La Grammatica del DNA

IIS E. Majorana Mirano (Ve)
Progetto Math en Jeans
2018-2019

Introduzione

La matematica nel DNA: checksum

Gli algoritmi “checksum” rilevano la presenza di errori nei codici della carta di credito, del codice IBAN, del codice fiscale, ...
Pare accertato che anche il DNA abbia un checksum in grado di rilevare dati mancanti e di riscostruirli.

Negli anni 1940 Barbara McClintock
(premio Nobel Medicina 1983)
fece un esperimento: danneggiò alcune parti del DNA di alcune piante di mais.
Le piante furono in grado di riconoscere le parti danneggiate e ricopiare il codice corretto da parti sane del DNA.

La matematica nel DNA: checksum

Come fanno le piante a correggere il DNA danneggiato?
Secondo il francese J. C. Lopez,
nel DNA è incorporato un algoritmo di autocorrezione.

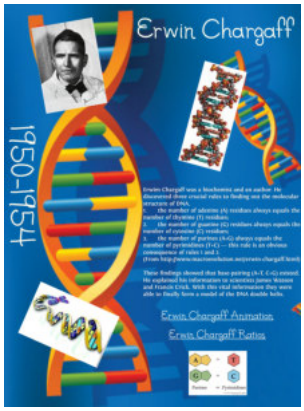
Quando le celle si replicano, esse contano il numero totale di lettere nel DNA della cellula della figlia.
Se i parziali non rispettano alcuni rapporti la cellula madre riconosce l'errore, termina il processo e uccide la cellula figlia.

La matematica nel DNA: pattern ergodico

In un testo inglese la lettera E ha una frequenza di 12.7%, la Z di 0.7%.
Le altre 24 lettere hanno valori compresi tra questi estremi.
In alcuni casi è possibile rilevare errori nei testi contando le lettere.
Nella Teoria della Comunicazione si chiama “pattern ergodico” ed è una caratteristica propria di ogni linguaggio.

Nel DNA le frequenze delle parole seguono delle regole.
Modelli del DNA si fondano su regole di simmetria.
Un esempio è costituito dalle leggi di parità di Chargaff, usate da Watson e Crick a sostegno del loro modello della struttura del DNA a doppia elica.

Le leggi di Chargaff

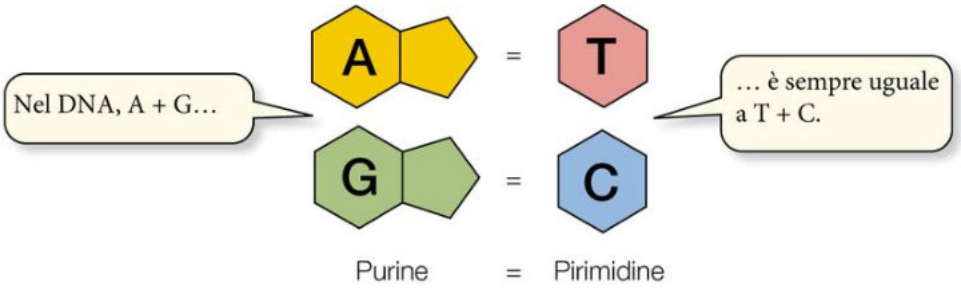


Nel 1950, il chimico di origine austriaca Erwin Chargaff scopre delle regolarità nel DNA.

Riguardano le concentrazioni dei nucleotidi A, C, T, G **nel DNA di cellule di uno stesso individuo.**

La prima regola di parità di Chargaff

La prima regola di parità di Chargaff verifica l'esistenza di un rapporto 1:1 tra la quantità di basi puriniche A+G e quella di basi pirimidiniche T+C contenute nel DNA di una cellula.



Il rapporto tra la percentuale di A e quella di G varia da una specie all'altra ma è invariante per organismi della stessa specie.

