Capitolo 1

DNA Computing

Il *DNA Computing* è un nuovo modello di calcolo sviluppatosi rapidissimamente a partire dal 1994, dopo un esperimento condotto negli Stati Uniti da Leonard Adleman in cui si dimostrava la possibilità di risolvere problemi combinatori con tecniche di ingegneria genetica.

L'incredibilmente complessa struttura degli esseri viventi è il risultato di poche semplici operazioni (mutazioni locali, tagli, copie, incollamenti, ricombinazioni) applicate a sequenze iniziali di DNA. Si tratta di un processo biologico perfettamente analogo al processo matematico che ottiene $f(w)$, il risultato dell'applicazione di una funzione calcolabile f ad un argomento w, come combinazione di funzioni semplici e basilari, la diversità sta nel modo in cui la natura procede: non aggiungendo e sottraendo, ma tagliando e incollando, con cancellazioni e inserzioni.

Questa osservazione, assieme alla possibilità di sintetizzare il DNA e di manipolarlo in condizioni di laboratorio, ha suggerito l'idea che ogni computazione potesse essere effettuata partendo da sequenze di DNA, che codificano un'informazione iniziale, e applicando ad esse operazioni usuali di ingegneria genetica.

Una cosa importante che ci hanno insegnato i grandi logici degli anni '30 è che il calcolo automatico è una cosa essenzialmente semplice. La macchina di Turing (vedi il paragrafo 2.3) memorizza le informazioni sotto forma di sequenze di lettere su un nastro, e le manipola attraverso semplici operazioni applicate secondo un meccanismo di controllo descritto da una stringa finita. In modo simile, un computer memorizza le informazioni sotto forma di sequenze di zero e uno, e le elabora tramite le operazioni che il chip del processore è in grado di compiere. In generale, per costruire una macchina

universale, cioè capace di calcolare qualsiasi cosa possa essere calcolata, solo due cose sono veramente indispensabili: un metodo per immagazzinare le informazioni e poche semplici operazioni per elaborarle.

La molecola di DNA è uno dei più compatti supporti di informazione, che da miliardi di anni le cellule usano per immagazzinare i "progetti" della vita, e ha una proprietà veramente importante da un punto di vista computazionale: è una sequenza formata da due stringhe "complementari".

Il DNA Computing porta alla formulazione di nuove strutture dati (per esempio, nella macchina di Turing, un doppio nastro con relazioni di complementarietà tra le celle corrispondenti) e nuovi modelli di calcolabilità. D'altra parte le molecole di DNA codificano un noto linguaggio (Twin-Shuffle, vedi Definizione 2.104) che, via trasduzioni (vedi paragrafo 2.6), caratterizza una classe di insiemi computazionalmente universali (vedi Teorema 2.105).

L'idea base del DNA Computing è quella di passare dai microchips alle molecole di DNA. Gli ovvi limiti di miniaturizzazione che si hanno con le attuali tecnologie informatiche spingono alla drastica innovazione che porta le componenti di base del computer a livello molecolare, e due recenti manifestazioni di questa tendenza sono il *Quantum Computing* e appunto il *DNA Computing*.

In realtà già negli anni '50 Richard Feynman[1] speculò sulla possibilità di utilizzare molecole come strumento di calcolo. Con maggiori dettagli, negli anni '80 sia Charles Bennett[2] che Michael Conrad[3] formularono proposte di modelli di calcolo molecolari. Di recente (novembre '94, vedi paragrafo 1.7) è stata sperimentata la reale fattibilità di tali computazioni, resa possibile dai grandi sviluppi della biologia molecolare e dalla capacità di produrre quantità enormi di molecole di DNA di una specifica sequenza e dimensione.

È un'idea rivoluzionaria, che rappresenta un'inversione di tendenza: da circa cinquant'anni si cerca di comprendere la struttura e il comportamento del DNA attraverso modelli computazionali (ricordiamo per esempio lo sviluppo dei processi stocastici e dei metodi statistici per risolvere problemi di genetica ed epidemiologia), invece il *Molecular Computing*, e in particolare il *DNA Computing*, processa variamente le biomolecole, e sfrutta la loro natura polimerica per utilizzarle come unità di memoria.

---

[1] R. P. Feynman, *Minaturization*, D. H. Gilbert ed., Reinhold, 282-296, 1961.

[2] C. H. Bennett, *The Thermodynamics of computation - a review*, Internat, J. Theoret. Physics, **21**, 905-940, 1982.

[3] M. Conrad, *On design principles for a molecular computer*, Communications of the ACM, **28**, 464-480, 1985.

Il fatto che i fenomeni interni alle cellule (copiare tagliare e incollare stringhe di DNA) possano essere computazioni suggerisce che la vita stessa possa consistere in una serie di complesse computazioni. E siccome la vita è uno dei fenomeni naturali più complessi potremmo generalizzare facendo la congettura che l'intero cosmo sia il prodotto di computazioni. La differenza tra le diverse forme di materia potrebbe quindi riflettere solo vari gradi di complessità computazionale: dal caos alla materia inorganica, dall'inorganico all'organico, l'intera evoluzione dell'universo potrebbe essere una storia di sempre crescente complessità di computazione.

1.1 Struttura del DNA

Il fisico-biologo inglese James Watson e il biochimico americano Francis Crick nel 1953 vinsero il Nobel per aver individuato la struttura del DNA; questa scoperta è stata fortemente ispirata dal libro "What is life? The Physical Aspect of the Living Cell", nel quale, per vie del tutto teoriche, Erwin Schrodinger aveva previsto una struttura polimerica per il DNA.

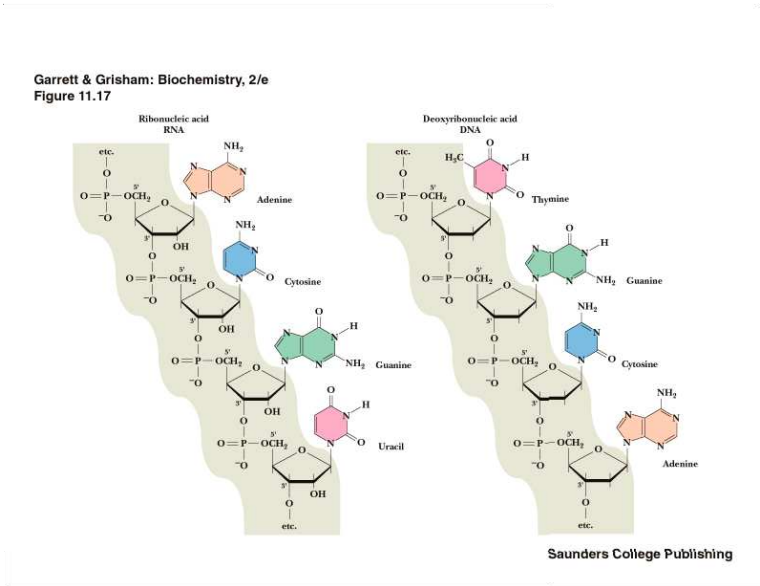

Figura 1.1: Struttura fisica di RNA e DNA (Garrett&Grisham [15])

Il DNA (acido deossiribonucleico) è un *polimero*, formato da una catena di *monomeri* chiamati *deossiribonucleotidi*, che hanno tre componenti: uno *zucchero*, un *gruppo fosforico*, e una *base azotata* (vedi Figura 1.1).

Lo zucchero del deossiribonucleotide si chiama *deossiribosio* (da cui prende il nome anche il DNA) e ha esattamente cinque atomi di carbonio, che numeriamo da $1'$ a $5'$ secondo una nota convenzione. Dentro la struttura dello zucchero troviamo un *gruppo ossidrile* (**OH**) collegato col $3'-$ carbonio, il gruppo fosforico (**P**) legato al $5'-$ carbonio, e la base azotata attaccata al $1'-$ carbonio.

Tra le quattro basi azotate del DNA distinguiamo le due *pirine* dalle due *pirimidine*, rispettivamente *adenina* (**A**) e *guanina* (**G**), *citosina* (**C**) e *timina* (**T**). I nucleotidi si differenziano solo per le loro basi, motivo per cui spesso useremo base come sinonimo di nucleotide.

Nella Figura 1.2, lo zucchero di ogni nucleotide è quell'asta verticale che ha la base azotata in cima, all'altezza del $1'$ - atomo di carbonio, e il $5'$ - atomo di carbonio in basso, dove appunto troviamo il gruppo fosforico (la numerazione degli atomi di carbonio è intesa in ordine progressivo dall'alto verso il basso).

I nucleotidi si possono legare tra loro in due modi diversi:

1. Il $5'$-gruppo fosforico di un nucleotide si lega al $3'$-gruppo ossidrile di un altro nucleotide formando un legame covalente che si chiama *fosfodiesterico* (vedi Figura 1.2).

   Così si viene a formare una sequenza di basi consecutive, che diciamo lunga n se formata da n nucleotidi. Il termine *oligonucleotide* o *oligo* indica stringhe di piccola lunghezza, con n fino a 100. Oggi è possibile scrivere una sequenza di DNA su un foglio di carta, inviarla ad una ditta specializzata e ricevere, in pochi giorni, una provetta contenente circa 10^{18} molecole di DNA, tutte (o quasi) contenenti la sequenza desiderata: in questo modo, oggi si possono maneggiare in modo efficiente sequenze lunghe anche 100 nucleotidi (le molecole vengono spedite liofilizzate e appaiono come una masserella bianca e amorfa sul fondo della provetta).

   Questo tipo di concatenazione dà ad ogni stringa un suo naturale *orientamento*: parliamo di direzione $5' - 3'$ per una stringa come quella della Figura 1.2, in cui a sinistra rimane libero il gruppo fosforico e a destra il gruppo ossidrile, e di direzione $3' - 5'$ per una stringa ribaltata rispetto a quella della figura, che quindi sulla sinistra ha libero il gruppo

ossidrile e sulla destra il gruppo fosforico. Quando non specificato, si intende che una stringa ha direzione $5' - 3'$, secondo una convenzione usuale in biologia molecolare.

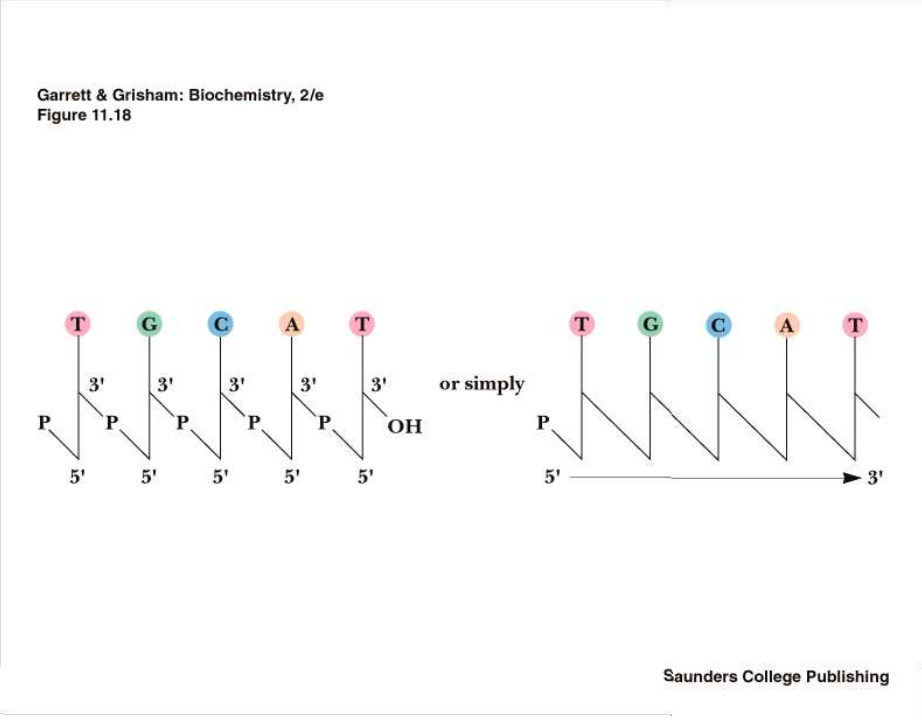

Figura 1.2: Struttura dei nucleotidi

2. La base di un nucleotide interagisce con la base di un altro nucleotide tramite legame chimici molto deboli, che si chiamano *legami idrogeno*. Questo tipo di legame avviene esclusivamente tra coppie di basi *complementari*, ovvero tra A e T, oppure tra C e G, e tale principio di appaiamento si chiama *complementarietà di Watson-Crick*. Tra la A e la T si vengono a formare due legami idrogeno, mentre la C e la G si legano più solidamente con tre legami idrogeno: questa differenza ha un grande peso nella formazione del filamento di DNA. In realtà i legami idrogeno sono troppo deboli per tenere assieme due singoli nucleotidi,

ma il loro effetto cumulativo su diverse coppie di basi complementari in una molecola di DNA ne fa un legame stabile. Avviene un'interazione di tipo cooperativo, nel senso che è debole l'energia di legame coinvolta in ciascuno dei punti di contatto, ma l'interazione complessiva è stabilizzata dal simultaneo verificarsi del sistema di legami deboli.

