine di Turing [20].
In tale linea di ricerca, le strutture discrete di rappresentazione dell'informazione nei modelli di calcolo mostrano sorprendenti analogie con la natura discreta dell'informazione genetica. E anche da un punto di vista cronologico, si nota uno svolgimento parallelo di "modelli digitali" che da una parte produce le prime macchine di calcolo a programma e dall'altro conduce, nel 1953, alla scoperta del modello di DNA, prima descritto, che portò al Nobel di Watson e Crick. In questa prospettiva, l'esperimento di Adleman del 1994, al di là del suo significato specifico e degli sviluppi che potrà avere, sembra l'epigono di una convergenza naturale tra discipline in cui l'informazione è un concetto fondante. Ed in un certo senso, il *DNA Computing* mostra un verso di interazione tra Informatica e Biologia che è complementare all'interazione Info-Bio propria della Biologia computazionale, alla base dei successi straordinari dei sequenziamenti genomici su grande scala ottenuti negli anni recenti.
Per concludere, è opportuno ricordare che nel 1998 Gheorghe Paun, ispirandosi ai meccanismi di compartimentazione di popolazioni di molecole nelle membrane biologiche, ha introdotto i *P* sistemi [21]. In essi sono espresse, in termini simbolici, sia regole che trasformano oggetti, sia regole che li spostano da una membrana ad un'altra. Con oppurtuni adattamenti, tali modelli risultano adeguati nella rappresentazione di dinamiche biologiche, fornendo alternative promettenti ai classici modelli basati su equazioni differenziali [5, 15].

4. OPERAZIONI DI BASE SU DNA

Diamo nel seguito una descrizione sintetica delle operazioni su molecole di DNA che risultano fondamentali nel *DNA Computing* [20, 8] (riquadro di p. 25). Sorvoleremo sui dettagli biochimici e di laboratorio che talvolta sono molto complessi, soffermandoci solo sugli aspetti informazionali di elaborazione di sequenze e doppie sequenze.
Elementi fondamentali per realizzare le operazioni DNA sono: nucleotidi e molecole DNA, *contenitori* (*provette, piastre*, ...), *sorgenti di energia ed enzimi, oltre a reagenti, supporti e strumenti* per la misurazione di unità chimico-fisiche, per la rilevazione e visualizzazione di molecole e per la attivazione e la regolazione dei processi biochimici. Gli enzimi dal punto di vista biochimico sono dei *catalizzatori* di pro-

Operazioni di base nel DNA Computing

1. **Denaturazione** = Disaccoppiamento di sequenze nucleotidiche appaiate.
2. **Rinaturazione** = Riaccoppiamento di sequenze nucleotidiche disaccoppiate.
3. **Unione** = Versamento del contenuto DNA di una provetta in un'altra contenente altro DNA.
4. **Divisione** = Distribuzione del contenuto di una provetta in due provette.
5. **Ibridizzazione** = Appaiamento, per complementarietà, di due singoli filamenti DNA.
6. **Amplificazione** = Generazione di molte copie uguali di una data porzione di DNA (doppio).
7. **Sequenziamento** = Lettura della sequenza di basi che caratterizza una data molecola di DNA.
8. **Sintesi** = Produzione sintetica di molecole di DNA aventi una specificata sequenza di basi.
9. **Gel-Elettroforesi** = Discriminazione di molecole di DNA, in bande di uguale lunghezza.
10. **Separazione** = Estrazione, dopo elettroforesi, di una banda di molecole di data lunghezza.
11. **Selezione per affinità** = Estrazione di molecole di DNA che includono una data sottosequenza.
12. **Ligasi** = Concatenazione fosfodiesterica tra due filamenti contigui appaiati ad uno stesso filamento.
13. **Estensione** = Allungamento nel verso 5´-3´ "copiando" dal filamento antiparallelo appaiato.
14. **Blocco** = Preclusione di concatenazione nell'estremo OH o PO_4 (con possible sblocco successivo).
15. **Ancoraggio** = Adesione di un frammento singolo di DNA ad un supporto solido.
16. **Restrizione** = Taglio, tramite enzima, di doppi filamenti DNA che contengono una porzione specifica.