La seconda regola di parità di Chargaff

Table 8.1 **Base composition of DNA from different sources and ratios of bases**

Source of DNA	A	T	G	C	Ratio A/T	Ratio G/C	Ratio (A + G)/(T + C)
E. coli	26.0	23.9	24.9	25.2	1.09	0.99	1.04
Yeast	31.3	32.9	18.7	17.1	0.95	1.09	1.00
Sea urchin	32.8	32.1	17.7	18.4	1.02	0.96	1.00
Rat	28.6	28.4	21.4	21.5	1.01	1.00	1.00
Human	30.3	30.3	19.5	19.9	1.00	0.98	0.99

In una molecola di DNA a doppio filamento
la concentrazione di adenina **è uguale a quella di** timina
la concentrazione di citosina **quella di** guanina**.**

E' una legge universale.
E' stata una delle prove usate da **Watson e Crick a sostegno del loro modello di DNA.**

DNA in Wolfram Language

Modello del DNA in Wolfram Language: una stringa di testo con sole 4 lettere A, C, G, T (*nucleotidi*)

I cromosomi umani sono specificati con un numero intero (da 1 a 22) oppure con le lettere "X", "Y"

Importiamo il segmento iniziale del cromosoma 9**, da 1 a** 10^7

```
L = GenomeData[{"9", {1, 10^7}}]
```

```
TAACCCTAACCCTAACCCTAACCCAACCCCACCCCAACCCCAACCCCAACCCAACCCTAACCCTAACCCTAACCCAACCC:
    TAACCCTAACCCTAACCCAACCCTCACCCTCACC …
AGACCTCTGTTTTAATGGTAATGCTGGTCAGTTGTGCCTGAATTCCAAAGGGAGGAGAGTATAATGAAGCATATCCAAT:
    TCTCCCTTCCCATCATGGCCTGAATTAGCTTTTCAG
```

large output | show less | show more | show all | set size limit...

Verifica sperimentale della 2° legge di Chargaff

Possiamo calcolare la frequenza dei 4 nucleotidi sul segmento L del DNA:

```
StringCount[L, "A"]
StringCount[L, "C"]
StringCount[L, "G"]
StringCount[L, "T"]
```

La ricerca della grammatica di Chargaff

In un articolo del 1971
Preface to a Grammar of Biology
Chargaff chiarisce l'obiettivo del suo programma di ricerca:
la scoperta della *grammatica della biologia* **che spieghi le regolarità del DNA.**

L'articolo di Prabhu

Dopo le due leggi del 1950, per fare un passo avanti nel programma di Chargaff, occorre attendere un articolo di Prabhu del 1993

Nucleic Acids Research, 1993, Vol. 21, No. 12 2797–2800

Symmetry observations in long nucleotide sequences

Vinayakumar V.Prabhu
National Center for Biotechnology Information, National Institutes of Health, Building 38-A, 8th Floor, Bethesda, MD 20894, USA

Received May 3, 1993; Accepted May 14, 1993

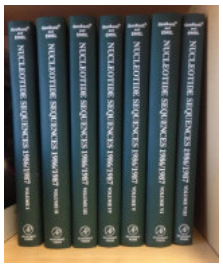
A study of all sequences longer than 50 000 nucleotides currently in GenBank (1, 2) reveals a simple symmetry principle. The number of occurrences of each n-tuple of nucleotides on a given

Table 3 illustrates concisely the symmetry in sets of complementary pairs of 3,4,5,6-tuples in all sequences of Table 1. One notices from Table 3 that for each sequence the correlation

L'articolo di Prabhu (2)

Vinayakumar V. Prabhu comincia così:

A study of all sequences longer than 50 000 nucleotides currently in *GenBank* ***reveals a simple symmetry principle***

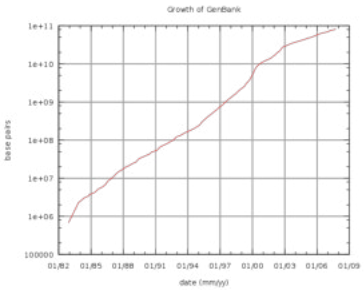


Genbank **è la banca dati del DNA, ad accesso libero, fondata nel 1982 da** Walter Goad **e il** Los Alamos National Laboratory**.**

Alla data del 15 giugno 2018,
GenBank è giunta alla release 226.0
contiene 209 775 348 **loci e** 263 957 884 539 **basi**

La banca dati *Genbank*

Genbank ha indirizzo https://en.wikipedia.org/wiki/GenBank



GenBank riceve sequenze di DNA prodotte nei laboratori di tutto il mondo
La crescita attuale è esponenziale con un tempo di raddoppio pari a 18 mesi circa.
Attualmente contiene il DNA di oltre 100 000 organismi.