Si viene a formare la classica doppia elica del DNA quando si uniscono due sequenze di nucleotidi e il risultante doppio filamento si avvolge attorno ad un asse (vedi Figura 1.3).

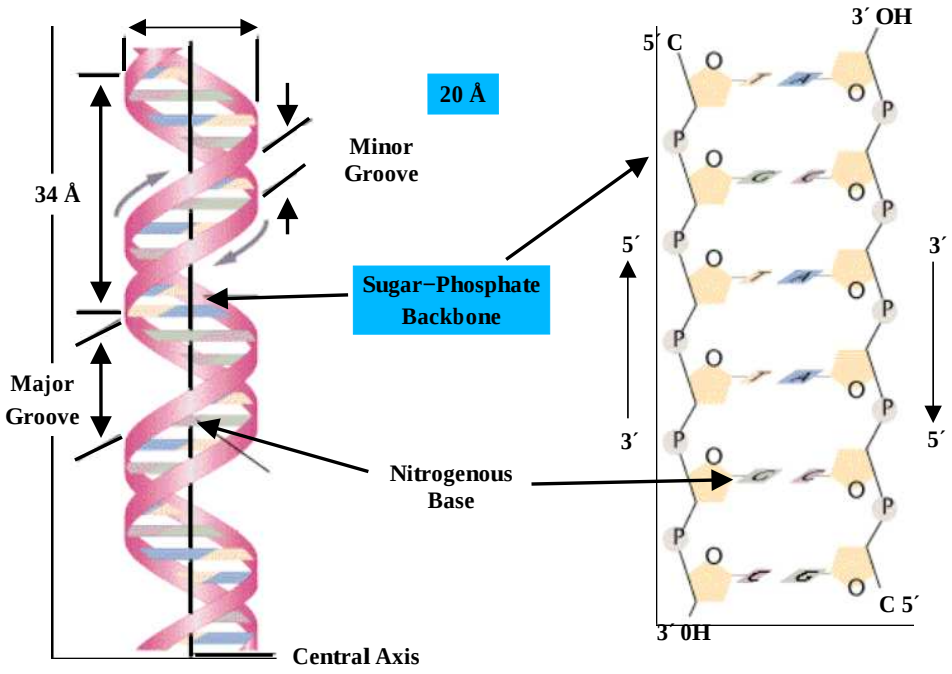

Figura 1.3: Doppia elica del DNA

Il processo che unisce due singole sequenze di DNA va sotto il nome di *annealing* ed è basilare nell'implementazione degli algoritmi del DNA Computing.

Un aspetto molto importante è che una sequenza α si unisce alla sua complementare (ovvero a quella avente le rispettive basi complementari) se e solo se questa ha direzione (detta anche *polarità*) opposta; per esempio, se $\alpha = \text{ATTCGAG}$ allora si appaia con $\overline{\alpha} = \text{CTCGAAT}$. Se ne deduce che un filamento della doppia elica si snoda da $5'$ a $3'$ e l'altro da $3'$ a $5'$.

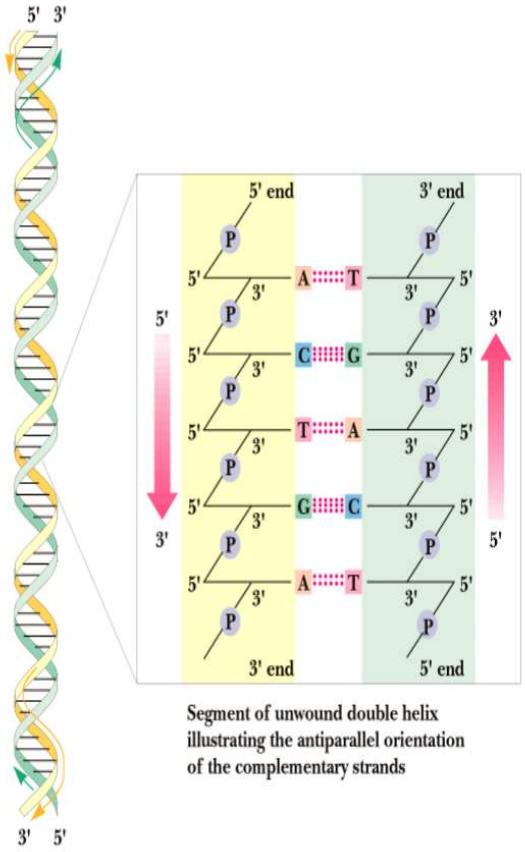

Figura 1.4: Orientazione antiparallela delle stringhe complementari

In conclusione abbiamo tre aspetti essenziali nella struttura del DNA:

1. La *bilinearità*, necessaria per una riproduzione efficiente della molecola.

2. La *complementarietà*, che fa avvenire l'annealing e, come vedremo (per esempio nell'esperimento di Adleman, par 1.7), realizza efficientemente una selettività concatenativa tra stringhe.

3. L'*antiparallelismo*, che favorisce la struttura simmetrica della molecola e le reazioni enzimatiche di importanza fondamentale per la vita.

1.2 Vantaggi del DNA Computing

Andiamo dunque a vedere, in concreto, i vantaggi nel "fare calcoli" col DNA.

- **Velocità di calcolo**.

  Secondo Adleman [2] un ipotetico computer DNA esegue approssimativamente $1,2 \times 10^{18}$ operazioni al secondo, quindi è circa 1.200.000 volte più veloce del più veloce degli attuali supercomputers.
  Un calcolo per il quale un computer richiede cento mila anni il DNA lo fa in un mese! Il segreto sta in un notevole **parallelismo**. Il DNA è una molecola stabile, che può essere trattata in molti modi già ben noti in laboratorio e può essere duplicata esponenzialmente, questo rende possibili interessanti computazioni con un enorme parallelismo, molto superiore a quello fornito dagli attuali computers.

Nel risolvere un problema di cammini hamiltoniani, Adleman ha condotto un esperimento in un volume pari alla cinquantesima parte di un cucchiaino, e il sistema ha permesso di concatenare circa 10^{14} archi di un grafo in un solo secondo! È chiaro che tutto questo acquista grande rilevanza per problemi per loro natura decomponibili in sottoproblemi, che richiedono operazioni elementari e tempi di risoluzione tra loro confrontabili (per esempio la fattorizzazione di un intero).

La maggior parte dei famosi problemi computazionalmente *intrattabili* (quelli che con i modelli di computazione convenzionali sono risolvibili *deterministicamente* in un tempo non polinomiale rispetto alla dimensione del problema, vedi Definizione 2.127) si risolvono con la ricerca esaustiva di tutte le soluzioni, e l'insormontabile difficoltà sta nel fatto che tale ricerca è troppo vasta per l'attuale tecnologia. Invece la densità di informazioni che abbiamo nelle stringhe di DNA, di cui riusciamo con facilità a costruirne molte copie,

e soprattutto l'enorme capacità di calcolo parallelo di tali stringhe rende possibile una ricerca esaustiva delle soluzioni in tempi lineari con la dimensione del problema (contro i tempi esponenziali del calcolo sequenziale).

- **Compattezza di informazione.**

  La compattezza del DNA come contenitore di dati ha dell'incredibile. Una mole di DNA contiene 6.02×10^{23} (che è il numero di Avogadro) monomeri (basi di DNA) e il peso medio di una coppia di basi è approssimativamente 650 dalton, pertanto 1 grammo di DNA contiene circa 10^{21} bp (1 bp = una coppia di basi complementari).

  Siccome ci sono 4 basi di DNA, ogni base codifica 2 bits (perché per poterle distinguere con liste di zero e uno ho bisogno di due cifre), ma possiamo anche dire che una coppia di basi complementari codifica 2 bits, perché, avendo una delle due basi, la conoscenza della complementare non aggiunge alcuna informazione.

  Da questi presupposti segue che 1 grammo di DNA contiene approssimativamente 2×10^{21} bits, mentre con le tecnologie convenzionali si possono immagazzinare al più 10^9 bits per grammo, quindi il DNA potenzialmente ha una capacità di contenimento dati 2×10^{12} volte superiore a quella dell'attuale tecnologia. Infatti, possiamo calcolare che le molecole di DNA hanno una densità d'informazione di circa 2 bits per nanometro cubo, contro 1 bit per 10^{12}nm^3 dell'attuale tecnologia. Praticamente, un solo grammo di DNA, che secco occuperebbe un volume di un centimetro cubo, può immagazzinare le informazioni contenute in circa cinquanta miliardi di CD! Infatti, assumendo che i dischetti più grossi in commercio contengano circa 5Gb (cioè 5×10^9 bytes, che corrisponde a $5 \times 8 \times 10^9$ bits) e quindi l'informazione di 2×10^{10} bp, si ha che:

$$1 \text{ grammo DNA} = \frac{10^{21}}{2 \times 10^{10}} = 50.000.000.000 \text{ dischetti!}$$

  È stato stimato che pochi grammi di DNA possono contenere più informazione di quanta ne contengano al momento tutte le memorie dei computers sulla terra.

Nei lavori [2, 3, 10, 50] gli autori, per ogni grammo, calcolano una compattezza d'informazione doppia rispetto a quella appena vista, in quanto

considerano singole stringhe di DNA, quindi nel calcolo tutte le basi presenti vanno a codificare esattamente 2 bits. Questo loro approccio non è inverosimile, in quanto oggi esistono delle tecniche di cristallizzazione che impediscono l'appaiamento delle stringhe complementari anche a temperatura ambiente, tuttavia abbiamo preferito fare un calcolo ridimensionato, che consideri la reale informazione contenuta in una molecola di DNA senza alcun intervento esterno. Le cifre ottenute sono comunque molto interessanti: anche se riuscissimo ad utilizzare solo il 10% di tale contenuto informativo otterremmo comunque un'enorme avanzamento rispetto alla tecnologia attuale. D'altra parte, l'essere come siamo dipende dall'informazione contenuta nei geni del nostro DNA, che occupano solo il 10% della molecola (detta *parte codificante*).

Ricerche nel campo del DNA Computing stanno indagando sulla possibilità di codificare, accumulare, manipolare e recuperare informazioni biologiche in sequenze di DNA. Per esempio [50] fornisce i dettagli di uno studio che ha coinvolto la progettazione, la costruzione e la sperimentazione di una grande base-dati per il contenimento e il ripristino di informazioni dentro sequenze di basi nucleotidiche di molecole artificiali di DNA. Queste basi di dati consistono in un grande insieme di singole stringhe di DNA (libere in soluzione o immobilizzate su sferette, lastre vetrose, o chips) che contengono una particolare sequenza di basi, consistente in sottostringhe di cui ciascuna fa parte di un predeterminato insieme. L'esperimento prevede un insieme di sferette metalliche, di cui ciascuna ha attaccate molte copie di una certa sequenza. Per recuperare un dato in tale "libreria", viene utilizzata una tecnica che si chiama FACS (Fluorescence Activated Cell Sorting), che sostanzialmente separa le sferette che contengono il dato dalle altre (tramite stringhe fluorescenti complementari a quelle che codificano il dato), seguita dall'amplificazione del dato per favorirne il recupero.

- **Risparmio energetico**.

  Potenzialmente, i computers molecolari hanno un'efficienza energetica straordinaria. In linea di massima, un joule è sufficiente per compiere circa 2×10^{19} reazioni di ligasi (vedi paragrafo 1.5). Una cosa tanto più notevole se pensiamo che la seconda legge della termodinamica prevede che si possano compiere al massimo 34×10^{19} operazioni irreversibili per joule, a temperatura ambiente. Gli esistenti supercomputers eseguono 10^9 operazioni per joule, quindi l'efficienza energetica di un ipotetico

computer DNA sarebbe 10^{10} volte maggiore (perché verrebbero eseguite 2×10^{19} operazioni per joule).

1.3 Le 3 fasi degli algoritmi DNA

Quando i legami tra due stringhe di DNA hanno luogo (sotto condizioni ideali) sappiamo che le basi opposte sono complementari, quindi basta conoscerne una per avere l'altra, perché la complementarietà di Watson-Crick da un singolo filamento di DNA ci permette di ottenerne uno doppio. Questo fenomeno è cruciale nelle ultime due fasi del seguente schema generale di un algoritmo DNA.
Dato un problema, la tipica metodologia del DNA Computing si avvale di tre fasi:

1. **Codifica**

   Consiste nel tradurre, con una funzione opportuna, l'istanza del problema in un multinsieme (per la definizione vedi paragrafo 1.5) di stringhe DNA.

2. **Operazioni molecolari**

   Si tratta di operazioni in provetta che processano i dati in dimensione molecolare. Questa fase consiste nel favorire le condizioni che permettono alle molecole di DNA di reagire tra di loro, in modo da ottenere una provetta con stringhe codificanti un insieme di potenziali soluzioni del problema (generalmente ottenute per "annealing" e "ligazione" dalle stringhe iniziali codificanti i dati, vedi paragrafo 1.5 punto 2).