cessi biomolecolari, ma dal punto di vista informazionale fungono da veri e propri "automi di calcolo biomolecolare", specializzati in operazioni elementari di riconoscimento, attivazione e regolazione di specifici processi di elaborazione di biopolimeri. Essi sono particolari proteine e quindi codificati da opportuni geni. Oltre alla *ligasi*, alla *polimerasi*, sono importanti gli enzimi di *restrizione*, che tagliano sequenze contenenti certe specifiche sottosequenze (suddividendole opportunamente tra le sequenze prodotte dal taglio). Tale caratteristica si presta a vari usi, di cui alcuni molto sofisticati, ma sopratutto consente di frammentare porzioni molto lunghe di DNA in modo da potere elaborare piu facilmente i frammenti prodotti, cosa che risulta fondamentale nel sequenziamento di interi genomi. Altre classi di enzimi realizzano operazioni più complesse di cui non parleremo affatto, citiamo solamente alcuni nomi: *trascrittasi inversa, terminal-trasferasi, topoisomerasi, integrasi, elicasi*. Da un punto di vista matematico, gran parte delle operazioni DNA che consideriamo non agiscono su singole molecole, ma piuttosto su popolazioni di molecole (a doppio o singolo filamento) e anche gli enzimi occorrono in popolazioni, in cui ciascuna copia dello stesso enzima svolge lo stesso compito dei suoi simili, ed in parallelo ad essi, sulle molecole appropriate alla sua funzione. Ciascuna molecola DNA costituisce una occorrenza individuale di un tipo espresso da una sequenza (singola o doppia) di basi (una parola). Tale nozione di popolazione corrisponde al concetto matematico di *multinsieme* in cui ciascun elemento (filamento) non solo appartiene o non appartiene ad un multinsieme, ma vi occorre in un certo numero di copie (l'inclusione, l'unione, la differenza ed altre operazioni insiemistiche si estendono in modo standard ai multinsiemi). Ad ogni popolazione *P* di molecole DNA è associato quindi un insieme di stringhe (singole o doppie), diciamolo Type(*P*), ovvero un *linguaggio formale*, costituito da tutti i tipi delle molecole di *P*. L'effetto di molte operazioni consiste nel passaggio da una popolazione con un certo tipo ad un'altra con un altro tipo. In questo senso, spesso non è importante sapere esattamente come è fatta una certa popolazione, ma solamente conoscerne il tipo ed essere in grado di manipolarla in modo da farle assumere il nuovo tipo desiderato.

Quando le molecole di DNA si trovano in un ambiente in cui la temperatura supera valori intorno agli 80° Celsius, esse si ***denaturano***, ovvero si scindono nei due filamenti singoli. La temperatura a cui questo avviene dipende da parametri chimico-fisici determinati dalle sequenze di basi. Abbassando la temperatura i due filamenti si riappaiano in modo tale che ciascun nucleotide si lega con il suo complementare. Il processo inverso alla denaturazione si dice ***rinaturazione***. La figura 4 illustra la denaturazione e la rinaturazione di un doppio filamento. Si consideri una situazione in cui si hanno filamenti singoli, per esempio

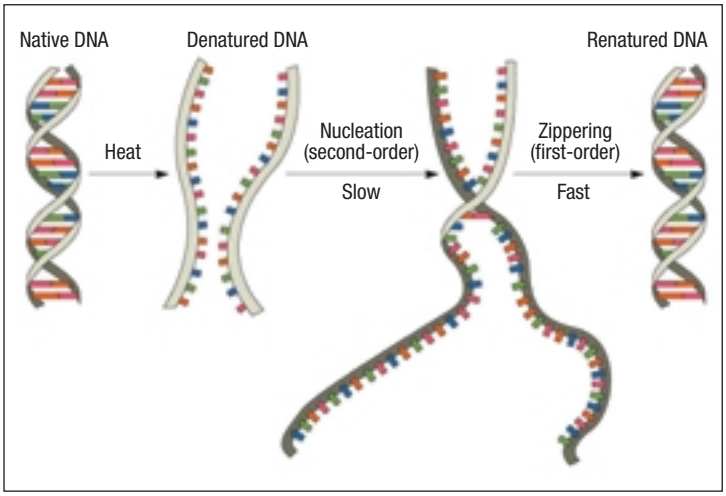


FIGURA 4

Denaturazione e Rinaturazione (da [10])

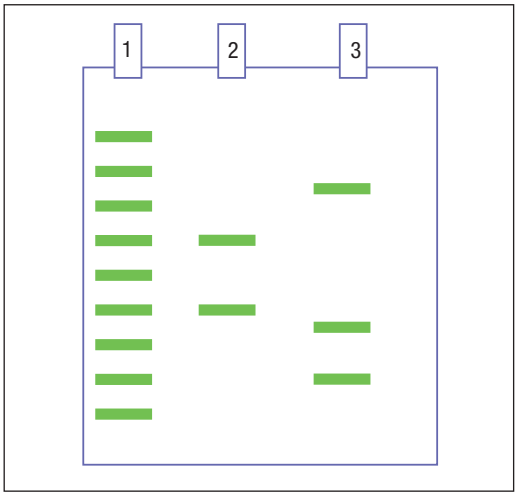


FIGURA 5

Forma stilizzata di una lastra radiografica di rilevazione dopo gel-elettroforesi. In colonna 1 è posto il ladder di marcatori. Le lunghezze (in bp) delle bande si valutano confrontandone la posizione rispetto ai marcatori.

di tipo NNNNNCACTTGNNNNN, in cui N indica un generico nucleotide (la scrittura si sottintende nel verso 5´-3´). Allora, ponendo frammenti a singolo filamento di tipo CAAGTG, che è antiparallelo e complementare a CACTTG (cioè complementare alla sua sequenza invertita), sotto appropriate condizioni termiche, si ottiene l'appaiamento tra CAAGTG e NNNNNCACTTGNNNNN secondo una struttura che possiamo rappresentare come segue (nella sequenza inferiore, si sottintende il verso 3´-5´):

$$\frac{\text{NNNNNCACTTGNNNNN}}{\text{GTGAAC}}$$

Tale fenomeno si chiama *ibridizzazione* o *annealing*. Le operazioni di appaiamento e disappaiamento di nucleotidi sono possibili perchè il legame chimico che lega le basi appaiate è un legame a Idrogeno notevolmente più debole rispetto agli altri legami interni ai nucleotidi e ai legami fosfodiesterici che legano gli zuccheri. Quando, dopo una ibridizzazione, due porzioni di DNA si trovano ad essere contigue in uno stesso filamento, un enzima *ligasi* svolge il compito di creare un legame fosfodiesterico tra i nucleotidi che si trovano accanto. Tale operazione può essere considerate una forma particolare di concatenazione di sequenze, nel contesto dei doppi filamenti.

