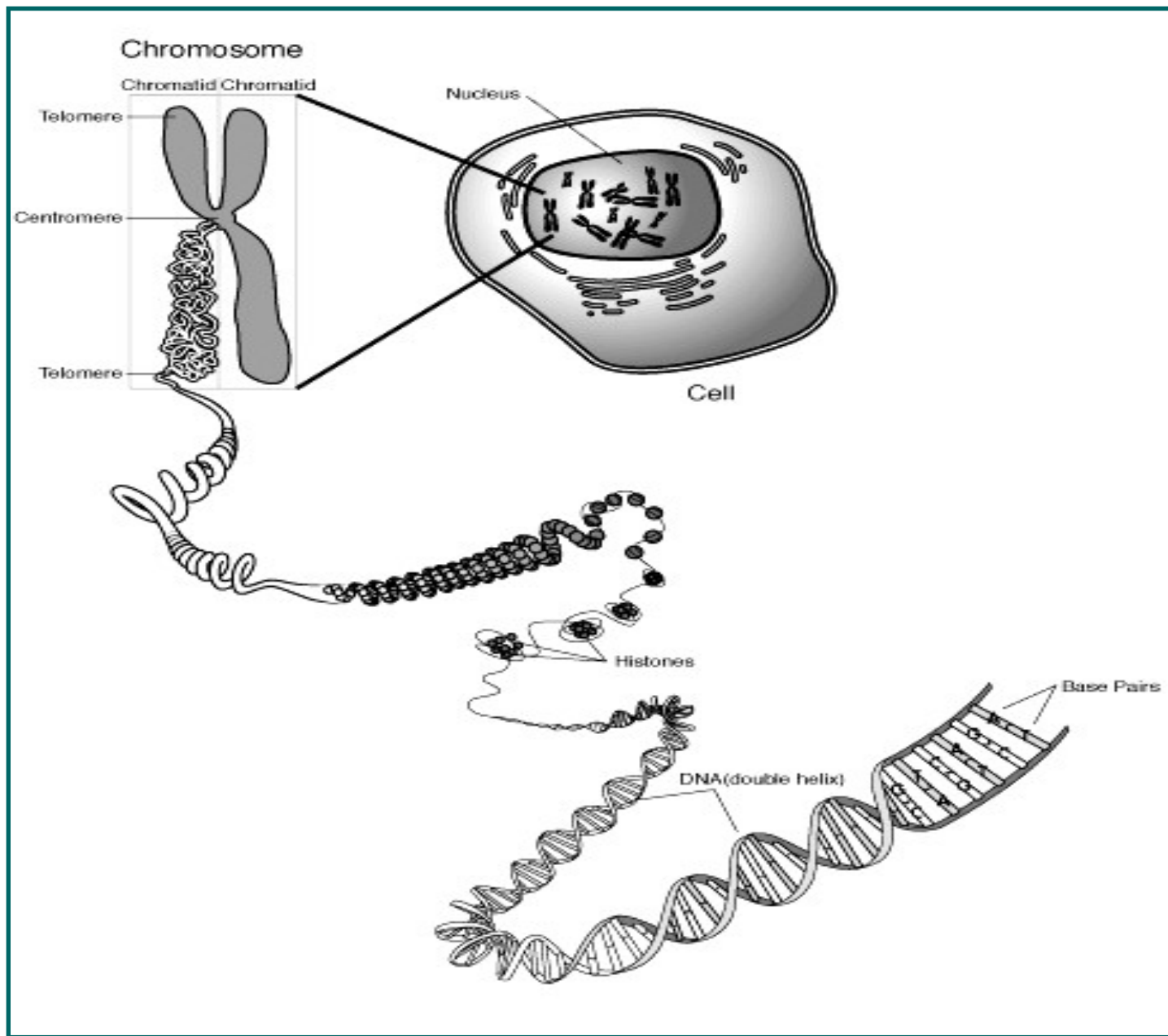
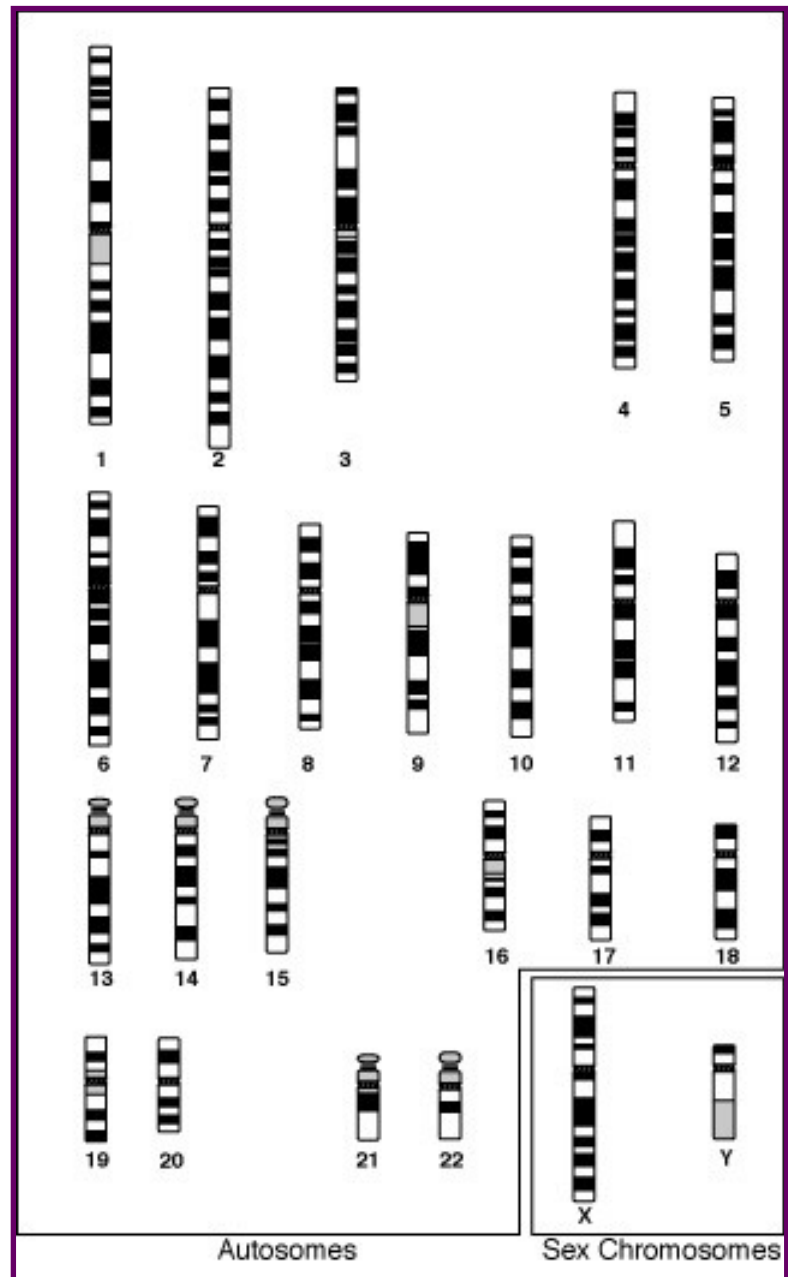