3. **Estrazione del risultato**

   Si usano dei protocolli per estrarre il risultato (in forma molecolare): nella miscela ottenuta nella fase precedente si separano le soluzioni del problema ("stringhe buone") dalle non soluzioni ("stringhe non buone"), e si estraggono.

Sono stati proposti algoritmi DNA per l'espansione dei determinanti simbolici[4], per la connessione dei grafi e il problema dello zaino usando la pro-

[4] T. Leete, M. Schwartz, R. Williams, D. Wood, J. Salem, H. Rubin, *Massively parallel DNA computation: expansion of symbolic determinants*, 2th DIMACS workshop on DNA based computers, Princeton, 49-66, 1996.

grammazione dinamica[5], per il problema della colorazione delle strade[6], per la moltiplicazione di matrici[7] e per l'addizione[8].

In passato ci sono stati anche "programmi molecolari" per la rottura del DES (Data Encryption Standard del governo degli Stati Uniti). Il DES cifra messaggi di 64 bits e usa chiavi di 56 bits; rompere il DES significa che data una coppia (testo scritto, testo cifrato), possiamo trovare una chiave che traduce il testo scritto in quello cifrato. Affrontare il problema con i metodi convenzionali richiede una ricerca esaustiva attraverso tutte le 2^{56} chiavi DES, che con un tempo medio di 100.000 operazioni al secondo comporta 10.000 anni di lavoro (oggi abbiamo supercomputer specializzati che invece riescono ad esaminare l'intero spazio delle chiavi in tempi ragionevoli, evidentemente rispetto a vent'anni fa gli attuali sistemi crittografici devono necessariamente avere chiavi più lunghe!). Invece si è stimato che, usando la computazione molecolare, il DES potrebbe essere rotto in circa 4 mesi di lavoro di laboratorio e approssimativamente basterebbe un grammo di DNA.

Nei capitoli successivi verranno proposti, con maggiori dettagli, algoritmi DNA che risolvono istanze di SAT (vedi Definizione 2.132).

1.4 Sequenze di DNA come stringhe

Il modello prevalente in biologia molecolare è legato ad un'analogia tra la sequenza del DNA e un testo scritto in un alfabeto a 4 lettere, e questa metafora linguistica è stata estremamente fruttuosa, come testimoniano termini quali "trascrizione", "traduzione", "codice genetico", "messaggio genetico" etc.

Sul finire degli anni '60, con la decifrazione di un codice universale di corrispondenza tra aminoacidi e triplette di nucleotidi, e con la delucidazione di sistemi di regolazione comune per gruppi di geni ad azione coordinata (operoni), l'organizzazione del genoma appariva comprensibile e logica. Il DNA veniva visto come un nastro-programma contenente istruzioni, sorta di "progetto" dell'organismo, "scritto" in termini intelligibili sia a livello lessicale

[5]E. Baum, D. Boneh, *Running dynamic programming algorithms on a DNA computer*, 2th DIMACS workshop on DNA based computers, Princeton, 141-147, 1996.

[6]N. Jonoska, S. Karl, *A molecular computation of the road coloring problem*, 2th DIMACS workshop on DNA based computers, Princeton, 148-158, 1996.

[7]J. Oliver, *Computation with DNA: matrix multiplication*, 2th DIMACS workshop on DNA based computers, Princeton, 236-248, 1996.

[8]F. Guarnieri, C. Bancroft, *Use of a horizontal chain reaction for DNA-based addition*, 2th DIMACS workshop on DNA based computers, Princeton, 249-259, 1996.

che grammaticale. Sono stati costruiti algoritmi per la ricerca automatizzata di "parole significative", basati sulla determinazione della frequenza con cui un dato oligonucleotide si presenta in una sequenza data, e sulla significatività o meno, dal punto di vista statistico, dello scostamento di tale frequenza da quella prevista nel caso di una distribuzione puramente casuale degli stessi nucleotidi.

A sconvolgere questo quadro ottimistico (basato soprattutto sullo studio dei meno evoluti genomi procariotici) sono venute negli anni '70 nuove conoscenze, rese possibili dalle nuove tecniche di clonaggio molecolare e dalla loro applicazione allo studio dei più evoluti genomi eucariotici. La stessa inattesa scoperta della struttura discontinua dei geni (e della fedele trascrizione e successiva escissione degli introni[9] nell'RNA) pone problemi nei confronti dell'idea di un programma in chiara corrispondenza con le funzioni codificate. A ciò si aggiungono le nuove acquisizioni circa la possibilità di trasposizioni e riarrangiamenti genomici, che rendono impossibile considerare il DNA alla stregua di un deposito statico di informazioni, e mostrano come esso stesso sia sottoposto a continua dinamica.

È noto da diversi anni infatti, che la molecola di DNA corrisponde alla struttura individuata da Watson e Crick solo in media, mentre se ne discosta localmente in dipendenza dal tipo di sequenza. La doppia elica può infatti assumere localmente conformazioni diverse (comportanti, ad esempio, il restringimento di uno dei due solchi, o la presenza di curvature dell'asse, maggiore o minore flessibilità etc.), suscettibili di essere riconosciute con maggiore o minore affinità dai diversi fattori proteici che sappiamo essere implicati nei processi di regolazione.

Oggi sappiamo che la struttura dei genomi attuali è collegata attraverso molteplici eventi, puntiformi o estesi (mutazioni puntiformi, ma anche delezioni, inversioni, amplificazioni), ai genomi degli organismi che li hanno preceduti nel corso dell'evoluzione. Questi vincoli influenzano la stessa composizione media in nucleotidi, che nelle elaborazioni statistiche non può essere trattata come una variabile inessenziale rispetto alla quale normalizzare.

Sappiamo che, nelle regioni codificanti per proteine, le "parole" trovate scandendo linearmente la sequenza sono in corrispondenza biunivoca con il loro significato funzionale, così che un ipotetico vocabolario ne permetterebbe

[9] La codifica delle proteine negli eucarioti comincia nel nucleo delle cellule con la trascrizione di una sequenza di DNA su una di RNA: a partire dalla generica sequenza $a_1 b_1 a_2 b_2 \cdots a_n b_n$ alcune sottostringhe $b_1, \ldots, b_n$, dette *introni*, vengono escisse ed eliminate, le altre $a_1, \ldots, a_n$, dette *esoni*, vengono trascritte.

la decodificazione senza ambiguità, ma non vi è alcuna evidenza che ciò sia vero al di fuori di tali regioni. Anzi, in generale, uno stesso tratto di DNA può assumere significati funzionali diversi (e a volte anche opposti, per esempio attivazione o blocco della trascrizione) in dipendenza dal contesto non solo nelle immediate vicinanze, ma anche a distanza, infatti la molecola di DNA è in grado di ripiegarsi nello spazio in modo da realizzare punti di interazione tra siti linearmente molto distanti.

Possiamo dire che la molecola di DNA appare certamente lineare nel suo processo di replicazione, ma in generale ha una "densità lineare d'informazione" relativamente bassa [14].

Da tutto ciò se ne deduce che il modello consistente in un linguaggio, inteso come multinsieme di stringhe, su un alfabeto di quattro lettere, non è valido per uno studio che riguarda la struttura del DNA e le leggi che ne organizzano il contenuto informativo.

Tuttavia tale modello risulta perfettamente calzante per gli scopi del DNA Computing, in quanto i principali fattori che lo invalidano sono proprio quelli che negli esperimenti si cerca di ridurre al minimo: le stringhe di DNA con cui "si calcola" vengono *linearizzate* il più possibile.

Infatti, uno degli accorgimenti più importanti di un algoritmo DNA è la minimizzazione delle "strutture secondarie", ovvero dei ripiegamenti nello spazio che le stringhe di DNA in natura tendono ad avere (per favorire i fenomeni di catalisi le molecole tipicamente si *deformano*, in modo da instaurare legami solo con molecole "compatibili", cioè aventi una forma che combacia perfettamente alla loro). Per esempio nell'esperimento descritto in [50], vengono sintetizzate stringhe aventi solo A, C e T, perchè si è notato che la presenza di G provoca un notevole aumento della struttura secondaria.

Per poter codificare le sequenze di DNA come stringhe di un linguaggio servono 4 simboli, in quanto 2 sono necessari per formare efficientemente le parole di un linguaggio e gli altri due seguono dalla complementarietà (non possono essere gli stessi due perché il legame tra la C e la G è più forte, e in particolare è diverso, da quello tra la T e la A).

Pertanto, se consideriamo le stringhe di DNA prive di polarità, è naturale metterle in relazione con i linguaggi formali (vedi capitolo 2).

Più precisamente, viene considerato un alfabeto $X = \{A, C, G, T\}$, su cui (C, G) e (A, T) si dicono coppie di simboli *complementari*, e su $X^\star$ (vedi Definizione 2.3) si definisce una *polarità* nel modo seguente: la stessa sequenza assume due significati diversi in relazione al fatto che abbia la direzione $5' - 3'$ oppure $3' - 5'$. Due singole stringhe complementari con orientazione opposta

si congiungono per formare la *doppia elica* in un processo chiamato *annealing*, il processo inverso si chiama *melting*.

Dobbiamo comunque tener conto delle principali differenze tra le stringhe di $X^\star$ e quelle di DNA in provetta:

SEQUENZE DI DNA	STRINGHE
mobili	fisse
illeggibili	leggibili
doppie	singole
deformabili	non deformabili
Multinsiemi	Linguaggi

Andiamo a vedere come alcune di queste differenze siano superabili grazie alla biotecnologia.

1.5 Operazioni sulle stringhe

In questa sezione descriviamo le operazioni con cui le stringhe di DNA vengono "processate" in laboratorio. Nella nostra schematizzazione si tratta di operazioni definite su un *multinsieme* di stringhe sull'afabeto $X = \{A, C, G, T\}$.

Un **multinsieme** è un *insieme quantificato*, ovvero un insieme del tipo $M = \{3a, 2b, 7c, \dots\}$, in cui di ogni elemento si conosce il numero di copie.

Un multinsieme $M = \{3a, 2b, 7c, \dots\}$ si può vedere anche come una funzione f_M, che ad ogni elemento associa il relativo numero di copie (nel caso riportato $f_M(a) = 3, f_M(b) = 2, f_M(c) = 7, \dots$), oppure come una stringa commutativa, dove la stringa è una concatenazione degli elementi di cui appunto non conta la disposizione.

Le bio-operazioni che seguono, e altre ancora, vengono usate per scrivere *procedure*, che ricevono come input una provetta contenente stringhe di DNA e restituiscono come output una risposta affermativa o negativa, o un'insieme di provette. Una procedura parte con il multinsieme delle stringhe che rappresentano i dati del problema secondo qualche codifica. Una computazione consiste in una sequenza di provette contenenti stringhe di DNA [24]. Ciascuna operazione elementare assume un insieme iniziale di oligonucleotidi.

1. **Unione** (o *Merge*)

   Consiste nel versare i contenuti di più provette in una singola provetta.

   Formalmente, dati T_1 e T_2 multinsiemi, con questa operazione si ottiene

   $$\cup(T_1, T_2) = T_1 \cup T_2$$

   dove $\cup$ denota l'unione di multinsiemi ed è ovviamente associativa.

   Le molecole di DNA sono fragili e vanno maneggiate con delicatezza, nel riversare o miscelare il contenuto delle provette si rischia che le forze vincolari possano frammentare le stringhe. In questa operazione bisogna anche stare attenti alla quantità di DNA che può rimanere nell'uno o l'altro contenitore, e può far perdere informazioni importanti per i nostri scopi.

2. **Denaturazione/Rinaturazione/Ibridizzazione/Ligazione**

   I legami idrogeno tra le basi complementari sono molto più deboli dei legami fosfodiesterici tra i nucleotidi consecutivi dentro una stringa, e questo ci permette di separare due stringhe di DNA senza spezzare le singole parti. La tecnica standard consiste nel portare la temperatura della provetta a $85° - 95°$ C, in modo da fornire l'energia necessaria per spezzare i legami idrogeno, e il processo che fa separare i due filamenti si chiama *denaturazione* o *melting*.

   Il processo inverso si chiama *rinaturazione*, basta riabbassare la temperatura a $30° - 60°$ C, e aspettare che le basi complementari si ritrovino in soluzione, per vedere una nuova formazione della doppia elica (vedi Figura 1.5). Questo processo si chiama anche *annealing*.

   L'*ibridizzazione* degli acidi nucleici rende possibile trovare una specifica sequenza di DNA o RNA nell'intero filamento. Sfrutta la complementarietà di Watson-Crick dei nucleotidi, unendo singole o parzialmente doppie stringhe di DNA per formare molecole di doppie stringhe.

   In letteratura c'è un abuso di linguaggio nel considerare annealing e ibridizzazione come sinonimi: la differenza è che nel primo caso le due stringhe si appaiano in tutta la loro lunghezza, nel secondo solo in una parte.

   La *ligazione* si esegue assieme all'ibridizzazione per chiudere le "nicchie" che sono rimaste dopo che si sono congiunte due o più stringhe

a formare una molecola: la sua accuratezza è molto alta sebbene non perfetta. L'operazione consiste nel far agire l'enzima **ligasi**, che realizza il legame fosfodiesterico tra gruppo ossidrile e gruppo fosforico delle stringhe concatenate dopo l'annealing.