L'operazione di *gel-elettroforesi* consiste nel depositare il DNA di una provetta in una piastra ai cui bordi è applicata una differenza di potenziale elettrico e sulla superficie è stato depositato un gel opportuno (di Agarosio o Poliacrilammide). In tal modo, in base al fatto che le molecole di DNA hanno una leggera carica negativa (nel gruppo fosforico), si ha una migrazione dal polo negativo a quello positivo la cui velocità è inversamente proporzionale alla lunghezza, in quanto la maggiore lunghezza determina un maggiore attrito sulla superficie del gel. Dalla posizione in cui le molecole si trovano, dopo una "corsa" che dura un tempo assegnato, si risale alla lunghezza delle molecole. Infatti, dopo la corsa, le molecole si ragguppano in bande di uguale lunghezza che si evidenziano attraverso rilevatori di radiazioni che impressionano delle lastre radiografiche (Figura 5). L'altezza delle bande rilevate è posta in relazione all'altezza di bande di riferimento, di lunghezze note, e quindi si riesce a risalire con notevole precisione alla lunghezza delle bande ottenute. In tal modo, oltre a suddividere le molecole per lunghezza, si può separare una banda di una certa lunghezza. Basta incidere la parte di gel in cui si trova una certa banda e quindi, dopo aver rimosso il gel, liberare il DNA in esso contenuto riportandolo in provetta.

L'operazione di *sequenziamento* corrisponde alla lettura di una sequenza e si realizza con un metodo (dovuto a Sanger, che per questa scoperta ha ottenuto il Nobel per la seconda

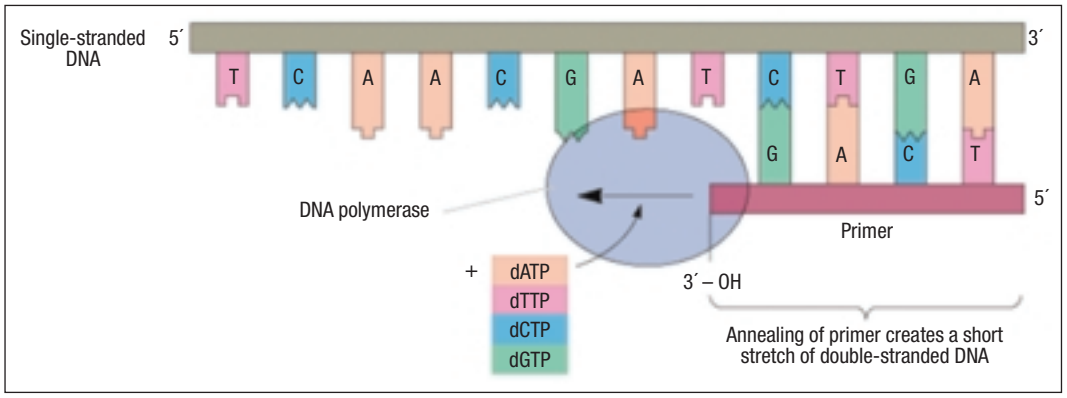


FIGURA 6
L'enzima Polimerasi in azione (da [10])

volta) che descriveremo solo in modo metaforico. A prima vista potrebbe sembrare che la lettura di una sequenza si possa fare in qualche modo non troppo complicato, magari con qualche ultrapotente miscroscopio. Ma non è così. In tutti i modelli nanotecnologici, leggere è complicato e dispendioso, molto più che scrivere! Dapprima si producono molte copie della sequenza da leggere, quindi si dividono tali copie in 4 provette distinte cha diremo $\mathbf{P}_T$, $\mathbf{P}_A$, $\mathbf{P}_C$, $\mathbf{P}_G$ in ciascuna delle quail si usa un qualche meccanismo che ha l'effetto di tagliare le sequenze in un solo punto: in $\mathbf{P}_T$ sempre dopo una base **T**, in $\mathbf{P}_A$ sempre dopo una base **A**, in $\mathbf{P}_C$ sempre dopo una base **C**, e in $\mathbf{P}_G$ sempre dopo una base **G**. Quindi utilizzando il processo di gel-elettroforesi si determinano le lunghezze di tutti I frammenti presenti in $\mathbf{P}_T$ e analoganente le lunghezze di quelli in $\mathbf{P}_A$, $\mathbf{P}_C$, $\mathbf{P}_G$. Dalle misure di tali lunghezze si scoprono quindi le posizioni delle varie basi nella sequenza originale. In definitiva, è come se, volendo leggere una parola se ne facesserto miliardi di copie e successivamente si distruggessero, risalendo dai resti di tale distruzione alla parola data. Ovviamente i dettagli biochimici di tale processo sono piuttosto complessi, e sopratutto è cruciale modulare il meccanismo dei tagli per essere sicuri di realizzarne un numero adeguato, in tutte le posizioni in cui essi sono possibili. Comunque, si può notare che, nell'essenza, il metodo di sequenziamento è un algoritmo su sequenze, o meglio su multinsiemi di sequenze.