# **ALINEAMIENTO DE**

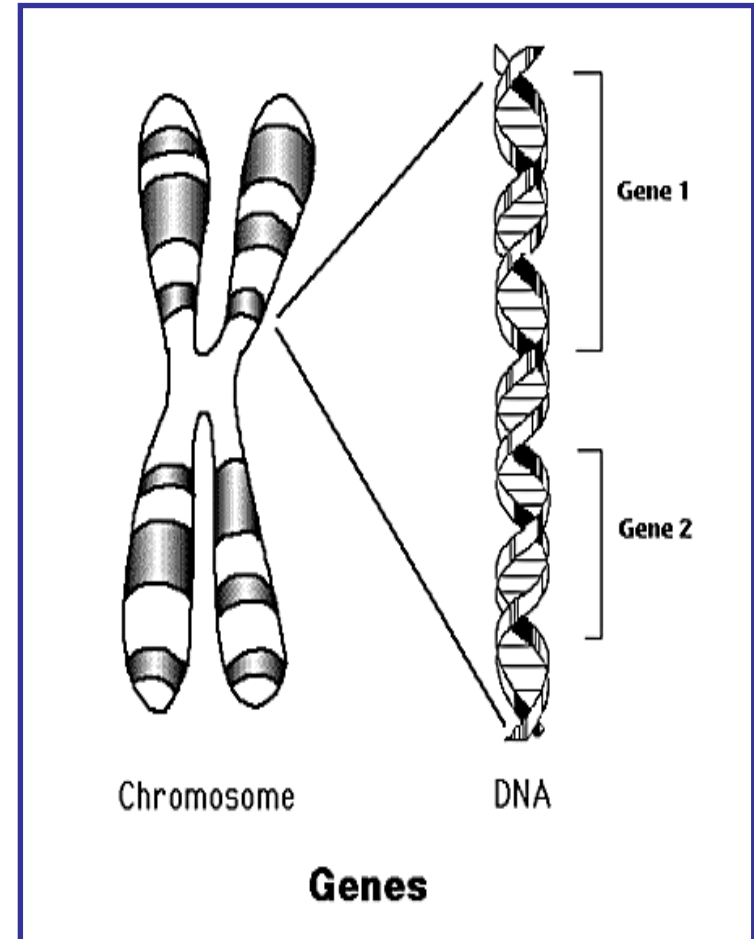
# **SECUENCIAS**



# CROMOSOMAS HUMANOS



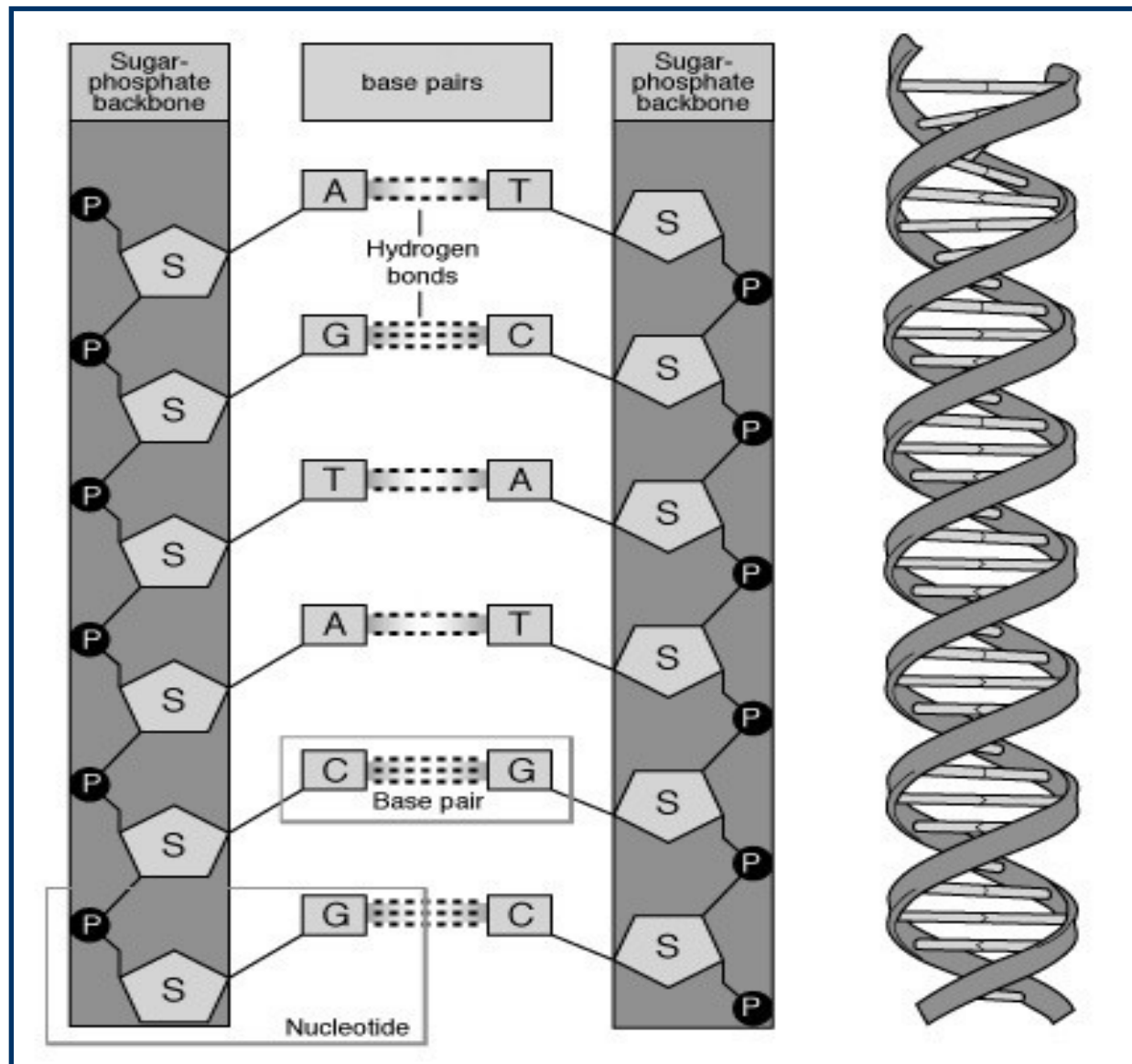
- **Gen** – unidad básica de información genética. Determinan la información para codificar una proteína.
- **Genoma** – colección de toda la información genética de un individuo.
- **Cromosomas** – Moléculas lineales de ADN que contienen genes y material no codificante.