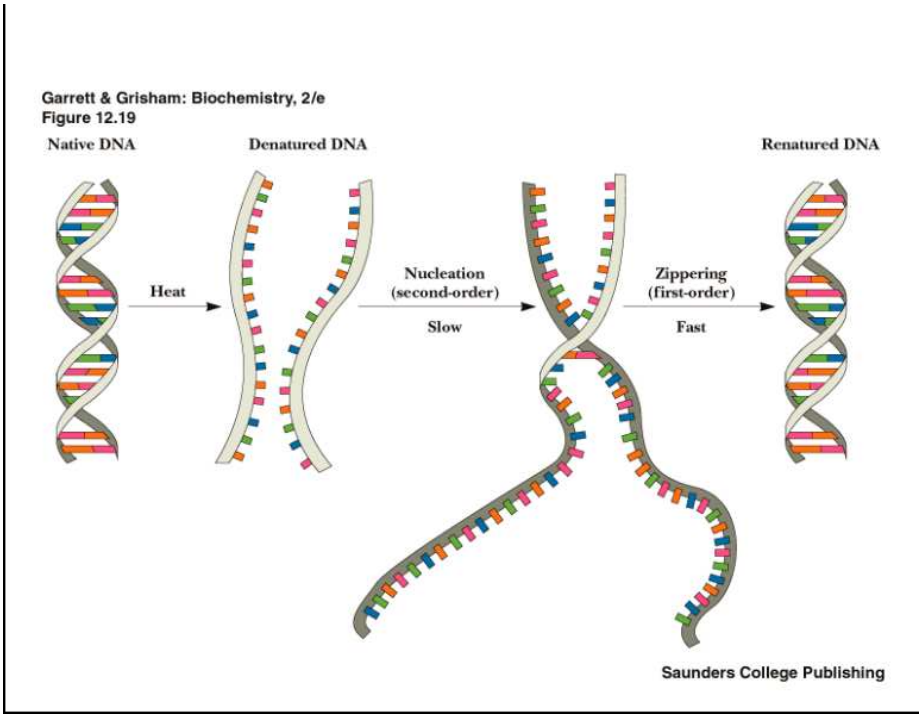

Figura 1.5: Denaturazione e Rinaturazione

Nelle cellule, la DNA-ligasi ha il compito di riparare le interruzioni che si creano nella molecola di DNA, come succede per esempio nelle cellule della pelle, dove il DNA subisce rotture accidentali causate dalla radiazione ultravioletta. Nelle doppie stringhe di DNA, se una delle singole stringhe contiene una discontinuità (ovvero un nucleotide non è legato a quello vicino) allora questo può essere "riparato" dalla DNA-ligasi, che

crea una stringa unificata grazie ai legami con le parti complementari (vedi Figura 1.6).

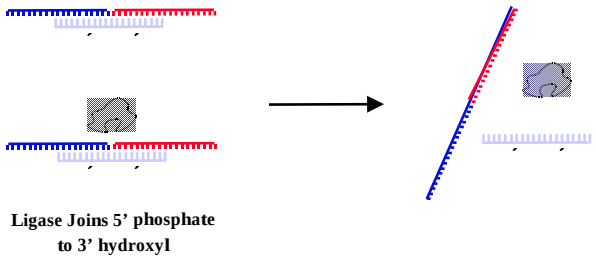

Figura 1.6: La DNA-ligasi

Le operazioni denaturazione, rinaturazione, ibridizzazione e ligazione sono basilari e frequenti per gli algoritmi di DNA Computing, spesso però sono soggette ad errori, in quanto le singole stringhe messe in soluzione, nelle giuste condizioni per l'annealing, possono avere un comportamento che impedisce l'annealing desiderato, e non solo per via della flessibilità delle molecole (o struttura secondaria, vedi paragrafo 1.4): per esempio le stringhe possono ripiegarsi su se stesse, in un modo tale che l'annealing avviene tra porzioni complementari ma appartenenti alla stessa stringa, e questo fenomeno, che produce le *strutture hairpin*, non solo genera un tipo di annealing indesiderato ma chiaramente impedisce quello sperato.

L'articolo [18] suggerisce di manipolare doppie stringhe che abbiano ai lati delle sporgenze (basta prendere il prodotto di un opportuno enzi-

ma su una doppia stringa) chiamate *sticky ends*, favorendo l'annealing tra queste sporgenze. Ma anche in questo caso si possono incontrare dei problemi: nell'esperimento [37] per esempio, le strutture a doppia stringa, che si sarebbero dovute attaccare l'un l'altra in un certo modo, inizialmente hanno portato anche alla formazione di strutture indesiderate, a causa della palindromicità degli enzimi che hanno tagliato gli sticky ends. Questo problema è stato poi risolto con una procedura chiamata *defosforilazione* (vedi punto 8).

L'annealing deve essere condotta con molta attenzione perché i parametri termodinamici dipendono dalla sequenza stessa, e il fatto che si appaino male le basi o si vengano a formare strutture indesiderate dipende da diversi fattori: concentrazione salina, durata della reazione, temperatura, relativa percentuale di nucleotidi etc. Un accorgimento per ridurre questo tipo di errori è quello di intercalare le parti che codificano i bits informativi nelle stringhe di DNA con delle sequenze random. Dentro una lunga regione di complementarietà possiamo anche trovare delle basi appaiate male senza che questo alteri la struttura della doppia stringa, ma bisogna tenerne conto per calcolare la temperatura di melting T_m (che in tal caso è più bassa). Sono state proposte, in letteratura, diverse formule empiriche per le temperature di denaturazione (per differenti lunghezze di segmenti di DNA, differenti strutture e basi dispaiate). La denaturazione talvolta viene facilitata dalla presenza in soluzione di alcune sostanze chimiche, che possono abbassare notevolmente la temperatura T_m.

Negli esperimenti la fase che produce tutte le soluzioni candidate del problema spesso consiste nell'operazione "annealing+ligazione", che ha un'efficienza del $43 \pm 4\%$ [60].

3. **Restrizione/Frammentazione**

   Questa operazione consiste nel tagliare il DNA nei punti in cui ci sono delle precise sottosequenze, e facilita enormemente l'isolazione e la manipolazione del DNA individuale.

   Gli *enzimi* sono proteine che catalizzano le reazioni chimiche nelle cellule viventi. La natura fornisce degli enzimi, detti di *restrizione*, come "bisturi molecolari" aventi un'attività ottimale; la loro efficienza è del 100%, in pratica possiamo essere sicuri che in presenza di un enzima tutte le stringhe aventi come sottostringhe una certa sequenza vengono

tagliate senza che le altre stringhe vengano minimamente alterate. In particolare, si chiamano *nucleasi* gli enzimi che tagliano le molecole di acidi nucleici.

Le **endonucleasi** di restrizione "perlustrano" il DNA alla ricerca di specifiche sequenze di basi, lunghe 4, 6 o poco più, e quando incontrano una di queste sequenze tagliano la molecola in due parti, distruggendo il legame fosfodiesterico all'interno. Le endonucleasi sono molto specializzate su cosa, dove e come tagliare: una doppia stringa che contiene il *sito di restrizione* dentro la sua sequenza viene tagliata dall'enzima in quel punto, il tipo di taglio dipende dall'enzima (vedi Figura 1.7).

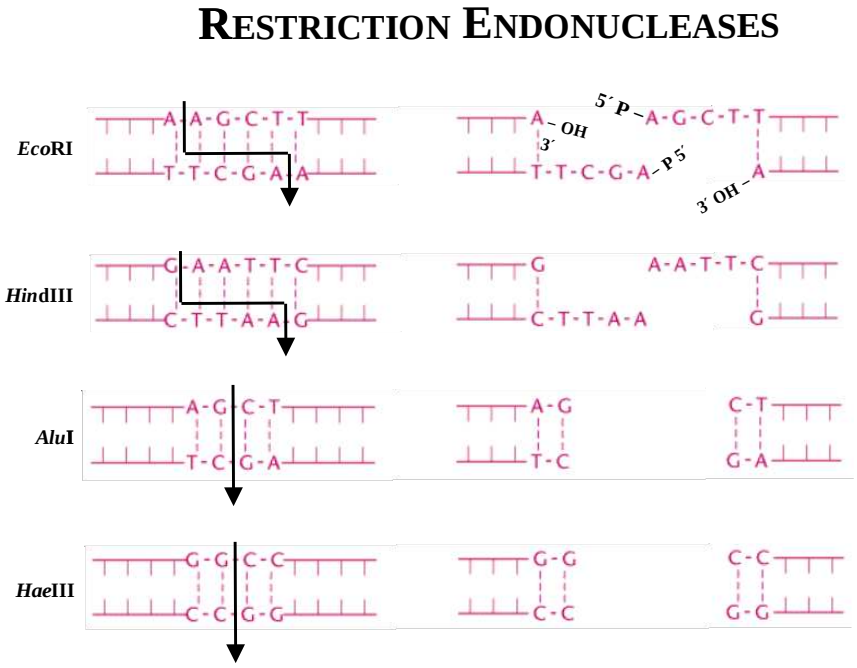

Figura 1.7: Endonucleasi

Per esempio, EcoRI (che prende il nome dal batterio Escherichia Coli) è un enzima di restrizione che taglia il DNA immediatamente dopo la base G della sequenza 5′ -GAATTC (vedi Figura 1.7). La sua azione è

altamente specifica, ed è rarissimo che tagli il DNA in sequenze diverse da questa. La teoria più accreditata è che le endonucleasi di restrizione si siano evolute per proteggere i batteri dai virus. Per esempio, il batterio Escherichia Coli è in grado, attraverso un processo chiamato *metilazione* (vedi punto 8) di proteggere il proprio DNA dall'azione di EcoRI, mentre il DNA di un virus invasore che contenga la sequenza GAATTC viene tagliato in due.

Le **exonucleasi** frammentano il DNA tagliando i nucleotidi uno per uno alle estremità della molecola. Alcune exonucleasi rimuovono i nucleotidi dalla parte $5'$ della molecola, altre dalla parte $3'$, alcune exonucleasi agiscono sulle singole stringhe, altre sulle doppie.

Per gli algoritmi del DNA Computing abbiamo a disposizione alcune centinaia di enzimi diversi, di cui conosciamo sia il comportamento che la temperatura a cui sono attivi.

4. **Gel-elettroforesi**

L'elettroforesi è uno dei pochi sistemi che non abbiamo "rubato" alle cellule: si tratta di una tecnica di routine che i biologi molecolari usano da più di venti anni e che serve per conoscere la lunghezza (i.e. il numero di nucleotidi consecutivi) delle stringhe. Si basa sul fatto che le molecole di DNA sono caricate negativamente, e quindi, poste in un campo elettrico, tendono a migrare verso l'elettrodo positivo. Essendo la carica proporzionale alla lunghezza della stringa, anche la forza elettrica che le sposta risulta proporzionale a tale lunghezza, e quindi frammenti di DNA di lunghezza diversa viaggiano con la stessa velocità verso il polo positivo. Per questo motivo vengono fatte muovere in una lastra di gel, in modo che l'attrito sia maggiore per le stringhe più lunghe che quindi si muovono più lentamente di quelle più corte.
Si versa una soluzione calda di gel in un contenitore rettangolare di plastica o vetro e si fa raffreddare; durante questa fase si inserisce un "pettine" lungo un lato del contenitore in modo che, rimuovendo il pettine dopo il raffreddamento, si creino dei "pozzetti" nel gel. In tali pozzetti si riversano quantità di strighe di DNA di lunghezza diversa e si attiva un campo elettrico che va dal lato vicino ai pozzetti a quello opposto, si aspetta che le stringhe più corte arrivino dall'altra parte e si disattiva il campo elettrico (vedi Figura 1.8).

A questo punto, (avendo utilizzato particolari reagenti) illuminando il

gel con una luce ultravioletta è possibile vedere l'insieme delle molecole di DNA che hanno la stessa lunghezza: appaiono su una pellicola come bande ad una certa distanza dal pozzetto. Da questa distanza si può risalire alla lunghezza; o più semplicemente, di solito in uno dei pozzetti si mettono delle stringhe di lunghezza nota, in modo che la loro disposizione finale nel gel faccia da marcatore per le altre stringhe, quelle di cui si vuole misure la lunghezza.