Se esiste un'operazione per "leggere" una sequenza, deve esserci anche un'operazione per potere "scrivere" una sequenza data. Tale operazione corrisponde alla *sintesi* di sequenze DNA, ovvero al procedimento secondo cui, da una data sequenza simbolica si produce un *clone* di una molecola del tipo specificato dalla sequenza. Il termine clone indica semplicemente che si tratta di una certa quantità di molecole (per esempio una pico-mole), tutte dello stesso tipo. Un tale processo ormai avviene in modo molto efficiente usando iterativamente tre meccanismi di base:

i) l'*ancoraggio* di molecole ad un supporto solido,

ii) il *blocco* di concatenazione in posizione 5′ dei nucleotidi, suddivisi secondo i loro quattro tipi,

iii) il successivo *sblocco,* non appena l'estremo in 3′ di un nucleotide si è concatenato alla sequenza ancorata. Senza entrare in ulteriori dettagli, possiamo dire che di fatto basta ordinare (ormai *online*) ad una ditta specializzata la sequenza desiderata per ottenere, ad un prezzo inferiore ad un dollaro per base, una certa quantità molare di molecole del tipo richiesto (provate a fare esperimenti con 20 sequenze da 100 basi, reagenti, enzimi, ... !).

L'operazione *extract* è fondamentale per il DNA computing. Essa consiste in una *selezione per affinità*, che permette di estrarre da una provetta molecole che contengano una data porzione di DNA. Un modo per realizzare tale selezione è quello di usare sequenze complementari (e antiparallele) alla porzione da "pescare", ancorandole a opportuni supporti solidi. In tal modo, dopo aver favorito l'ibridazione, si estraggono dalla soluzione i supporti solidi e quindi con essi anche le molecole ibridate ai frammenti ancorati.

L'*estensione* è un'operazione chiave svolta dall'enzima *polimerasi*. Tale enzima estende un filamento, nel verso 5′-3′, usando l'altro filamento come stampo (template). La figura 6 visualizza tale meccanismo. Una volta che una

piccola sequenza è appaiata ad un singolo filamento (*primer*, ovvero innesco), la parte mancante del filamento viene ricostruita dall'enzima polimerasi se nella soluzione sono presenti dei nucleotidi singoli (in forma trifosfata: dATP sta per "deossiadeninatrifosfato", dTTP sta per "deossitiminatrifosfato", e via dicendo) che vengono correttamente allineati secondo la regola di appaiamento di Chargaff. Piu precisamente, in soluzione si pongono deossinucleosidi 5´-trifosfato, in cui rispetto ai nucleotidi, in 5´ vi è $P_3 O_{10}$ (un trifosfato al posto di un fosfato PO_4, vedi riquadro a p. 21). Quindi il deossinucleoside 5´-trifosfato entra nella catena trattenendo un PO_4 che realizza il legame fosfodiesterico con il nucleotide a cui si concatena e da questo elimina l'ossidrile OH, rilasciando quindi un pirofosfato $P_2 O_6 OH$ (l'ossidrile in 3´ del nucleotide a cui si concatena viene sostituito da un ossigeno del PO_4).
Operazioni semplici da realizzare, ma estremamente utili nella costruzione di algoritmi DNA sono *merge* e *split*, ovvero *unione* e *divisione*. Con la prima si mescolano in un'unica provetta i contenuti di due provette distinte, con l'altra si divide in due provette il contenuto di una (in quantità pressochè uguali).
Concludiamo la rassegna di operazioni DNA con un cenno al metodo principale di *amplificazione*, che permette di produrre molte copie di una sequenza di DNA. La sua realizzazione attraverso il procedimento della *PCR* (*Polymerase Chain Reaction*) fu scoperta nel 1983 da Kary Mullis (laureato Nobel per questa scoperta nel 1993). Lo sviluppo di tutta la biologia molecolare è impensabile senza la PCR. Altri metodi di amplificazione, che rimangono tuttora attuali, sono basati sulla *clonazione* tramite vettori di clonazione (batteri e virus), ma richiedono procedimenti molto complessi su cui non possiamo soffermarci. L'idea geniale di Mullis fu quella di utilizzare l'estensione della polimerasi per "fotocopiare" una porzione voluta di un doppio filamento di DNA, delimitata da due primer. Il metodo usa un'idea astuta di regolazione della temperatura. Riferendoci alla figura 7 (relativa ad un caso semplice), si parte con una molecola perfettamente bilineare, diciamola *bersaglio*, e da due primer (usualmente di circa 20 bp) che indichiamo γ e $rev(\delta^c)$, ove δ^c indica la sequenza complementare δ e *rev* indica l'inversione di sequenze. Assumiamo che il primer γ indicato a sinistra si appai, per ibridizzazione, al filamento inferiore del bersaglio, e che quello indicato a destra $rev(\delta^c)$ si appai al filamento superiopre del bersaglio. Regolando opportunamente le temperature, si ottengono i passi indicati nella figura 7:
i) il doppio filamento bersaglio si scompone nei suoi due filamenti,
ii) i due *primer* ibridizzano con le porzioni ad essi complementari della molecola bersaglio ii) la polimerasi estende i *primer* ibridati ciascuno nel verso 5´ – 3´ in modo da produrre due molecole uguali. Riapplicando tali tre passi (denaturazione, ibridizzazione dei *primer*, estensione dei *primer* ibridati) per altre *n* volte, alla fine del processo si ottengono 2^n molecole che iniziano con γ e che terminano con δ includendo la porzione che, nella molecola ottenuta al primo passo, era compresa tra γ e δ. Questo processo assume che nella provetta si trovino una quantità adeguata di:
i) nucleotidi singoli per alimentare l'estensione della polimerasi;
ii) unità di enzima polimerasi in modo che al generico passo *k* ogni singolo enzima possa, in parallelo con gli altri, condurre il processo di estensione sulle molecole bersaglio presenti in quell'istante.