Prabhu e il Principio di Simmetria

A study of all sequences longer than 50 000 nucleotides currently in *GenBank* ***reveals a simple symmetry principle.***

Prabhu usando i dati disponibili all'epoca (1993) può congetturare che

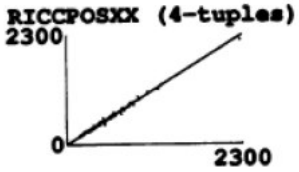
The *number of occurrences* ***of each*** *n-tuple of nucleotides* ***on a given strand approaches that of its*** *complementary n-tuple* ***on the same strand.***

Per ogni parola w **c'è una** *parola simmetrica* w' **con la stessa frequenza.**

L'asserzione generalizza la 2° legge di Chargaff **ed è ora nota come il** Principio di Simmetria

Prabhu e il Principio di Simmetria

This symmetry is true for all long sequences at small n
(e. g. n = 1, 2, 3, 4, 5)
It extends to sets of n-tuples of higher order n
with increase in the length of the sequence.
Each dot represents a complementary tuple pair and has for (X, Y) coordinates, the number of occurrences of (tuple, complementary tuple).



The dots agglutinate around the line of unit slope showing that the number of occurrences of each tuple approaches that of its complementary tuple on the same strand of the sequence.

Frequenza di parole di lunghezza qualsiasi

Definiamo la frequenza relativa di una parola del DNA:

```
F[x_] := N@StringCount[L, x] / StringLength[L]
```

Definiamo una relazione di equivalenza sulle parole a meno di un piccolo ϵ

$$x \equiv y \quad \text{se} \quad F[x] - F[y] < \epsilon$$

L'idea è di lavorare sull'insieme W_2 dei dinucleotidi (parole di lunghezza 2) per determinare, fissato $\epsilon = 10^{-3}$, le coppie di parole equivalenti.

Liste su un insieme.

Dato un *L*, detto alfabeto, vogliamo calcolare l'insieme di tutte le coppie ordinate {*x, y*}, di elementi *x, y* ∈ *L*.
Ad esempio, sia *L* = {*a, b, c, d*}

```
L = {a, b, c, d}
```

```
{a, b, c, d}
```

allora L^2 = *L* x *L* si ottiene con Tuples [*L*, 2]

```
Tuples[L, 2]
```

mentre l'insieme L^3 di tutte le terne ordinate si ottiene con

```
Tuples[L, 3]
```

Esercizio. **Calcolare l'insieme delle coppie ordinate di nucleotidi e l'insieme delle terne ordinate di nucleotidi.**

Parole su un insieme

Sia dato un insieme di caratteri *L* = {*a, b, c, d*}, detto *alfabeto*,

```
L = {"a", "b", "c", "d"}
```

{a, b, c, d}

vediamo come si costruisce l'insieme di tutte le parole di lunghezza 2 con caratteri in *L* .
Costruiamo dapprima le liste di lunghezza 2 su *L* :

```
Tuples[L, 2]
```

{{a, a}, {a, b}, {a, c}, {a, d}, {b, a}, {b, b}, {b, c}, {b, d}, {c, a}, {c, b}, {c, c}, {c, d}, {d, a}, {d, b}, {d, c}, {d, d}}

```
StringJoin["a", "b"]
```

ab

```
Table[StringJoin[x, y], {x, L}, {y, L}]
```

{{aa, ab, ac, ad}, {ba, bb, bc, bd}, {ca, cb, cc, cd}, {da, db, dc, dd}}

Scoprire il principio di simmetria nei dinucleotidi

```
W2 = StringJoin /@ Tuples[{"A", "C", "G", "T"}, 2]
```

Ordiniamo l'insieme W_2 secondo la frequenza:

```
SW2 = SortBy[W2 , F]
```

Verifichiamo le frequenzze dei 16 dinucleotidi:

```
Table[{x, F[x]}, {x, SW2}]
```

```
{{CG, 0.0082755}, {CC, 0.0382121}, {GG, 0.0382146}, {GC, 0.0393492}, {GT, 0.050517},
 {AC, 0.0506259}, {GA, 0.0589243}, {TC, 0.0590924}, {AG, 0.0689246}, {CT, 0.0692205},
 {TA, 0.0701253}, {CA, 0.0715715}, {TG, 0.0715905}, {TT, 0.073816}, {AA, 0.0739151},
 {AT, 0.0810706}}
```

Scoprire il principio di simmetria nei dinucleotidi (parte 2)