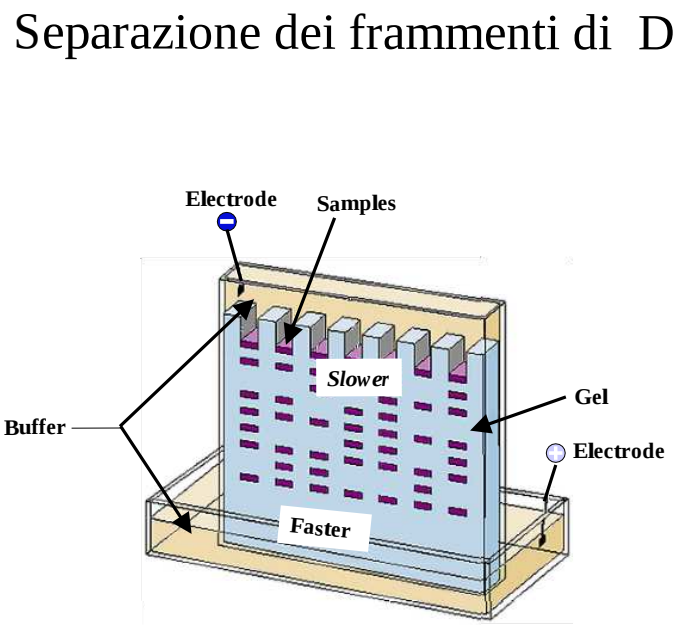

Figura 1.8: Gel-elettroforesi

La scelta del tipo di gel è cruciale per l'efficienza dell'operazione. Di solito si usano l'*agarosio* e il *polyacrylammide*: il primo per misurare grossi frammenti (lunghi più di 500), il secondo, che ha una maggiore porosità e quindi è più efficiente, è indicato soprattutto per misurare

piccoli frammenti. È consigliato l'uso di un gel "denaturante", che cioè si accorga di quando oltre alle singole stringhe sono presenti strutture più complesse di DNA, la cui mobilità nel gel non dipende solo dalla lunghezza (come richiede l'operazione) ma anche dalla forma.

Siccome le stringhe sono incolore, prima di essere immerse nel gel vengono marcate. Dopo l'elettroforesi possono essere recuperate: i pezzi di gel contenenti le stringhe si congelano nell'azoto liquido, poi si scongelano e si frullano attraverso un filtro speciale, dove rimangono attaccate solo le stringhe di DNA.

5. **Separazione**

   Nei nostri algoritmi abbiamo bisogno di un metodo che ci permetta di estrarre tutte le stringhe contenenti una data sottostringa. Date in input una provetta T ed una stringa s (i.e. un oligo la cui sequenza di basi corrisponde alla stringa s), una tecnica di separazione fornisce come output due provette T_1 e T_2, di cui la prima contiene tutte le sequenze di T che hanno s come sottostringa e la seconda le rimanenti (tutte le sequenze di T che non contengono s come sottostringa). Ciò corrisponde a fare la differenza di due multinsiemi.

   (a) Filtraggio

   Questa tecnica agisce sulle singole stringhe di DNA (quindi eventualmente è previsto un primo passo di denaturazione); consiste nell'ancorare molte stringhe di tipo $\overline{\alpha}$ (se α è la sottostringa che vogliamo selezionare) ad un supporto magnetico o vetroso, sul quale viene versata, con la dovuta cautela, la soluzione contenuta nella provetta T. Le sottostringhe α si attaccano alle loro complementari ancorate al filtro, e le stringhe non contenenti α passano attraverso il filtro. Si libera in soluzione il contenuto del filtro e in un'altra provetta (che sarà T_2) il resto. Le stringhe che si sono attaccate al filtro (dopo una denaturazione) vanno nella provetta T_1. Talvolta vengono usati come filtri le membrane resinose. Questo procedimento è concettualmente semplice, ma ormai superato.

   (b) Affinità biotina - avidina/streptavidina

   Per aumentare e velocizzare la capacità di annealing di due stringhe complementari si sfrutta la forza attrattiva che si sviluppa tra

la biotina e l'avidina (o anche la streptavidina) cospargendo di queste due sostanze le stringhe destinate all'annealing.

Nei primi esperimenti si immergeva in una soluzione di DNA biotinato un supporto solido su cui erano state ancorate le stringhe $\overline{\alpha}$ cosparse di avidina, e si aspettava che avvenisse l'annealing desiderato prima di portar fuori questo supporto con le doppie stringhe. Questa tecnica, usata per esempio da Lipton nel suo esperimento (vedi paragrafo 3.1), è soggetta a troppi errori, e soprattutto non è accettabile la quantità relativa di stringhe che riesce a "pescare". In parte questo problema è dovuto alla discrepanza tra $\mathbb{R}^2$ ed $\mathbb{R}^3$, nel senso che la quantità di stringhe su una superficie non può essere confrontabile con quella in un contenitore tridimensionale, infatti, come è stato calcolato in [34], 1cm^2 contiene al più 10^{12} stringhe, e dal calcolo che abbiamo fatto in precedenza (parlando della compattezza di infomazione del DNA) si può dedurre che 1cm^3 contiene circa 10^{19} oligonucleotidi. Tra i vari tentativi di superare questo problema ricordiamo quello di [7], il quale descrive un macchinario che separa le stringhe per affinità facendole passare attraverso un gel in cui sono immerse le sottostringhe $\overline{\alpha}$, tale passaggio è guidato da un sofisticato termoregolatore.

Ultimamente, in [60], si è usata la seguente tecnica: nella soluzione con le molecole denaturate si mettono stringhe $\overline{\alpha}$ biotinate in gran quantità, si aspetta che avvenga l'annealing e dopo si aggiungono delle sferette metalliche cosparse di streptavidina. Tutte le doppie stringhe biotinate (quindi quelle contenenti la coppia $\alpha, \overline{\alpha}$) si attaccheranno alle sferette per via del potere attrattivo della streptavidina nei confronti della biotina. A questo punto con un magnete si separano le sferette (e le doppie stringhe attaccate) dal resto, e basta procedere alla denaturazione per avere le stringhe desiderate. L'efficienza di questo metodo è $88 \pm 3\%$, perché il $12 \pm 3\%$ delle stringhe contenenti α finiscono nella provetta sbagliata (T_2).

6. **Amplificazione**

Il DNA risiede nel nucleo delle cellule viventi, dove gioca un ruolo fondamentale in quanto svolge due funzioni importanti: la codifica per la produzione delle proteine e l'autoduplicazione fatta in modo che un'e-

satta copia venga trasmessa alle cellule della prole (vedi Figura.1.9). Questi processi senza l'intervento degli enzimi (che velocizzano le reazioni chimiche di almeno un trilione di volte) avverrebbero troppo lentamente per consentire la vita. La *polimerasi* è l'enzima che, attraverso un'estensione duplicativa, permette al DNA di riprodursi, cosa che a sua volta permette alle cellule di riprodursi, e infine fa sì che noi stessi possiamo riprodurci.

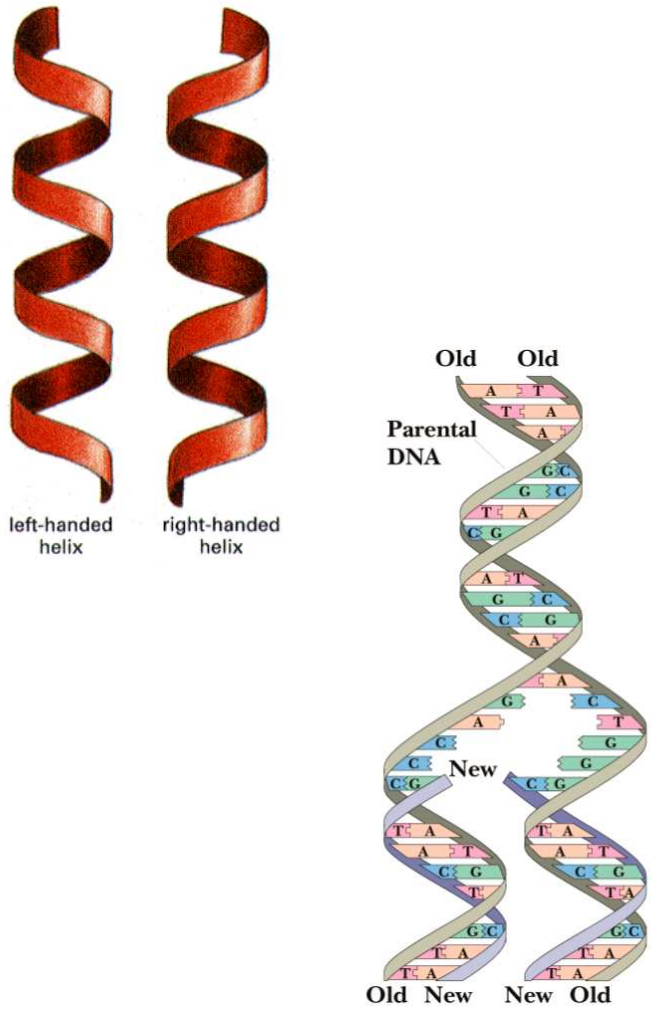

Figura 1.9: Autoduplicazione della doppia elica

Nel 1985 lo scienziato americano K.B. Mullis ha inventato un metodo per moltiplicare le sequenze di DNA in modo molto rapido: si chiama Polimerase Chain Reaction (**PCR**) e riesce a "fotocopiare" il tratto di DNA migliaia di volte. Grazie a questa tecnica in grado di amplificare anche quantità minime di DNA, si sono aperte nuove possibilità. Una di queste, ad esempio, è la famosa prova del DNA che viene usata per stabilire con certezza la paternità o per smascherare un assassino da cellule che può aver lasciato sulla scena del delitto. Sono stati anche messi a punto nuovi test diagnostici, estremamente sensibili, capaci di individuare malattie genetiche e diversi tipi di tumore, prima ancora che i sintomi si manifestino. Le ragioni del successo e della rapida diffusione di tale tecnica vanno ricercate nell'estrema rapidità, nella relativa semplicità di esecuzione, nella notevole sensibilità e specificità della reazione.

Il segreto di questa "reazione a catena" è un enzima: la **polimerasi**. Si tratta, come dice Adleman [3], del "re degli enzimi, il vero costruttore della vita". In condizioni appropriate, partendo da un elica di DNA, la DNA-polimerasi produce una seconda elica complementare alla prima, in cui ogni C è rimpiazzata da una G, ogni G da una C, ogni A da una T e ogni T da una A; praticamente viene copiata l'informazione da una molecola ad un'altra, e questo processo in generale si chiama *polimerizzazione*.

La polimerasi per agire ha bisogno di un "elica stampo" e di una sorta di segnale di inizio che le indichi dove cominciare a fabbricare l'elica complementare. Il segnale è costituito dal *primer*, o *innesco*, che è un tratto di DNA, anche corto, che si appaia all'elica stampo (solitamente in una zona terminale) grazie alla sua complementarietà. Dove si è formata una porzione di doppia elica fra il primer e lo stampo, la DNA-polimerasi comincia ad allungare il primer aggiungendo ad uno ad uno i nucleotidi e formando alla fine una copia complementare allo stampo (vedi Figura 1.10). Una cosa importante è che la polimerasi prolunga il filamento esclusivamente nella direzione $5' - 3'$.

Come eccezione a questa regola esistono delle polimerasi che estendono un filamento di DNA senza che dall'altra parte ci sia un'elica stampo, per esempio la *terminal transferase*, che viene anche usata nel DNA Computing.

Molte polimerasi hanno anche l'attività di un'exonucleasi che può tagliare in entrambe le direzioni, e questo è cruciale nel processo di duplicazione del DNA: serve per "tornare indietro" in caso di errore. Tutto questo ci porta ad osservare che la DNA-polimerasi ha molto in comune con una macchina di Turing: si tratta infatti di una molecola capace di "saltare" su un'elica di DNA e di scorrere lungo il filamento, di "leggere" ogni base che incontra e di "scrivere" la sua forma complementare in una nuova elica di DNA che sta crescendo, e ha anche la facoltà di "cancellare".

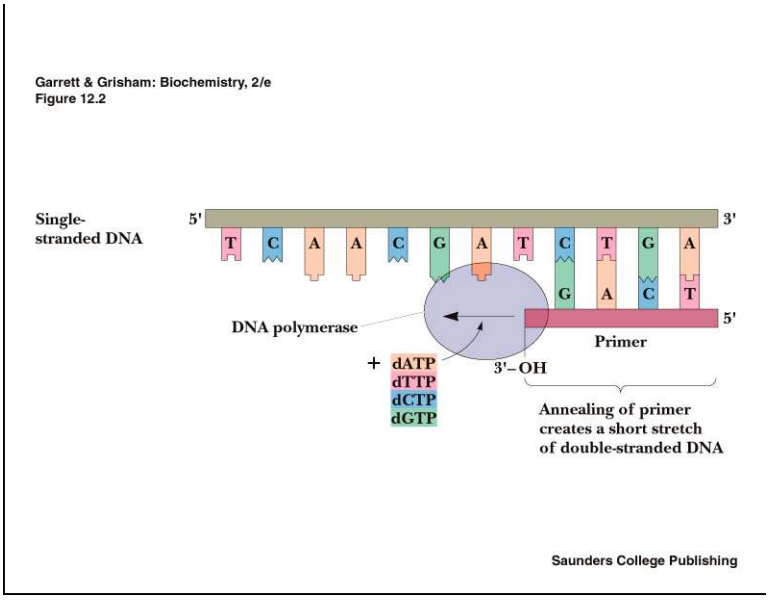

Figura 1.10: La DNA polimerasi

Per descrivere la PCR supponiamo di voler amplificare una molecola (doppia stringa) di DNA, che chiamiamo α, e di conoscerne le parti terminali, γ (estremo $5'$ di una delle stringhe) e β (estremo $5'$ dell'altra stringa). È necessario disporre degli oligonucleotidi $\overline{\beta}$ e $\overline{\gamma}$, chiamati *primers*, di una polimerasi resistente al calore (che possiamo isolare dai batteri che vivono nelle sorgenti termali ad una temperatura vicino ai

100° C), e di svariati nucleotidi. La tecnica PCR consiste in cicli di tre fasi: denaturazione, primer, ed estensione duplicativa.