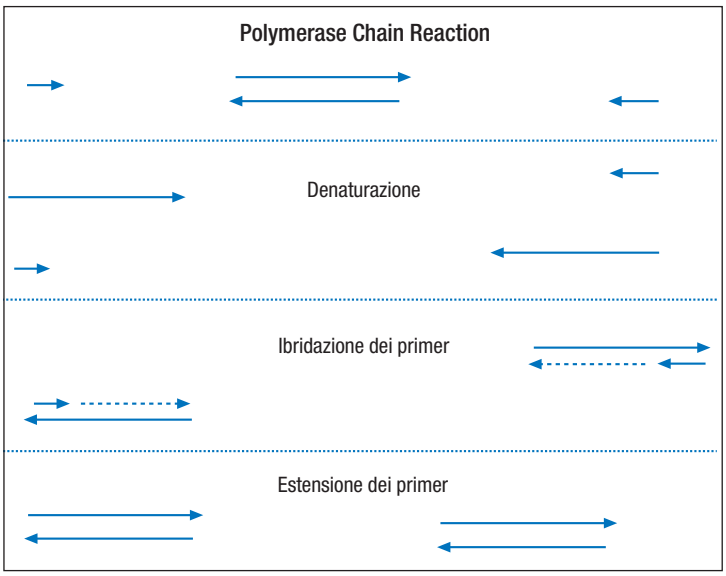


FIGURA 7
Fasi di un processo di PCR

5. L'ESPERIMENTO DI ADLEMAN

L' esperimento di Adleman prende le mosse da un tipico problema combinatorio detto del *cammino hamiltoniano* (dal matematico Hamilton che lo ha studiato). Dato un grafo orientato con *vertici* tra i quali vi sono degli *archi* di collegamento (orientati dal primo al secondo) si chiede di determinare, se esistono, dei percorsi che, partendo da un vertice iniziale, raggiungono un vertice finale passando per ciascun vertice una ed una sola volta (e attraversando gli archi secondo il loro orientamento). Si dimostra che al crescere del numero dei vertici la complessità di tale problema diventa proibitiva. In termini più precisi, possiamo dire che verificare se un dato cammino è hamiltoniano può essere fatto in un numero di passi linearmente proporzionale al numero di vertici del grafo, ma i possibili cammini da esaminare sono in numero talmente grande che se si dovessero considerare uno alla volta si impiegherebbero tempi superiori ad ogni reale possibilità. Quello del cammino hamiltoniano è un esempio della classe di problemi cosiddetti NP-completi (NP = *Nondeterministico Polinomiale*).

La soluzione di Adleman è molto intuitiva. Identifichiamo i vertici del grafo con i numeri 1, 2, ..., *n* e codifichiamoli con frammenti di DNA ad un solo filamento costituiti da due parti (tranne il vertice iniziale 1 e quello finale *n*). Siano B1, A2 B2, A3 B3 ...., An le codifiche dei vertici con parole di DNA a singolo filamento. Un arco che collega il vertice i con quello j è allora codificato con una parola DNA, a singolo filamento, di due parti Aj^c Bi^c ove Aj^c e Bj^c sono le sequenze DNA complementari ai pezzi Aj e Bj delle codifiche dei nodi i e j (in verso 3´–5´). A questo punto, ponendo in provetta le codifiche DNA di vertici e di archi, queste, sfruttando il fenomeno di ibridizzazione, si compongono in strutture doppie (Figura 8) che utilizzando l'enzima ligasi si trasformano in doppi filamenti DNA. Ciascuno di questi rappresenta ovviamente un cammino nel grafo da 1 a *n*. Se vi sono moltissime copie di ciascuna codifica (diciamo 10^{14}) è auspicabile che tutti i possibili cammini possano essere prodotti.

L'algoritmo risolutivo di Adleman procede ad estrarre i cammini hamiltoniani con i passi seguenti:

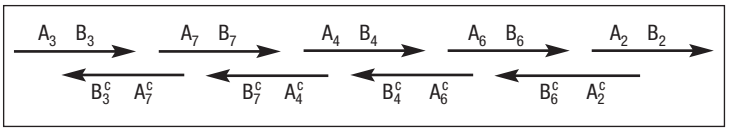


FIGURA 8

La formazione, per ibridizzazione, di un cammino di Adleman: le frecce superiori codificano i nodi, mentre le inferiori codificano gli archi. Gli spazi indicano la mancanza dei legami fosfodiesterici, poi realizzati dall'enzima ligasi

1. Si pongono in provetta le codifiche dei nodi e degli archi, favorendo la formazione di cammini del tipo riportato nella figura 8;
2. Si esegue un'amplificazione tramite PCR con primer B1 e rev(A_n^c), in modo da incrementare il numero di filamenti che codificano cammini che iniziano e finiscono come richiesto dal problema;
3. Si separano con gel-elettroforesi i filamenti di lunghezza corrispondente a quella dei cammini hamiltoniani (in generale, $(2kn - 2k) = 2k(n-1)$ ove *k* è la lunghezza dei pezzi A e B);
4. Si selezionano per affinità i filamenti in cui occorrono le codifiche di tutti i vertici;
5. I filamenti rimasti in provetta codificano i cammini hamiltoniani cercati.