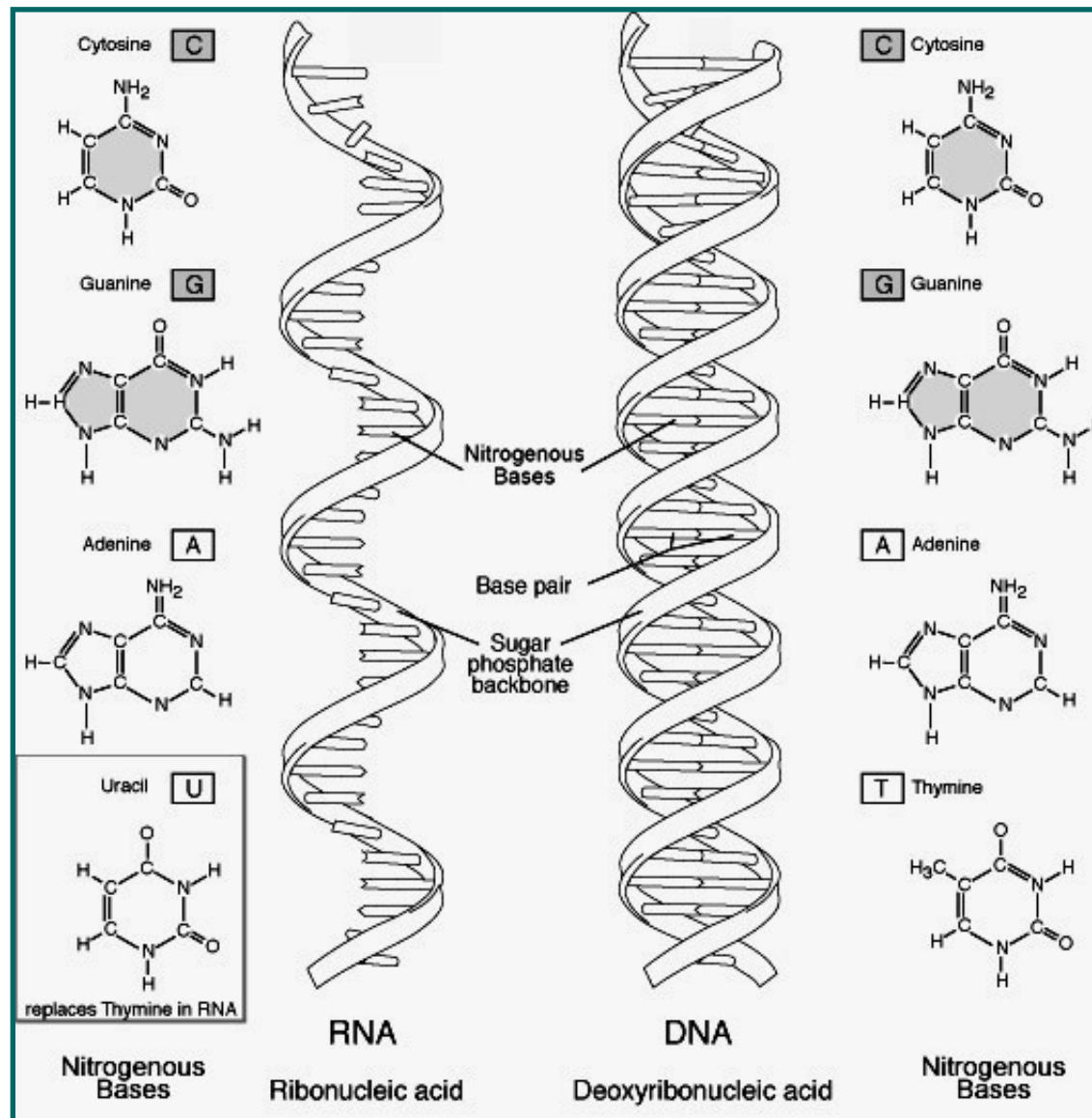
Tamaño del Genoma Humano

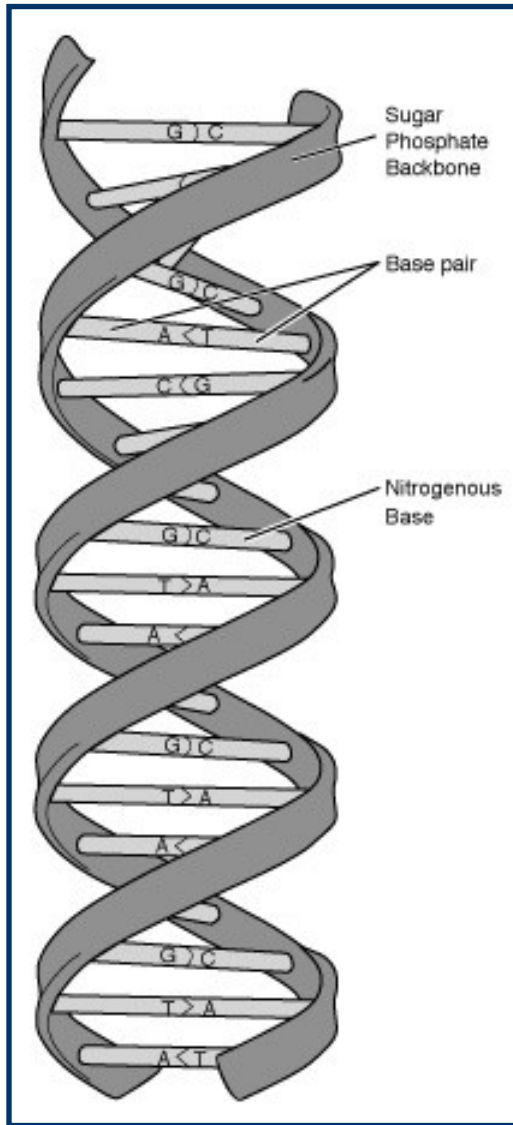
$3 \times 10^9$  bases

# ADN



# ADN / ARN

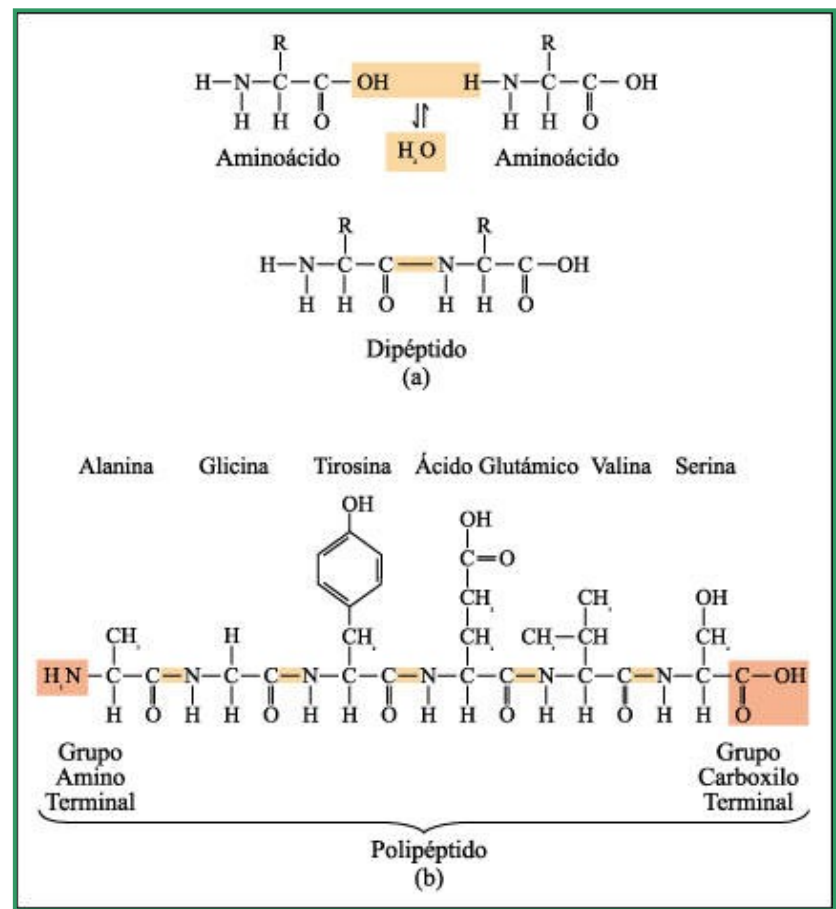
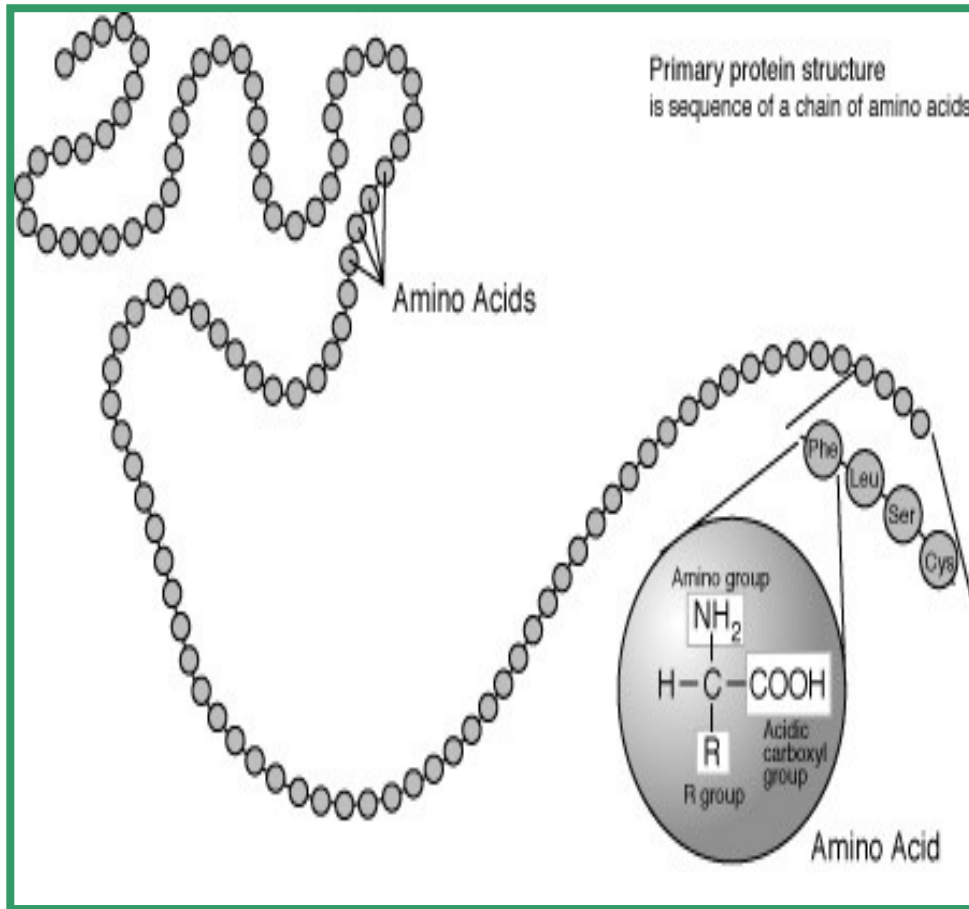




Simplificación

GGACGTATGTCAGTA

# PROTEÍNAS



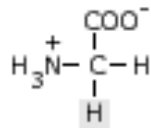


<b>Name</b>	<b>1-Letter Nickname</b>	<b>3-Letter Nickname</b>
<b>Glycine</b>	G	Gly
<b>Alanine</b>	A	Ala
<b>Valine</b>	V	Val
<b>Leucine</b>	L	Leu
<b>Isoleucine</b>	I	Ileu
<b>Serine</b>	S	Ser
<b>Threonine</b>	T	Thr
<b>Cysteine</b>	C	Cys
<b>Methionine</b>	M	Met
<b>Glutamic Acid</b>	E	Glu
<b>Aspartic Acid</b>	D	Asp
<b>Lysine</b>	K	Lys
<b>Arginine</b>	R	Arg
<b>Asparagine</b>	N	Asn
<b>Glutamine</b>	Q	Gln
<b>Phenylalanine</b>	F	Phe
<b>Tyrosine</b>	Y	Tyr
<b>Tryptophan</b>	W	Trp
<b>Unknown</b>	X	xxx
<b>Proline</b>	P	Pro
<b>Terminator</b>	*	End

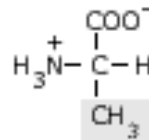
## Grupos de aminoácidos según sus propiedades químicas

- Polares con carga neta positiva: H, R, K
- Polares con carga neta negativa: E, D
- Polares sin carga neta: Q, N, S, T
- No polares: A, G, V, L, I, P, C, M
- Aromáticos: W, F, Y