(a) La prima fase consiste in una denaturazione della doppia stringa, e viene effettuata portando la soluzione (contenente sia le molecole α, che una quantità di primers 100 volte superiore a quella di α, che singoli nucleotidi) a 90° C.

(b) Nella seconda fase la temperatura viene abbassata fino a 55° – 65° C, favorendo così l'ibridizzazione tra le stringhe e i primers (nonostante per rinaturare le stringhe sia richiesta una temperatura inferiore, a volte avviene ugualmente il riattaccamento delle stringhe stampo della PCR e di solito si cercano delle strategie per evitare questo inconveniente). È chiaro che la grande quantità di primers garantisce la capacità ricombinante dell'operazione.

(c) Nella terza fase la soluzione viene portata a 72° C, e viene fatta agire una polimerasi (solitamente la Taq) che procede all'estensione duplicativa delle due stringhe.

È evidente che dopo un ciclo di PCR si hanno 2 copie identiche della molecola di partenza α, e che dopo n cicli (effettuati sulla stessa soluzione perchè la polimerasi resiste ai 90° C della prima fase ma agisce solo ai 72° C della terza fase) si ottengono 2^n molecole α (vedi Figura 1.11).

La PCR è una tecnica di routine usata nel DNA Computing per amplificare le molecole durante la computazione; spesso provoca la formazione indesiderata di complesse strutture tridimensionali che si cerca di ridurre con una scelta opportune delle sequenze e della loro lunghezza [60].

7. **Marcatura**

(a) Chemiluminescenza, fluorescenza

Si "tingono" le doppie stringhe di DNA con particolari reagenti: per esempio col bromuro di etidio che risulta fluorescente sotto la luce ultravioletta.

(b) Radioattività

Si attaccano dei marcatori radioattivi alle estremità delle molecole di DNA, che così risultano radioattive.

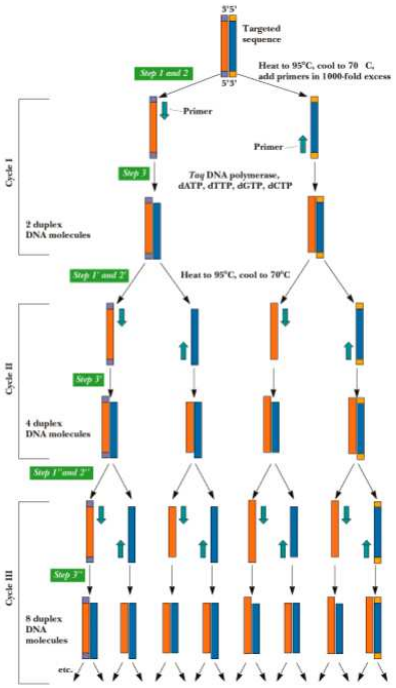

Figura 1.11: Tecnica PCR

8. **Metilazione, fosforilazione**

   Le methylasi sono enzimi che la natura usa assieme agli enzimi di restrizione. Un batterio usa gli enzimi di restrizione per fare a pezzi il DNA dei virus invasori, quindi avere un gran varietà di enzimi di restrizione lo porta a potersi difendere da una gran varietà di virus invasori, ma nello stesso tempo deve proteggere il proprio DNA dall'attacco di tali enzimi: ne "modifica" le sequenze che corrispondono al sito di ri-

conoscimento. Per far questo utilizza la methylasi che ha lo stesso sito di riconoscimento dell'enzima di restrizione, la quale, aggiungendo un gruppo metilico ad un nucleotide del sito, lo protegge dall'attacco dell'enzima di restrizione corrispondente. In laboratorio si utilizza questa stessa tecnica del batterio, detta *metilazione*, per alterare le molecole di DNA.

La *defosforilazione*, togliendo il gruppo fosforico dalle estremità $5'$ della molecola, impedisce che lì si possa formare un legame fosfodiesterico. Questa procedura viene effettuata tramite l'azione dell'enzima *fosfatasi alcalina*.

La procedura inversa, la *fosforilazione*, consiste nel ripristino del gruppo fosforico (ad opera dell'enzima *kinasi*), e talvolta viene effettuata negli algoritmi solo per garantire il recupero dei gruppi fosforici che possono essersi staccati nei vari maneggiamenti del DNA [60].

9. **Linearizzazione**

   La capacità di "leggere" rapidamente tutti i nucleotidi di un frammento di DNA rende possibile per esempio determinare i contorni di un gene e la sequenza di aminoacidi che esso codifica. Ma anche per quanto riguarda gli algoritmi del DNA Computing è una cosa di grande utilità, infatti grazie al famoso **metodo di Sanger** riusciamo a leggere le stringhe di DNA che codificano la soluzione del nostro problema.

   L'idea, semplice ed elegante, si può schematizzare come segue.

   La sequenza di DNA che vogliamo leggere (di lunghezza t) si identifichi con una corda di cui si sa qual è l'inizio e quale la fine; sulla corda si distribuiscano i "nodi" A, C, G, T, che rappresentano, nell'ordine, i nucleotidi della sequenza. Si immagini di avere moltissime copie di questa corda, e di disporre di 4 forbici F_A, F_C, F_G, F_T, di cui ognuna taglia in modo diverso: F_A taglia solo i nodi A, F_C solo i nodi C, F_G solo i nodi G ed F_T solo i nodi T. Ogni corda può essere tagliata una sola volta. Con questi dati si vuole conoscere la disposizione dei nodi nella corda (ovvero la disposizione dei nucleotidi nella sequenza).

   Facciamo agire, non deterministicamente, queste forbici sulla moltitudine di corde, fin quando non si ottengono corde di tutte le possibili lunghezze (per esempio se la corda è lunga 10, si aspetta fin quan-

do non ci sono pezzi lunghi $1, 2, 3, 4, 5, 6, 7, 8, 9, 10$). Poi raccogliamo, separatamente, i prodotti di ciascuna forbice e li misuriamo.

La sequenza è data, nell'ordine, dagli indici delle forbici che hanno tagliato i pezzi lunghi $1, 2, 3, \ldots, t$ rispettivamente.

Andiamo a vedere la combinazione di operazioni biologiche che effettua questo algoritmo.

Assumiamo di avere una soluzione *omogenea*, cioè contenente principalmente molte copie (eventualmente dopo un'amplificazione) della singola stringa α che vogliamo sequenziare.

Aggiungiamo una stringa γ nota, per esempio di lunghezza 20, all'estremità $3'$ di α (sia $\beta = \gamma\alpha$) in modo che una polimerasi possa agire utilizzando $\overline{\gamma}$ come primer e α come stampo.

Ibridizziamo le stringhe β con dei $\overline{\gamma}$ fluorescenti e suddividiamo la soluzione ottenuta in 4 vaschette, che indichiamo con A, G, C e T. Nella vaschetta A ci sono diversi nucleotidi, una polimerasi che non abbia l'attività di exonucleasi, e dei nucleotidi A "alterati": nel metodo di Sanger sono dideossinucleotidi (ddA), ovvero nucleotidi tali che il loro $3'$ -OH è stato modificato in $3'$ -H, in generale sono nucleotidi chimicamente modificati nello zucchero e/o nel gruppo fosforico e/o nella base azotata. Analogamente nelle altre vaschette, con i nucleotidi ddG, ddC e ddT.

Nelle singole vaschette generiamo una PCR modificata, in quanto l'estensione di $\overline{\gamma}$ si blocca quando viene preso un nucleotide modificato (perché non possono avvenire i legami fosfodiesterici che fanno crescere in lunghezza la nuova stringa).

Facciamo un'elettroforesi con quattro pozzetti A, C, G, e T, in ciascuno dei quali viene posto il DNA ottenuto dalla vaschetta corrispondente, dopo la denaturazione e la selezione delle singole stringhe contenenti $\overline{\gamma}$ (in questa fase si sfrutta la fluorescenza di $\overline{\gamma}$). Se supponiamo che la PCR modificata complessivamente si sia bloccata in tutti i punti possibili, dalla disposizione delle stringhe delle varie vaschette sulla lastra dell'elettroforesi possiamo risalire alla lettura (fatta dall'elettrodo positivo a quello negativo, perché la lunghezza delle stringhe cresce in questa direzione) di $\overline{\alpha}$, e quindi a quella di α.

Metodi più recenti di quello di Sanger si avvalgono della stessa idea ma usano tingere i nucleotidi con quattro diversi colori fluorescenti, in modo che le quattro basi possano essere processate simultaneamente.

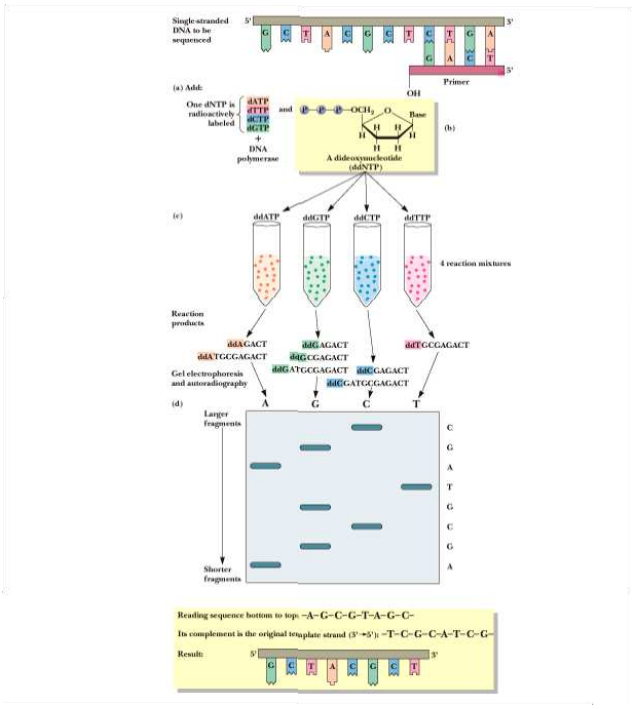

Figura 1.12: Metodo di Sanger

L'implementazione di questo metodo richiede molta cautela:

- la sequenza iniziale non deve essere troppo lunga, eventualmente si deve prima frammentare e poi ricostruire,

- calcolare la percentuale "giusta" di nucleotidi alterati/marcati chimicamente: se ne mettiamo troppi dopo la PCR non si trovano le sequenze più lunghe, se ne mettiamo pochi non si trovano le sequenze più corte, invece un buon numero favorisce la formazione di sequenze di tutte le lunghezze,
- trovare una polimerasi che non abbia anche un'attività di exonucleasi, altrimenti potrebbe tagliare i nucleotidi alterati durante la fase di polimerizzazione impedendone l'arresto (di solito si usa la Kenow fragments).

10. **Discriminazione**

    È l'operazione che, con tecniche chiamate DGGE (Denaturing Gradient Gel Electrophoresis), in una soluzione discrimina il DNA in relazione alla quantità di G presenti. Si basa sulla regolazione precisa della temperatura, in quanto la temperatura di melting T_m è proporzionale al numero di coppie C-G della molecola (perché il legame C-G è più forte di quello A-T, vedi paragrafo 1.1).

11. **Subclonazione**

    La *clonazione* del DNA permette di partire da una singola molecola e generarne bilioni di copie identiche. Questa tecnica consiste nel tagliare un plasmide (che viene chiamato *vettore clonante*) con un enzima di restrizione e inserirci un frammento esterno di DNA. Le estremità del plasmide si richiudono e si forma così un *plasmide ricombinante*. Il plasmide in condizioni favorevoli si riproduce velocemente, e in ognuno dei nuovi plasmidi c'è una copia della stringa di DNA iniziale, che siamo in grado di estrarre perché sappiamo esattamente dove si posiziona all'interno del plasmide.

    In [37] è stata usata la *subclonazione* per separare i vari tipi di stringhe (prodotti finali dell'esperimento) prima di sequenziarli: sono stati *infettati* dei plasmidi con una scarica elettrica (in questo modo si riesce ad inserirci al più una stringa), è bastato dargli dello zucchero perché si riproducessero, e infine tutti i plasmidi si sono agglomerati in colonie separate in relazione al tipo di stringa che avevano dentro. A questo punto si è saputo esattamente quante diverse stringhe di DNA c'erano (tante quante le colonie), e grazie a questa differenzazione si sono

potute sequenziare (è necessaria una soluzione iniziale *omogenea*, vedi punto 9).