La separazione del terzo passo consente di escludere cammini che sicuramente non possono rappresentare soluzioni, in quanto non avendo una lunghezza data non possono passare una ed una sola volta da tutti i vertici del grafo. La selezione del quarto passo invece serve ad accertarsi che tutti i nodi siano presenti nei cammini. Mettendo insieme le condizioni soddisfatte al terzo e al quarto passo, si garantisce che sicuramente un vertice appare una ed una sola volta in un cammino. Inoltre la amplificazione eseguita al secondo passo consente di ritenere trascurabile la possibilità di selezionare un cammino che non cominci e che non finisca nel modo richiesto. Nel caso che vi sia un solo cammino hamiltoniano (come nell'esperimento originale di Adleman) tale filamento viene sequenziato per "leggere" la sua sequenza di basi. Nel caso che vi possano essere più cammini hamiltoniani il procedimento si complica, ma si possono comunque determinare tutti i diversi cammini prodotti come risultato[1].

[1] Vi sono molte ottime e dettagliate presentazioni dell'esperimento di Adleman, facilmente reperibili in rete ricercando *Adleman experiment o DNA computing*.

6. OLTRE IL MODELLO DI ADLEMAN

È stato calcolato che un Joule è sufficiente per realizzare 2.10^{19} operazioni di ligasi, laddove un potente PC attuale arriva a circa 10^9 operazioni elementari per Joule, e che si possono svolgere circa 2.10^{18} operazioni DNA al secondo contro le 2.10^{12} di un attuale processore. Infine in pochi grammi di DNA si potrebbero codificare tutte le informazioni attualmente registrate sui supporti elettronici sparsi nel mondo. Tali valutazioni numeriche, sebbene formulate in termini del tutto astratti e sotto ipotesi particolari, indicano la "potenza informazionale" del DNA. Del resto, parlando sempre in termini generali, se si considerano i genomi degli organismi superiori ci si rende conto che la quantità di informazione digitale di un genoma (circa 3.10^9 basi nell'uomo) risulta del tutto paragonabile a quella di un sistema operativo moderno (parecchie decine di milioni di linee di codice). Quindi, ogni cellula si porta dietro il sistema operativo dell'organismo di cui fa parte come gli impiegati di un ufficio hanno lo stesso PC, anche se ciascuno fa girare solo un certo tipo di programmi legati all'attività che svolge nell'ufficio.
A partire dal 1995 si sono tenute delle conferenze internazionali su *DNA Based Computers* (siamo già alla dodicesima conferenza di tale ciclo). In un lavoro dello stesso anno Lipton, basandosi sul risultato di Adleman, definì lo schema generale di un algoritmo DNA per la soluzione di problemi combinatori [13]:
i) generazione dello spazio delle "possibili" soluzioni, codificate con filamenti DNA,
ii) estrazione delle "vere" soluzioni tramite meccanismi opportuni di selezione.
In tale schema la massiva ricombinazione del DNA è il punto cruciale per generare le soluzioni possibili a partire da frammenti di DNA che codificano i dati del problema. Tale schema è stato applicato in moltissimi modi, sopratutto per la soluzione di una famosa classe di problemi combinatori detti SAT (*Boolean Satisfiability Problem*), ed in particolare 3-SAT (*n*, *m*), in cui *n* ed *m* indicano numeri interi positivi, relativi ad un sistema di equazioni booleane (0,1 con somma, e negazione booleana) di n variabili ed m equazioni, in ciascuna delle quail occorrono al più 3 variabili. Nel riquadro è fornito un esempio di soluzione di un problema SAT, riconducibile allo schema di Lipton. Non entriamo nel dettaglio dell'algoritmo. Esso usa un interessante metodo, detto *mix-and-split*, che genera lo spazio delle soluzioni in un tempo linearmente proporzionale al numero di variabili delle equazioni booleane da risolvere. Ci limitiamo a sottolineare che:
i) si tratta di un procedimento a passi basato su operazioni che agiscono su provette;
ii) in particolare l'assegnamento := è da intendersi in modo analogo a quello dei linguaggi di programmazione, ovvero, la variabile alla sinistra di := indica una provetta in cui si pone il risultato di un'operazione applicata ai contenuti delle provette alla destra dell'assegnamento.
Nonostante si siano individuate moltissime strategie risolutive per SAT [17, 18], di cui alcune di grande interesse teorico, il risultato migliore fino ad ora ottenuto è relativo ad un problema con 20 variabili [4]. Questa difficoltà ha orientato sempre di più le ricerche successive verso altre direzioni.
Negli ultimi anni appare evidente un interesse crescente alla analisi algoritmica di procedimenti di manipolazione del DNA che possano migliorare tecniche biotecnologiche o fornire nuovi metodi in indagini diagnostiche o terapeutiche. Si è compreso che i "calcoli" svolti dalla natura e gli stessi protocolli biotecnologici includono aspetti algoritmici interessanti che si prestano a varie analisi e