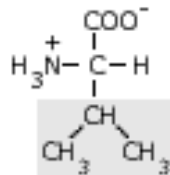
### Nonpolar, alphabetical R groups



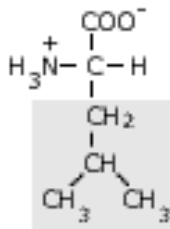
**Glycine**



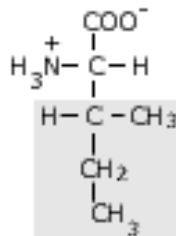
**Alanine**



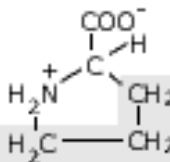
**Valine**



**Leucine**

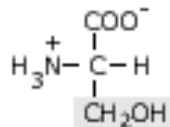


**Isoleucine**

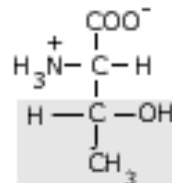


**Proline**

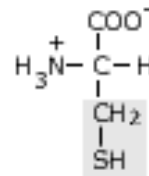
### Polar, uncharged R groups



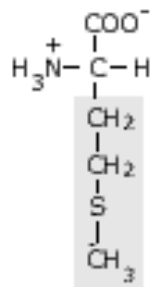
**Serine**



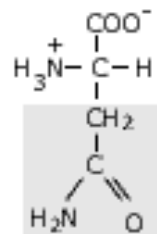
**Threonine**



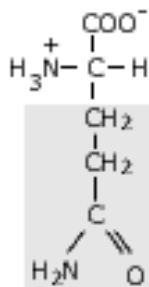
**Cysteine**



**Methionine**

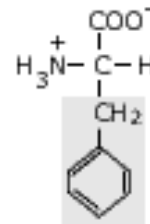


**Asparagine**

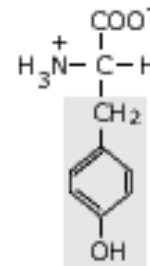


**Glutamine**

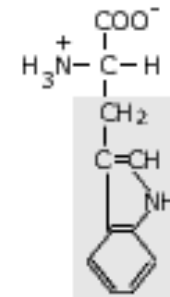
### Aromatic R-groups



**Phenylalanine**

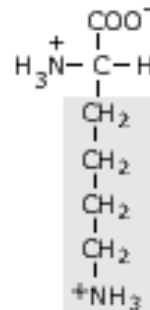


**Tyrosine**

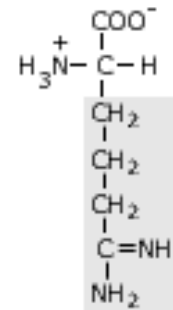


**Tryptophan**

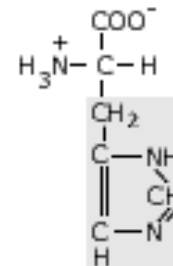
### Positively charged R groups



**Lysine**

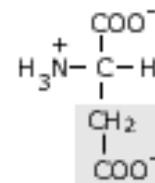


**Arginine**

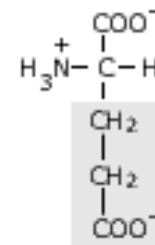


**Histidine**

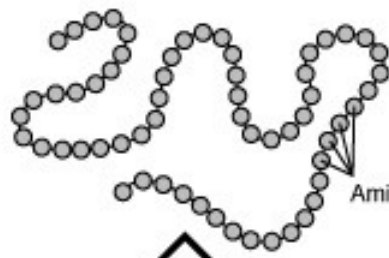
### Negatively charged R groups



**Aspartate**

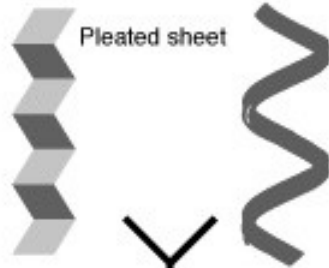


**Glutamate**



**Primary protein structure**  
is sequence of a chain of amino acids

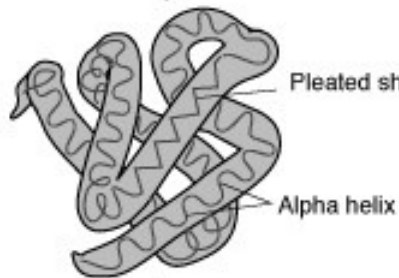