12. **Test**

    Data la provetta finale dell'esperimento, l'operazione *test* dice se questa non contiene alcuna molecola di DNA o se ne contiene almeno una, in tal caso viene estratta una stringa dalla provetta e viene "letta", o più propriamente "linearizzata", col metodo di Sanger.

Concludiamo trovando delle caratteristiche comuni ad alcune delle 12 operazioni descritte in questo paragrafo. Possiamo infatti fare una distinzione generale tra le operazioni da provetta (come unione, test, separazione) quelle enzimatiche (restrizione, metilazione), e quelle fatte con strumentazioni varie, termoregolatori nel caso della denaturazione e discriminazione, frullatori, pettini, colonne cromatografiche, computer di vario tipo per elettroforesi, PCR e tecniche molto più sofisticate.

Osserviamo infine che, col tipo di operazioni che abbiamo descritto, i processori molecolari risultano dinamici e transienti (in movimento e soggetti a cambiamenti fisici), a differenza dei microprocessori dei computers che sono statici e permanenti.

1.6 Complessità

La teoria della complessità computazionale concerne la quantificazione delle risorse (generalmente spazio e tempo) che necessitano per risolvere i problemi computazionali. Ma nel caso del DNA Computing serve una nuova nozione di complessità, che catturi la realtà chimico-fisica in cui le reazioni hanno luogo. Nell'analisi di un algoritmo DNA potrebbe avere rilevanza il numero di molecole che galleggia nella soluzione (di un dato volume ed una data densità), o il fatto che le molecole sono sensibili alle variazioni di temperatura e soggette ad operazioni di vario grado di complessità e difficoltà implementativa. In [13] per esempio, si definisce la complessità di una provetta di DNA come il numero dei diversi tipi di stringa.

Le istanze dei problemi vengono codificate in molecole, generalmente oligonucleotidi o piccole stringhe di DNA, sebbene siano possibili strutture tipo grafi [21]. Queste molecole sono contenute in una provetta, che può essere descritta astrattamente come un multinsieme di stringhe che ha un certo

numero di attributi fisici. In particolare ci sono il volume, la temperatura, il numero di nucleotidi e l'ammontare di ciascun tipo di molecola contenuta nella provetta. Questo stato fisico-chimico può essere cambiato da un prescritto numero di operazioni elementari, che sono generalmente di due tipi: fisiche e chimiche. Sono esempi di operazioni fisiche i cambiamenti di temperatura, gli attaccamenti di sferette metalliche, la separazione attraverso campi elettrici o forze centrifughe, l'unione delle provette; mentre abbiamo esempi di operazioni chimiche nelle reazioni che avvengono con gli enzimi (endonucleasi, ligasi, exonucleasi, polimerasi).

Un fattore rilevante nella complessità di un algoritmo DNA è sicuramente la quantità di molecole DNA che necessitano, e questo è legato allo *spazio* delle soluzioni del problema. Nei primi algoritmi DNA, ad ogni passo della computazione il volume di DNA presente nella provetta si mantiene esponenziale rispetto alla dimensione del problema; e anche molti algoritmi proposti successivamente, realizzabili in tempo polinomiale, lineare, costante, hanno sempre un supporto di DNA esponenziale con la dimensione del problema. Un limite pratico al massimo volume consentito è dato dal numero di Avogadro (6.02×10^{23}), e una funzione esponenziale raggiunge questo limite troppo rapidamente. L'algoritmo DNA così come è stato formulato da Adleman (vedi paragrafo 1.7) per esempio, è in grado di risolvere un problema di cammini hamiltoniani su un grafo che abbia al più 70 nodi, mentre già su un grafo con 200 vertici sarebbe necessaria una quantità di DNA paragonabile a quella della Terra!

Sembra quindi che, in generale, l'approccio della forza bruta (quello di generare tutte le soluzioni possibili per poi filtrare gradualmente quelle buone) comporti un'impraticabile quantità di DNA iniziale, almeno per problemi di grosse dimensioni, e a questo proposito sono stati proposti degli algoritmi alternativi [8]. In [60] (per i dettagli dell'algoritmo vai al paragrafo 3.4) per esempio, la quantità di DNA (per la risoluzione di un 3-SAT) è stata ridotta (da 2^n) a $2^{0.58n}$, con n numero di variabili del problema. Un'altra idea sarebbe quella di costruire direttamente le soluzioni buone con varie tecniche, ma fare questo richiede un tempo esponenziale.

La quantità di DNA al momento è un grosso limite per quello che si chiama lo "scale-up" degli algoritmi, ovvero l'aumento della dimensione dei problemi concretamente risolubili.

Molti modelli esistenti quantificano la complessità del *tempo* degli algorit-

mi DNA contando il numero di "passi biologici" richiesti per risolvere il dato problema, e questi passi comprendono la creazione delle stringhe iniziali codificanti i dati, la separazione di sottinsiemi di stringhe, la scelta di stringhe sulla lunghezza; ma si tratta di un modello poco realistico [6].

Sicuramente la complessità di un programma DNA dipende dall'insieme di bio-operazioni elementari che utilizza, ma possiamo indicare come operazione dominante [42] la separazione (vedi punto 5), che solitamente comporta le procedure più lunghe e delicate.

In conclusione, possiamo dire che, per il momento, sono due i parametri essenziali nella valutazione di un algoritmo DNA:

1. la dimensione dello *spazio delle soluzioni*, che corrisponde alla quantità di DNA necessaria per codificare l'insieme delle soluzioni candidate.

2. il numero dei passi di *separazione* necessari per ottenere la provetta finale.

1.7 Esperimento di Adleman

Nel novembre del 1994 Adleman, in una settimana di laboratorio, ha realizzato quello che è passato alla storia come il primo esperimento di DNA Computing. Ha risolto, con gli strumenti della biologia molecolare, un'istanza di un problema di cammini hamiltoniani in un grafo orientato: un piccolo grafo (avente 7 nodi) è stato codificato in molecole di DNA, e le operazioni della computazione sono state realizzate da enzimi e protocolli standard di laboratorio. Da lì è nata un'area emergente della scienza, resa possibile dalla nostra capacità sempre maggiore di controllare il mondo molecolare.

Ecco come Adleman [1] ha affrontato il problema.

Definizione 1.1. *Un grafo orientato con vertici designati v_{in} e v_{out} ha un cammino hamiltoniano sse esiste una sequenza compatibile di lati direzionati (cioè un cammino) che comincia in v_{in}, finisce in v_{out}, e passa da ogni altro vertice esattamente una volta.*

Il seguente algoritmo, non deterministico, risolve il problema di dire se, dato un grafo, esiste un cammino hamiltaniano.

1. Si generino in modo random (casuale) cammini che attraversano il grafo.
2. Si prendano solo i cammini che cominciano con v_{in} e finiscono con v_{out}.
3. Se il grafo ha n vertici, si prendano solo i cammini che passano da n vertici (ovvero quelli con $n-1$ archi).
4. Si prendano solo i cammini che passano da ciascun vertice del grafo almeno una volta.
5. Se rimane qualche cammino, l'algoritmo risponda affermativamente, altrimenti negativamente.

Questo non è un algoritmo perfetto; tuttavia, se i percorsi vengono generati in modo casuale e in quantità sufficiente, ci sono buone probabilità di ottenere una risposta corretta.

Passiamo all'implementazione a livello molecolare, che è stata fatta per un grafo orientato avente 7 vertici v_i con $i = 0, \ldots, 6$, in cui $v_{in} = v_0$ e $v_{out} = v_6$.

Notazione. Denotiamo con $\overline{\alpha}$ la sequenza complementare secondo Watson-Crick della sequenza α.

A ciascun vertice v_i del grafo viene associata una sequenza random di 20 nucleotidi, di cui denotiamo con α_i i primi 10 e con β_i gli ultimi 10:

$$v_i \longleftrightarrow O_i = \alpha_i \beta_i$$

a ciascun un arco orientato $\overrightarrow{a_{ij}}$ con $0 < i < 6$ e $0 < j < 6$:

$$\overrightarrow{a_{ij}} \longleftrightarrow O_{i \to j} = \overline{\beta_i}\,\overline{\alpha_j}.$$

Gli oligonucleotidi del tipo $\overline{\beta_i}\,\overline{\alpha_j}$ risultano lunghi 20; agli archi $\overrightarrow{a_{1j}}$ e $\overrightarrow{a_{i6}}$ invece associamo i seguenti nucleotidi lunghi 30:

$$\overrightarrow{a_{1j}} \longleftrightarrow O_{1 \to j} = \overline{O_1}\,\overline{\alpha_j}$$
$$\overrightarrow{a_{i6}} \longleftrightarrow O_{i \to 6} = \overline{\beta_i}\,\overline{O_6}$$

Osserviamo che tale codifica preserva l'orientazione dei archi, infatti in generale $O_{i \to j} = \overline{\beta_i}\,\overline{\alpha_j} \neq \overline{\beta_j}\,\overline{\alpha_i} = O_{j \to i}$.

La scelta degli oligonucleotidi random di lunghezza 20 è stata fatta per garantire con buona probabilità la non formazione di strutture secondarie e la differenzazione tra le codifiche dei vertici. Inoltre, con questa lunghezza, le molecole codificanti i cammini del grafo sono stabili anche a temperatura ambiente, e quindi possono essere maneggiate con più facilità.

Dopo la codifica, si parte con l'esperimento avendo circa 3×10^{13} copie degli oligonucleotidi codificanti i vertici e gli archi. E si eseguono, rispettivamente, le seguenti procedure:

1. Annealing + Ligation.

   Tutti i cammini del grafo orientato vengono creati quasi contemporaneamente dall'interazione simultanea di migliaia di miliardi di molecole (una reazione biochimica di questo genere rappresenta uno straordinario esempio di calcolo parallelo). In questo passo vengono eseguite circa 10^{14} operazioni.

   Nell'algoritmo si assume che se i cammini hamiltoniani esistono, allora si formano durante questa operazione, e questo si basa interamente sul comportamento ricombinante delle molecole del DNA.

2. PCR con i primers O_0 e $\overline{O_6}$.

   Vengono amplificati i cammini che cominciano con v_{in} e finiscono con v_{out}.

   Una variante poteva essere quella di imporre nella codifica l'inizio del cammino in v_{in} e la fine in v_{out}, per esempio eliminando l'ossigeno del gruppo OH di $3' - v_{in}$ e facendo una defosforilazione in $5' - v_{out}$, in modo che v_{in} e v_{out} non potessero far parte del cammino come nodi intermedi.

3. Elettroforesi + PCR.

   Vengono selezionate tramite elettroforesi le doppie stringhe lunghe 140 bp, perché sono quelle che rappresentano i cammini aventi esattamente $n - 1$ archi, e vengono amplificate.

4. Denaturazione + Separazione per affinità biotina-avidina.

   Denaturando il prodotto del passo precedente si ottengono le singole stringhe, che costituiscono l'input della separazione.

Vengono quindi selezionate le stringhe contenenti la sottostringa O_0 (utilizzando $\overline{O_0}$), e di queste quelle contenenti la sottostringa O_1, e così via, fino all'ultima selezione che trattiene le stringhe contenenti la sottostringa O_6. Di volta in volta si seleziona il multinsieme

$$\{\gamma \,/\, O_i \in \gamma\} \quad i = 0, \ldots, 6$$

dove "$\in$" vuol dire "ne fa parte come sottostringa" e γ sono le sequenze DNA codificanti i cammini sul grafo.
In tal modo vengono estratti i cammini che passano da ogni vertice del grafo almeno una volta.

Questo è stato il passo più laborioso, quello che ha richiesto un'intera giornata di lavoro intenso. È una fase molto delicata dell'algoritmo, in quanto gli eventuali errori possono farci perdere cammini hamiltoniani: si cerca di prevenire una cosa del genere con cicli di PCR che ne aumentano notevolmente le copie.

5. PCR + Test.

   Si amplificano gli eventuali cammini che sono rimasti dopo i passi precedenti, e in un secondo momento, con l'operazione test, l'algoritmo dice se esiste o meno un cammino hamiltoniano.

   In caso di risposta affermativa, il metodo di Sanger ci permette di "leggere" la soluzione.

Il numero di bio-operazioni cresce linearmente con il numero di vertici del grafo e il numero di oligonucleotidi diversi cresce linearmente col numero di archi del grafo. La quantità di ciascun oligonucleotide è una questione di teoria dei grafi: deve essere tale da garantire che, durante il primo passo dell'algoritmo, si venga a formare un cammino hamiltoniano (se esiste) con alta probabilità; tale quantità cresce esponenzialmente con il numero di vertici del grafo.

Da un punto di vista teorico può sembrare tutto semplice e realizzabile, ma da un punto di vista pratico le cose non stanno affatto così, dietro le procedure che abbiamo descritto per l'esperimento di Adleman ci sono diversi fattori di cui tener conto: la regolazione attenta della temperatura, del ph, della concentrazione salina.