Algoritmo di Lipton per la soluzione di problemi del tipo 3-SAT(N, M)

T, T1, T2, T3 provette, L(1, J), L(2, J), L(3, J) termini booleani dell'equazione *J*-esima.

begin

1. *Genera lo spazio delle soluzioni di dimensione N via "mix-and-split" e ponilo in T*;

2. **for** J = 1 **to** M **do**

begin
T1 := extract(T, L(1,J)) ;
T := T - T1 ;
T2 := extract(T , L(2,J)) ;
T := T - T2 ;
T3 := extract(T , L(3,J)) ;
T := merge(T1, T2) ;
T := merge(T, T3)
end;

3. **if** T ≠ ∅, **then** *prendi un clone e sequenzialo, il risultato è una soluzione*
else "*Problema insolubile*"
end.

possono suggerire interessanti applicazioni in campo biomedico. Per esempio, nei lavori [6, 7] si è sviluppata un'analisi combinatoria della PCR scoprendo aspetti matematici piuttosto complessi sulla base dei quali sono stati condotti vari esperimenti biotecnologici ed è emersa una variante della PCR, chiamata XPCR, che sembra adatta allo sviluppo di nuovi metodi nella elaborazione di molecole DNA, sopratutto per l'estrazione selettiva di tali molecole (evitando metodi di affinità chimica) e per la ricombinazione massiva di DNA. Per una comprensione dei fenomeni sottostanti risulta interessante definire una notazione ed un relativo linguaggio algoritmico che permetta di dominare l'enorme ricchezza combinatoria dei processi di elaborazione di strutture bilineari. In questo senso, la prospettiva iniziale del *DNA computing* sembra in un certo senso capovolta, poichè in questo caso è l'algoritmo su forme matematiche che descrive il processo naturale, piuttosto che il processo di trasformazione di molecole a realizzare algoritmi di interesse matematico. Un caso in cui questo punto di vista trova una realizzazione specifica è stato recentemente sviluppato per la comprensione di meccanismi genetici basilari in forme cellulari primitive [9].

7. CONCLUSIONI

Il *DNA Computing* è nato con l'idea di risolvere problemi combinatori sfruttando:
i) la potenza di codifica del DNA,
ii) il parallelismo massiccio con cui si opera su enormi popolazioni di molecole DNA,
iii) il meccanismo associativo di ibridizzazione.
Tuttavia, tale prospettiva iniziale si è ampliata sempre più, mostrando come il rapporto tra stringhe simboliche e molecole di DNA abbia radici profonde, alla base dei meccanismi fondamentali di elaborazione dell'informazione naturale. Nessun sistema può sviluppare comportamenti complessi se non si avvale di procedimetni sofisticati di elaborazione dell'informazione (rappresentazione, strutturazione, trasformazione, memorizzazione e trasmissione). I sistemi naturali necessitano quindi di calcoli, ma spesso tali calcoli obbediscono a logiche diverse da quelli dei sistemi artificiali. Capire e riprodurre tali logiche diventa quindi uno strumento, oltre che di interesse applicativo, di comprensione e conoscenza della natura, uno strumento di vera *philosophia naturalis*, nella sua accezione scientifica più genuina.
Da un punto di vista più specifico, oltre alla ricerca di algoritmi DNA, un'altra linea di ricerca, come già anticipato, è rivolta allo studio di meccanismi di aggregazione di molecole DNA che possano produrre specifiche forme di interesse nel campo delle nanotecnologie. In tale ambito le strutture ottenute per ibridizzazioni a partire da elementi di base possono essere descritte tramite opportuni grafi e quindi, studiando caratteristiche specifiche di tali grafi, si possono dedurre interessanti proprietà delle forme che si vengono a produrre. Una tecnica, nota come *Whirplash PCR*, permette di realizzare in modo naturale automi DNA, in cui semplificando alquanto, lo stato dell'automa è essenzialmente codificato dalla posizione di ibridizzazione di una struttura *hairpin* del tipo indicato nella figura 3 (in basso a sinistra). Gli automi DNA costituiscono una frontiera nelle applicazioni biomediche del DNA computing. Per, esempio, recentemente [3, 23] si sono progettati piccoli automi DNA, battezzati, in modo forse eccessivamente avveniristico, *DNA Doctor*, in grado di riconoscere molecole "nemiche" e quindi di rilasciare antitodi per la loro neutralizzazione.
Per concludere, il *DNA Computing*, dalla soluzione in provetta di problemi combinatori, agli algoritmi biomolecolari, alla analisi computazionale di protocolli biomolecolari e fenomeni genetici, fino agli automi DNA, apre scenari di grande interesse sia teorico che applicativo, mostrando possibilità sperimentali del tutto imprevedibili per l'informatica e nuovi strumenti informatici e matematici al servizio della biologia. Tale campo di ricerca, seppur incrocia temi classici della Bioinformatica (sequenziamenti genomici, e ricerche di similarità in sequenze di biopolimeri), nello stesso tempo si rivolge a temi fondamentali di discipline annunciate ed "in cerca di autore" quali *Cellular Computing*, *Biocomputing* e *Systems Biology* che saranno cruciali per la comprensione dei meccanismi di elaborazione dell'informazione biologica.

Bibliografia

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

[2] Alberts B., Raff M.: *Essential Cell Biology. An introduction to the molecular biology of the cell*. Garland Science, New York, 1997. (Trad. it. Zanichelli, Bologna, 2005).