Amino Acids



Pleated sheet

Alpha helix

**Secondary protein structure**  
occurs when the sequence of amino acids  
are linked by hydrogen bonds



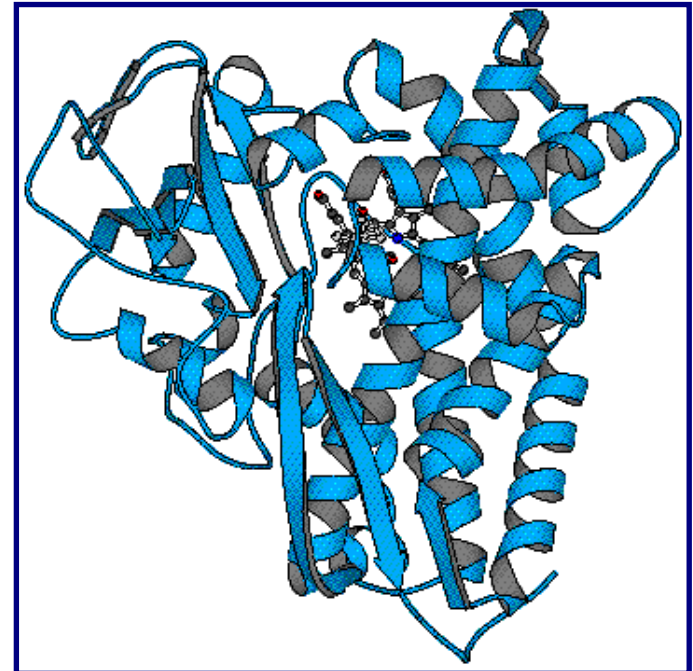
Pleated sheet

Alpha helix

**Tertiary protein structure**  
occurs when certain attractions are present  
between alpha helices and pleated sheets.

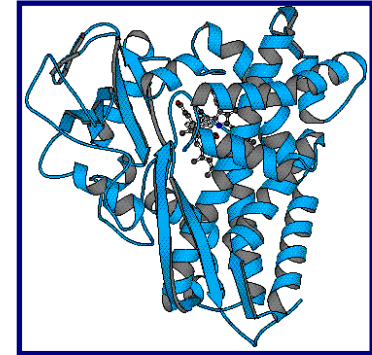


**Quaternary protein structure**  
is a protein consisting of more than one  
amino acid chain.



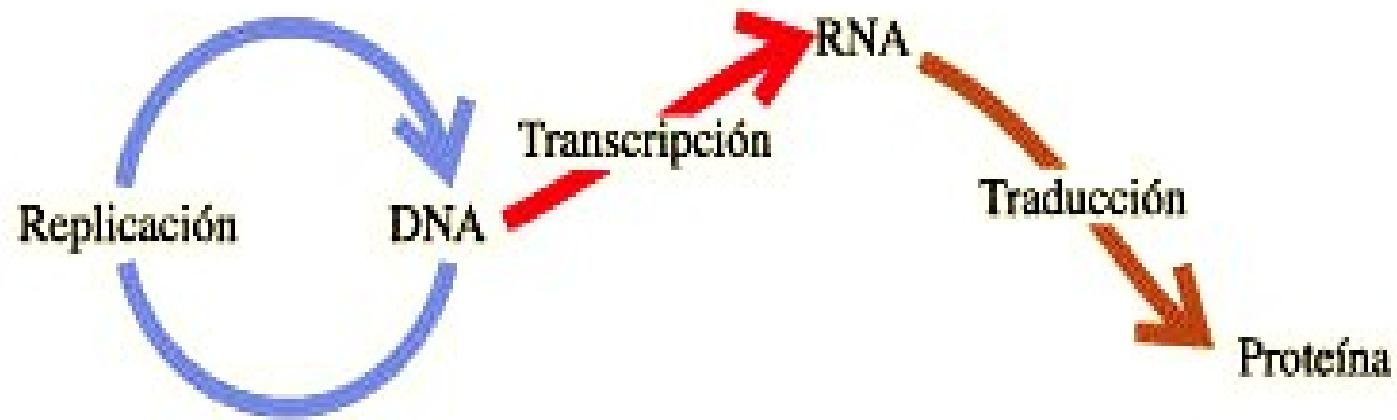
**Dominios  
en las proteínas**

# Simplificación de una proteína

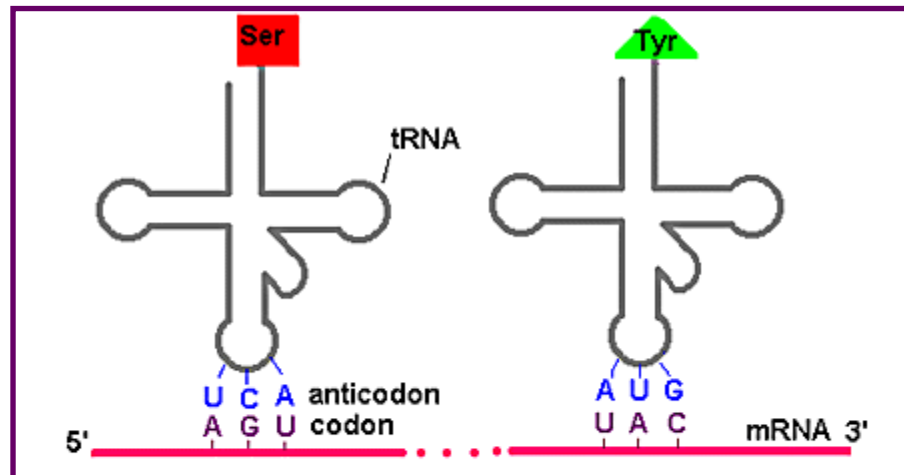


MEELQDDYEDMMEENLEQEEYEDPDIPESQMEE  
PAAHDTEATATDYHTTSHPGTHKVYVELQELVM  
DEKNQELRWMEAARWVQLEENLGENGAWGRP  
HLSHLTFWSLLELRRVFTKGTVLLDLQETSLAGV  
ANQLLDRFIFEDQIRPQDREELLRALLLKHSHAGE  
LEALGGVKPAVLTRDPSQPLLPQHSSLETQLFCE  
QGDGGTEGHSPSGILEKIPPDSEATLVLVGRADFL  
EQPVLGFVRLQEAAELEAELPVPIRFLFVLLGPEA  
PHIDYTQLGRAAATLMSESVFRIDAYM

## FLUJO DE LA INFORMACIÓN

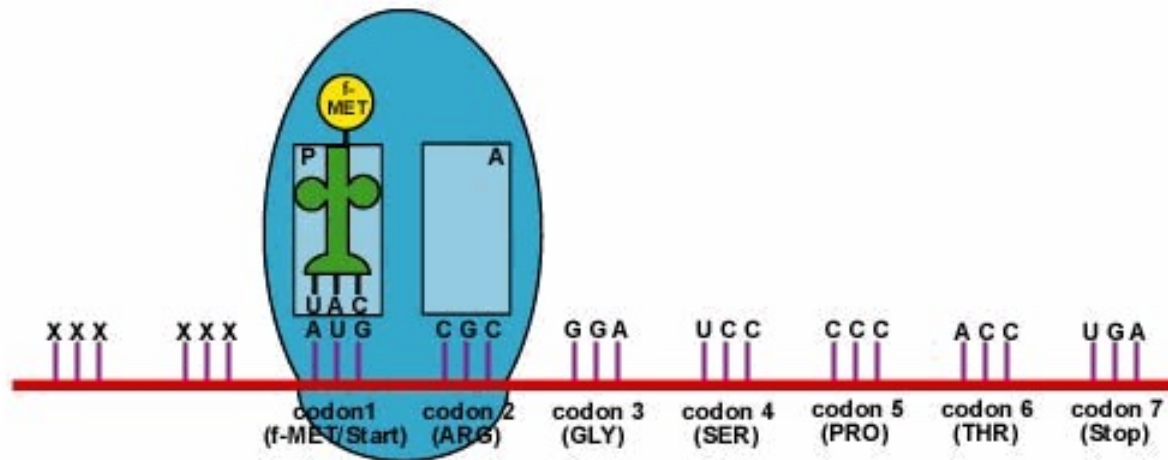


# Traducción



		2nd base in codon					
		U	C	A	G		
1st base in codon	U	Phe Phe Leu Leu	Ser Ser Ser Ser	Tyr Tyr STOP STOP	Cys Cys STOP Trp	U C A G	3rd base in codon
	C	Leu Leu Leu Leu	Pro Pro Pro Pro	His His Gln Gln	Arg Arg Arg Arg	U C A G	
	A	Ile Ile Ile Met	Thr Thr Thr Thr	Asn Asn Lys Lys	Ser Ser Arg Arg	U C A G	
	G	Val Val Val Val	Ala Ala Ala Ala	Asp Asp Glu Glu	Gly Gly Gly Gly	U C A G	