In generale è importante la scelta delle stringhe, per esempio la lunghezza degli oligonucleotidi ha effetto sulla temperatura di melting e sulla affidabilità dell'annealing. In [50] si suggerisce di massimizzare l'Hamming distance (definita come il numero di basi diverse tra due stringhe) tra tutte le possibili coppie di stringhe nella codifica (e anche su tutte le stringhe complementari), ed è anche preferibile minimizzare la differenza di T_m tra le sequenze, in modo che l'ibridizzazione possa avvenire simultaneamente.

Sebbene negli ultimi anni le nostre informazioni sul DNA siano aumentate notevolmente, soprattutto durante il progetto genoma cominciato nel 1988 (che oltre a studiare la locazione dei 32.000 geni umani si propone di analizzare la struttura del DNA), molti processi del DNA Computing sono ancora soggetti a troppi errori, e non sempre controllabili. In [9] viene proposto un algoritmo che trasforma una computazione DNA soggetta ad errori in una che ne è relativamente priva, e sostanzialmente si dimostra che la perdita di stringhe e l'appaiamento non standard non costituiscono ostacoli insormontabili per le computazioni con il DNA.

Infine, oltre all'accuratezza delle bio-operazioni, un pericolo per l'implementazione delle computazioni col DNA è il fatto che la dimensione del problema influenza la concentrazione dei reagenti, in particolare la quantità di DNA influenza la concentrazione salina, e questo può avere effetto sulla qualità dei prodotti finali della computazione.

Se il DNA Computing ha un significato in termini pratici per le applicazioni informatiche è ancora una questione prematura, perché le tecniche biochimiche non sono ancora sufficientemente sofisticate o accurate, e in particolare non si sono ancora sviluppate adeguatamente verso le esigenze specifiche del DNA. La percentuale dell'errore nelle operazioni con le stringhe di DNA può fare la differenza in modo veramente drammatico: il successo definitivo del DNA Computing dipende pesantemente dallo sviluppo di particolari tecniche di laboratorio.

Oggi le sfide del DNA Computing sono: uno studio approfondito degli errori reversibili e irreversibili delle bio-operazioni, l'aumento della dimensione dei problemi affrontati col DNA, l'individuazione di buone codifiche. Sarebbe auspicabile avere una migliore comprensione delle proprietà e dei processi di base delle biomolecole, soprattutto DNA e RNA, e misurazioni più accurate sulla riproducibilità ed efficienza di tali meccanismi.

Le speranze più grosse del DNA Computing per il lontano futuro sono

quelle di favorire una fertile collaborazione con la biologia molecolare, la biologia computazionale e l'evolutionary computing, di riuscire a costruire computers DNA che affrontino i problemi "difficili" per i computers elettronici, e di aumentare l'interesse che porta ad esplorare la connessione tra la vita e la computazione.

Il successo dell'esperimento di Adleman ha dimostrato che i metodi standard della biologia molecolare possono essere usati per risolvere (una piccola istanza di) un problema computazionalmente "intrattabile". Ma sarebbe sbagliato vedere questo tipo di ricerche solo da un punto di vista applicativo. Infatti, così come l'esperimento di Adleman si basa sulle doppie stringhe e la complementarietà del DNA e sulle reazioni biochimiche, che avvengono in parallelo, similmente Rozenberg & Salomaa deducono l'universalità del DNA Computing dal fatto che la complementarietà di Watson-Crick è presente in qualche forma in tutti i modelli del DNA Computing e, allo stesso tempo, la complementarità ha la stessa forma del linguaggio Twin-Shuffle, riconosciuto come base dell'universalità (Engelfriet & Rozenberg, vedi Teorema 2.105).

Nella teoria dei linguaggi formali questa possibilità sperimentale ha ispirato diversi studi (per esempio [48] e [51]) relativi ai modelli matematici che descrivono il comportamento ricombinante del DNA.

Bibliografia

[1] L. M. Adleman, *Molecular Computation of Solutions to Combinatorial Problems*, Science, **266**, 1021-1024, 1994.

[2] L. M. Adleman, *On Constructing A Molecular Computer*, R. J. Lipton E. B. Baum eds., DNA Computers: Proceeding of a DIMCS workshop, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, 1-21, 1996.

[3] L. M. Adleman, *Fare calcoli con il DNA*, Le Scienze, **362**, 82-89, 1998.

[4] B. Alberts et al., *Essential Cell Biology, An Introduction to the Molecular Biology of the Cell*, Garland Publ. Inc., New York, London, 1998.

[5] M. Amos, A. Gibbons, D. Hodgson, *Error-resistant Implementation of DNA Computations*, Research Report 298, Department of Computer Science, University of Warwick, 1996.

[6] M. Amos, A. Gibbons, P. E. Dunne, *The Complexity and Viability of DNA Computations*, CTAG-97001, Department of Computer Science, University of Liverpool, 1997.

[7] R. S. Braich, C. Johnson, P. W. K. Rothemund, D. Hwang, N. Chelyapov, L. M. Adleman, *Solution of a Satisfiability Problem on a Gel-Based DNA Computer*, in [11], 31-42, 2000.

[8] D. Boneh, C. Dunworth, R. J. Lipton, J. Sgall, *On The Computational Power of DNA*, Princeton University Computer Science Technical Report, TR-499-95, 1995.

[9] K. Chen, E. Winfree, *Error Correction in DNA Computing: Misclassification and Strand Loss*, in [59], 49-62, 2000.

[10] G. Ciobanu, *A Molecular Abstract Machine*, in [47], 61-79, 1998.

[11] A. Condon, G. Rozenberg eds., *6th International Meeting On DNA Based Computers*, Leiden, 2000.

[12] T. L. Eng, *On solving 3CNF-satisfiability with an in vivo algorithm*, BioSystems, **52**, 135-141, 1999.

[13] D. Faulhammer, R. J. Lipton, L. F. Landweber, *Counting DNA: estimating the complexity of a test tube of DNA*, BioSystems, **52**, 193-196, 1999.

[14] C. Frontali, *Limiti dei modelli linguistici applicati al DNA*, pubblicato in *L'informazione nelle scienze della vita* a cura di B. Continenza e E. Gagliasso, Franco Angeli editore, 1998.

[15] R. H. Garret, C. M. Grisham, *Biochemistry*, Saunders College Publishing, Harcourt, 1997.

[16] M. H. Garzon, N. Jonoska, S. A. Karl, *The bounded complexity of DNA computing*, BioSystems, **52**, 63-72, 1999.

[17] I. P. Gent, T. Walsh, *The satisfiability constraint gap*, Artificial Intelligence, **81**, 59-80, 1996.

[18] T. Head, *Hamiltonian Paths and Double Stranded DNA*, in [47], 80-92, 1998.

[19] T. Head, G. Rozenberg, R. S. Bladergroen, C. K. D. Breek, P. H. M. Lommerse, H. P. Spaink, *Computing with DNA by operating on plasmids*, BioSystems, **57**, 87-93, 2000.

[20] T. Head, X. Chen, M. J. Nichols, M. Yamamura, S. Gal, *Aqueous Solutions of Algorithmic Problems: emphasizing knights on a* 3X3, in [23], 219-230, 2001.

[21] N. Jonoska, S. A. Karl, M. Saito, *Three dimensional DNA structures in computing*, BioSystems, **52**, 143-153, 1999.

[22] N. Jonoska, S. A. Karl, M. Saito, *Graph Structures in DNA Computing*, in [47], 93-110, 1999.

[23] N. Jonoska N. C. Seeman eds, *7th International Meeting on DNA Based Computers*, Preliminary Proceedings, Tampa USA (FL), 2001.

[24] L. Kari, *DNA Computing: the arrival of biological mathematics*, The Mathematical Intelligencer, **19** (2), 9-22, 1997.

[25] S. Kauffman, *Investigations*, Oxford University Press, 188-194, 2000.

[26] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, *Optimization by Simulated Annealing*, Science, **220**, 671-680, 1983.

[27] S. Kirkpatrick, *Configuration space analysis of travelling salesman problems*, J. Physique, **46**, 1277-1292, 1985.

[28] S. Kirkpatrick, B. Selman, *Critical behavior in the satisfiability of random Boolean expressions.*, Science, **264**, 1297-1301, 1994.

[29] M. G. Lagoudakis, T. H. LaBean, *2D DNA Self-Assembly for Satisfiability*, in [59], 141-154, 2000.

[30] L. F. Landweber E. B. Baum eds, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, DNA Based Computers II, **44**, 1999.

[31] J. V. Leeuwen, *Handbook of Theoretical Computer Science*, Press, Utrecht, 1990.

[32] Z. Li, *Algebraic Properties of DNA Operations*, in [47], 327-339, 1998.

[33] R. Lipton, *DNA Solutions of Hard Computational Problems*, Science, **268**, 542-545, 1995.

[34] Q. Liu, A. G. Frutos, L. Wang, A. J. Thiel, S. D. Gillmor, C. T. Strother, A. E. Condon, R. M. Corn, M. G. Lagally, L. M. Smith, *Progress toward demonstration of a surface based DNA computation: a one word approach to solve a model satisfiability problem*, BioSystems, **52**, 25-33, 1999.

[35] V. Manca, *String Rewriting and Metabolism: A logical Perspective*, in [47], 36-60, 1998.

[36] V. Manca, C. Martin-Vide, G. Păun, *New Computing Paradigms suggested by DNA Computing: computing by carving*, BioSystems, **52**, 47-54, 1999.

[37] V. Manca, S. Di Gregorio, D. Lizzari, G. Vallini, C. Zandron, *A DNA Algorithm for 3-SAT(11,20)*, in [23], 167-178, 2001.

[38] V. Manca, *Formal logic*, J. G. Webster ed, Wiley Encyclopedia of Electrical and Electronics Engineering, **7**, John Wiley&Sons, 2001.

[39] V. Manca, *Logica Matematica*, Bollati Boringhieri, 2001.

[40] V. Manca, *Logical string rewriting*, Theoretical Computer Science (TCS), **264**, 25-51, 2001.

[41] V. Manca, *Membrane Algorithms for Propositional Satisfiability*, Workshop on Membrane Computing, Curtea de Arges, Romania, TR 17, GRLMC, Univ. Rovira i Virgili, Terragona (Spain), 181-192, 2001.

[42] V. Manca, C. Zandron, *A Clause String DNA Algorithm for SAT*, N. C. Seeman N. Jonoska eds, Proceedings of the 7th International Meeting on DNA Based Computers, Springer-Verlag, to appear, 2001.

[43] V. Manca, *On the generative power of iterated transduction*, M. Ito Gh. Păun, S. Yu eds., Words, Semigroups, and Translations, World Scientific Publications, Singapore, 2001.

[44] S. Marcus, *Language at the Crossroad of Computation and Biology*, in [47], 1-35, 1998.

[45] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, L. Troyansky, *Determining computational complexity from characteristic "phase transition"*, Nature, **400** I, 133-137, 1999.

[46] M. Ogihara, *Circuit Evaluation: Thoughts on a Killer Application in DNA Computer*, in [47], 111-125, 1998.

[47] G. Păun ed., *Computing with bio-molecules: theory and experiments*, Springer-Verlag, 1999.

[48] G. Păun, G. Rozenberg, A. Salomaa, *DNA Computing: New Computing Paradigms*, Springer-Verlag, 1998.

[49] N. Pisanti, *A survey on DNA computing*, Bulletin of the European Association for Theoretical Computer Science, EATCS, **64**, 188-216, 1998.

[50] J. H. Reif, T. H. LaBean, M. Pirrung, V. S. Rana, B. Guo, C. Kingsford, G. S. Wickham, *Experimental Construction of Very Large Scale DNA Databases with Associative Search Capability*, in [23], 241-250, 2001.

[51] G. Rozenberg, A. Salomaa, *Handbook of Formal Languages*, **3 voll**, 1996.

[52] K. Sakamoto, D. Kiga, K. Komiya, H. Gouzu, S. Yokoyama, S. Ikeda, H. Sugiyama, M. Hagiya, *State transitions by molecules*, BioSystems, **52**, 81-91, 1999.

[53] K. Sakamoto, H. Gouzu, K. Komiya, D. Kiga, S. Yokoyama, T. Yokomori, M. Hagiya, *Molecular Computation by DNA Hairpin Formation*, Science, **288**, 1223-1226, 2000.

[54] A. Salomaa, *Formal Languages*, Academic Press, NewYork and London, 1973.

[55] A. Salomaa, *Jewels of Formal Language Theory*, Computer Science Press, Rockville, 1981.

[56] P. T. Saunders ed., Morphogenesis, in Collected Works of A. M. Turing, North-Holland, Amsterdam, 1992.

[57] C. E. Shannon, *A Universal Turing Machine With Two Internal States*, C.J. Mc Carthy, C. E. Shannon eds., Automata Studies, Princeton Univ. Press, Princeton, 157-165 , 1956.

[58] Y. Takenaka, A. Hashimoto, *A proposal of DNA computing on beads and its application to SAT problems*, in [23], 331-339, 2001.

[59] E. Winfree D. K. Gifford eds, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, DNA Based Computers V, **54**, 2000.

[60] H. Yoshida, A. Suyama, *Solution to 3-SAT by Breadth First Search*, in [59], 9-22, 2000.