[3] Benenson K., Paz-Elitzur T., Adar R., Keinan E., Livneh Z., Shapiro E.: Programmable and Autonomous Computing Machine Made of Biomolecules. *Nature*, Vol. 414, 2001.

[4] Braich R.S., Chelyapov N., Johnson C., Rothemund P.W.K., Adleman L.: Solution of a 20-Variable 3-SAT Problem on a DNA Computer. *Science*, Vol. 296, 2002, p. 417-604.

[5] Ciobanu G., Paun G., Perez-Jimenez M.J.: *Applications of Membrane Computing*. Springer, Berlin, 2006.

[6] Franco G., Giagulli C., Laudanna C., Manca V.: *DNA Extraction by Cross Pairing PCR*. In: Ferretti C., Mauri G., Zandron C., et al. (Eds.): 10th International Meeting on DNA Based Computers, LNCS 3384. Springer-Verlag, Berlin, 2005, p. 106–114.

[7] Franco G., Manca V., Giagulli C., Laudanna C.: *DNA Recombination by XPCR*. In: Carbone A., A Pierce N. (Eds.): 11th International Meeting on DNA Computing, LNCS 3892. Springer-Verlag, Berlin, 2006, p. 233-242.

[8] Martyn A.: *Theoretical and Experimental DNA Computation*. Springer-Verlag, 2005.

[9] Ehrenfeucht A., Harju T., Petre I., Prescott D.M., Rozenberg G.: *Computation in Living Cell. Gene assembly in ciliates*, Springer-Verlag, Berlin, 2004.

[10] Garrett R.H., Grisham C.M.: *Biochemistry*. Saunders College Publishing, 1997. (Trad. it. Zanichelli, Bologna, 2002).

[11] Head T.: Formal language theory and DNA: An analysis of the generative capacity of specific recombinant behaviors. *Bulletin of Mathematical Biology*, Vol. 49, 1987, p. 737-759.

[12] Holland J.: Genetic algorithms. *Scientific American*, Vol. 267, n. 1, 1992, p. 66.

[13] Lipton R.J.: DNA solutions of hard computational problems. *Science*, Vol. 268, 1995, p. 542-544.

[14] Manca V.: On the Logic and Geometry of Bilinear Forms. *Fundamenta Informaticae*, Vol. 64, n. 1-4, 2005, p. 261-276.

[15] Manca V., Bianco L., Fontana F.: *Evolutions and oscillations of P systems: Theoretical considerations and applications to biochemical phenomena*. In: Membrane Computing, Mauri G., Paun G., Perez-Jimenez M.J., Rozenberg G.,Salomaa A., (eds), LNCS 3365, Springer, Berlin, 2005, p. 63-84.

[16] Manca V.: *A proof of regularity for finite splicing*. In: Jonoska N., Paun, Rozenberg G. (Eds.), Aspects of molecular computing, LNCS 2950, Springer-Verlag, Berlin, 2004, p. 309-317.

[17] Manca V., Zandron C.: *A Clause string DNA Algorithm for SAT*. In: Jonoska N., Seeman N.C. (Eds.), DNA Computing, LNCS 2340, Springer-Verlag, Berlin, 2003, p. 172-181.

[18] Manca V.: DNA and Membrane Algorithms for SAT. *Fundamentae Informaticae*, Vol. 49, 2002, p. 171-175.

[19] Manca V.: *On the logic of DNA bilinearity*. In: Hagiya M., Ohuchi A. (Eds.): DNA Computing, 8th International Meeting on DNA Based Computers, Preliminary Proceedings, Hokkaido, Japan, 2002, p. 330.

[20] Paun G., Rozenberg G., Salomaa A.: *DNA Computing*. Springer-Verlag, Berlin, 1998.

[21] Rozenberg G., Salomaa A., (Eds.): Handbook of Formal Language Theory, 3 Voll. Springer-Verlag, Berlin,1997.

[22] Seeman N.C.: DNA in a material worldblocking. *Nature*, Vol. 421, 2003, p. 427-431.

[23] Shapiro N., Benenson Y.: Bringing DNA Computers to Life. *Scientific American*, April, 2006, p. 44-51.

[24] Soloveichik D., Winfree E.: *Complexity of Self-assembled Shapes*. LNCS 3384, Springer-Verlag, Berlin, 2005, p. 244-354.

[25] Wolfram S.: *A new kind of science*, Wolfram Media, 2002.

Figure

Le figure 1, 2, 3, 5, 7, 8 sono tratte dagli appunti del corso "Modelli di calcolo non convenzionali" tenuto dall'autore presso l"Università di Verona nell'a.a. 2005/2006.

VINCENZO MANCA è professore ordinario di Informatica presso l'università di Verona. I suoi interessi di ricerca includono i modelli di calcolo non convenzionali (*DNA e Membrane Computing*) e l'analisi informazionale di sistemi biologici. È membro dell'*European Molecular Computing Consortium* ed è stato Visiting Professor ed Invited Speaker in varie istituzioni e conferenze internazionali. È autore di più di 80 pubblicazioni scientifiche internazionali. Ha diretto vari progetti di ricerca, e ha ideato e coordinato esperimenti biotecnologici su algoritmi DNA.
E-mail: vincenzo.manca@univr.it