**The Genetic Code**





# MUTACIONES Y EVOLUCIÓN DE LA INFORMACIÓN

## 1. En el ADN

A → G

G → A

C → T

T → C

} Transiciones

A → T

T → G

T → A

G → T

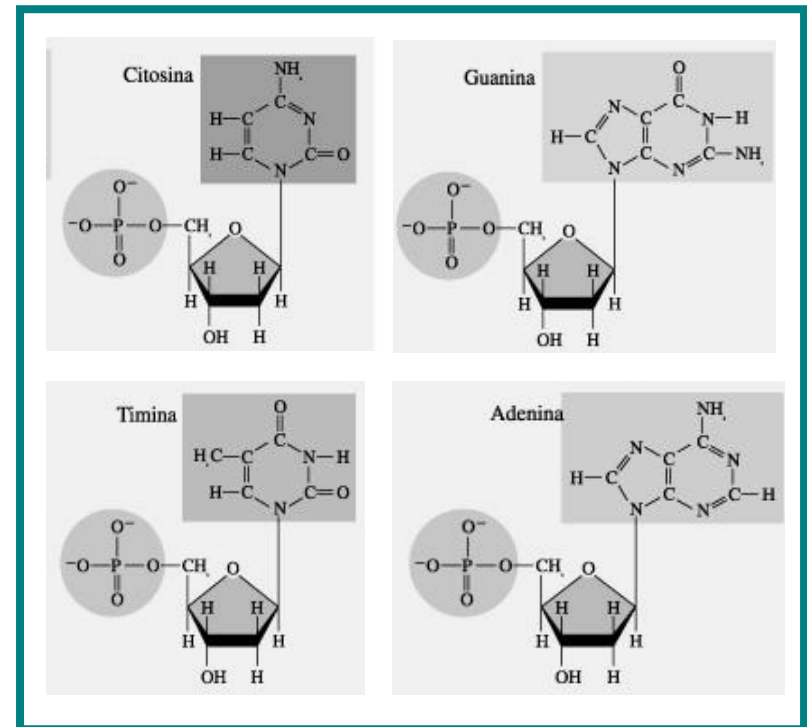
A → C

C → G

C → A

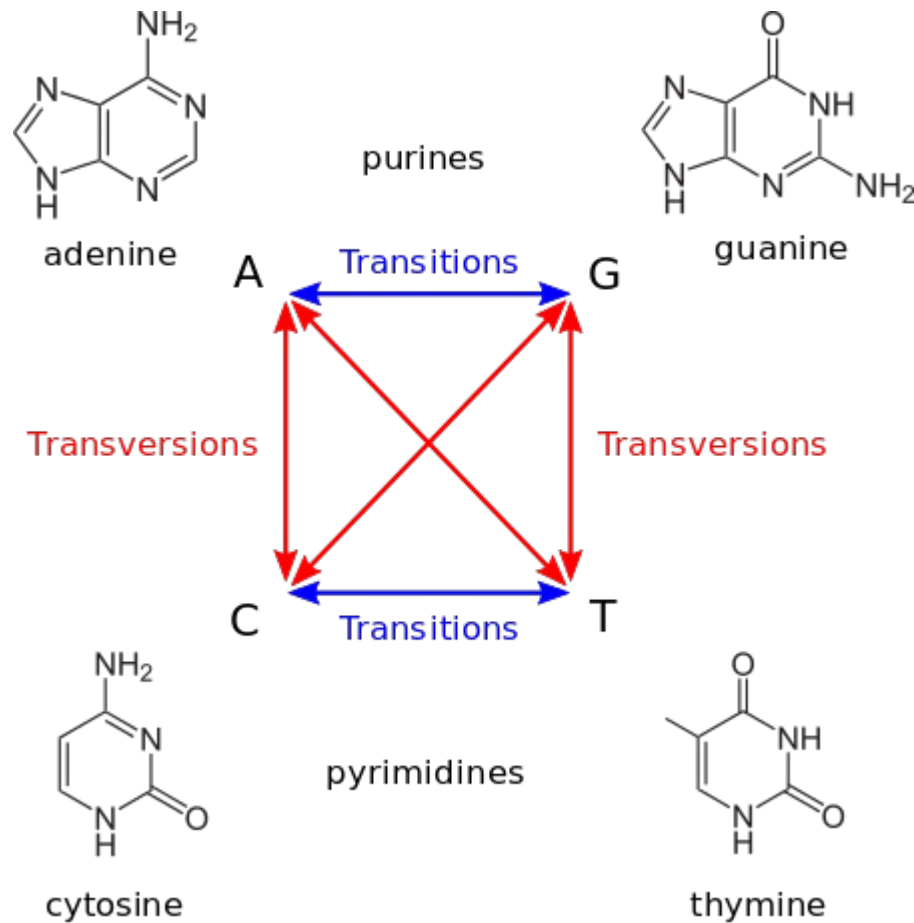
G → C

} Transversiones



# MUTACIONES Y EVOLUCIÓN DE LA INFORMACIÓN

## 1. En el ADN



## 2. En proteínas

- Silenciosa → tripletes que cifran el mismo aa
- Neutra → aa distintos equivalentes funcionalmente
- Cambio de sentido → aa distintos no equivalentes
- Sin sentido → cambio codifica para codón STOP
- Adición o deleción de un pb → cambia el marco de lectura. (no x3)

Missense mutation					
	L	Q	T	←	protein seq.
normal:	ctg	cag	act	←	nucleotide seq.
		*		←	mutation
mutated:	ctg	cgg	act		
	L	R	T		

Silent mutation					
	L	Q	T	←	protein seq.
normal:	ctg	cag	act	←	nucleotide seq.
		*		←	mutation
mutated:	ctg	caa	act		
	L	Q	T		

Nonsense mutation					
	L	Q	T	←	protein seq.
normal:	ctg	cag	act	←	nucleotide seq.
		*		←	mutation
mutated:	ctg	tag	act		
	L	***			

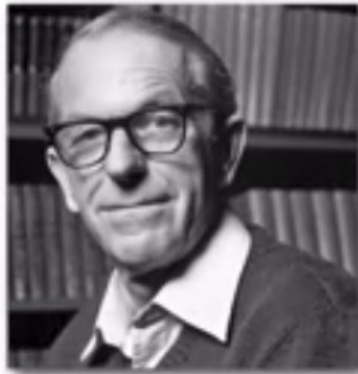
Frameshift mutation									
	L	Q	T	F	S	G	←	protein seq.	
normal:	ctg	cag	act	ttt	agt	gga	←	nucleotide seq.	
		*					←	mutation	
mutated:	ctg	aga	ctt	tta	gtg	ga.			
	L	R	L	L	V				

# ANÁLISIS GENÓMICO:

→ Identificar y caracterizar genes

- Cuántos hay?
- Cómo funcionan?
- Variedad protéica?

## First generation DNA sequencing



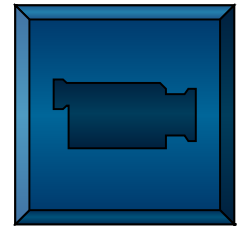
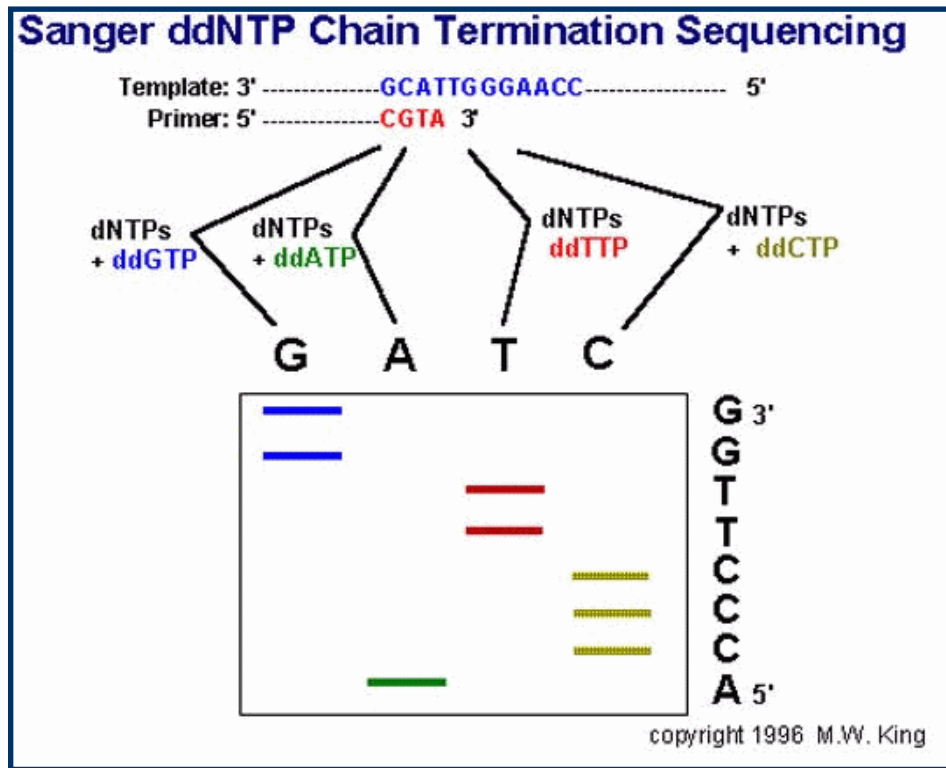
Fred Sanger