Parole equivalenti **dovrebbero essere** consecutive **in SW_2 ordinato per frequenza.**
Calcoliamo l'insieme di coppie di dinucleotidi consecutivi in SW_2

```
Consecutivi = Partition[SW2 , 2, 1]
```

Definiamo sull'insieme W_2 l'equivalenza a meno $\epsilon = 10^{-3}$

```
ϵ = 10^-3; uguali[{x_, y_}] := |F[x] - F[y]| < ϵ
```

In questo insieme di ricerca selezioniamo le coppie di parole uguali:

```
Uguali = Select[Consecutivi, uguali]
```

Scoprire il principio di simmetria nei dinucleotidi (parte 3)

Identifichiamo lettere complementari con una regola di sostituzione:

```
regole = {"A" -> "■", "T" -> "■", "G" -> "■", "C" -> "■"};
id[x_] := StringReplace[x, regole]
Map[id, Uguali]
```

Nota. **Le coppie di parole** uguali **sono quasi sempre** "simmetriche"
Esiste un'eccezione alla simmetria: la coppia `{"CT","TA"}`.
"CT" e **"TA" hanno frequenza quasi identica ma non simmetrici.**

La prima componente della simmetria: complementare

Definiamo dapprima il complementare delle lettere:

$$\mathcal{C}(\text{"A"}) = \text{"T"},\ \mathcal{C}(\text{"T"}) = \text{"A"},\ \mathcal{C}(\text{"C"}) = \text{"G"},\ \mathcal{C}(\text{"G"}) = \text{"C"}$$

La parola *complementare* **di** $w = a_1 a_2 \ldots a_k$ **è** $\mathcal{C}(w) = c_1 c_2 \ldots c_k$

dove $c_1 = \mathcal{C}(a_1),\ c_2 = \mathcal{C}(a_2),\ \ldots,\ c_k = \mathcal{C}(a_k)$

```
Unprotect[𝒞]; r = {"A" → "T", "T" → "A", "G" → "C", "C" → "G"};
𝒞[w_] := StringReplace[w, r]
𝒞 ["CTATGA"]
```

La seconda componente della simmetria: inversione

La parola *inversa* **di una parola** $w = a_1 a_2 \ldots a_k$ **è**

$$\mathcal{R}(w) = a_k a_{k-1} \ldots a_1$$

```
ℛ[w_] := StringReverse [w]
```

Esempio. **Parola inversa di** "CTATGA"

```
ℛ["CTATGA"]
```

Verifica sperimentale del principio di simmetria

Principio di simmetria

per ogni parola w **si ha** $\mathbb{F}(\mathscr{C}(\mathscr{R}(w))) \approx \mathbb{F}(w)$

Verifichiamo il principio di simmetria **nell'insieme dei dinucleotidi:**

per ogni dinucleotide w **si ha** $\mathbb{F}(\mathscr{C}(\mathscr{R}(w))) \approx \mathbb{F}(w)$

```
Table[|F[𝒞[ℛ[x]]] - F[x]| < 10^(-3), {x, W2}]
```

Un corollario del principio di simmetria

Per ogni parola w **si ha:** $\mathbf{F(\mathcal{C}(w)) \approx F(\mathcal{R}(w))}$

```
F[𝒞 ["CTATGA"]]
```

```
F[ℛ ["CTATGA"]]
```

Verifica sperimentale:

```
Table[|F[ℛ [x]] - F[𝒞 [x]]| < 10^(-3), {x, W2}]
```

Dimostrazione.

Se $u = \mathcal{R}(w)$ **allora** $\mathcal{R}(u) = \mathcal{R}(\mathcal{R}(w)) = w$

quindi $F(\mathcal{C}(w)) \approx F(\mathcal{C}(\mathcal{R}(u))) = F(u) \approx F(\mathcal{R}(w))$

Problema: come generalizzare la 1° legge di Chargaff?

Prabhu generalizza la 2° legge di Chargaff nel 1993
estende la simmetria dalle lettere A, C, T, G **alle parole di lunghezza >1**

$$\mathbb{F}(A) + \mathbb{F}(G) \approx \mathbb{F}(T) + \mathbb{F}(C)$$

$$\mathbb{F}(A) + \mathbb{F}(C) \approx \mathbb{F}(T) + \mathbb{F}(G)$$

Il metodo usato è puramente matematico