"Chain termination" sequencing



# SECUENCIAMIENTO

## Orden de los nucleótidos en un fragmento de ADN

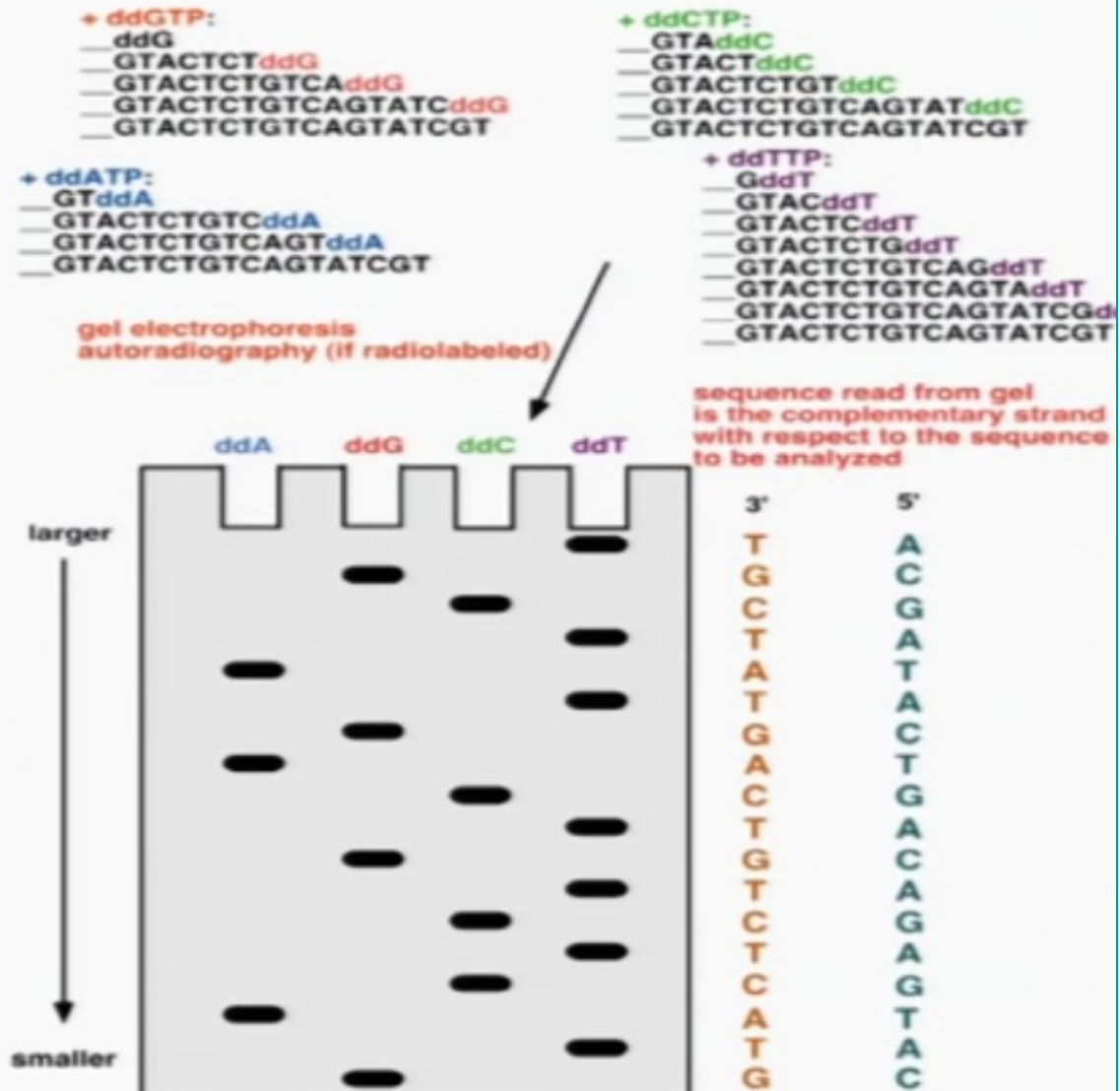


Dideoxynucleotides are chain-elongating inhibitors of [DNA polymerase](#), used in the [Sanger method for DNA sequencing](#). They are also known as 2',3' dideoxynucleotides, and abbreviated as ddNTPs (ddGTP, ddATP, ddTTP and ddCTP).

The absence of the 3'-hydroxyl group means that, after being added by a DNA polymerase to a growing nucleotide chain, no further nucleotides can be added as no [phosphodiester bond](#) can be created

Primer  
5' NNN  
3' NNNCATGAGACAGTC...  
Template

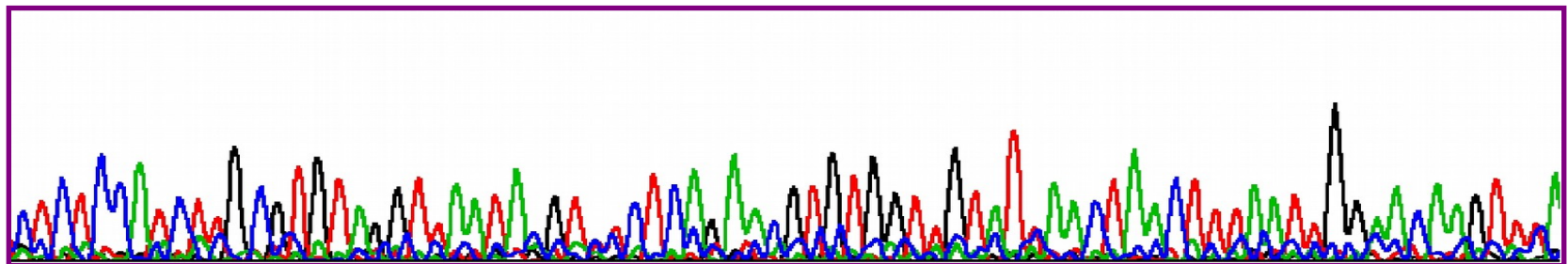
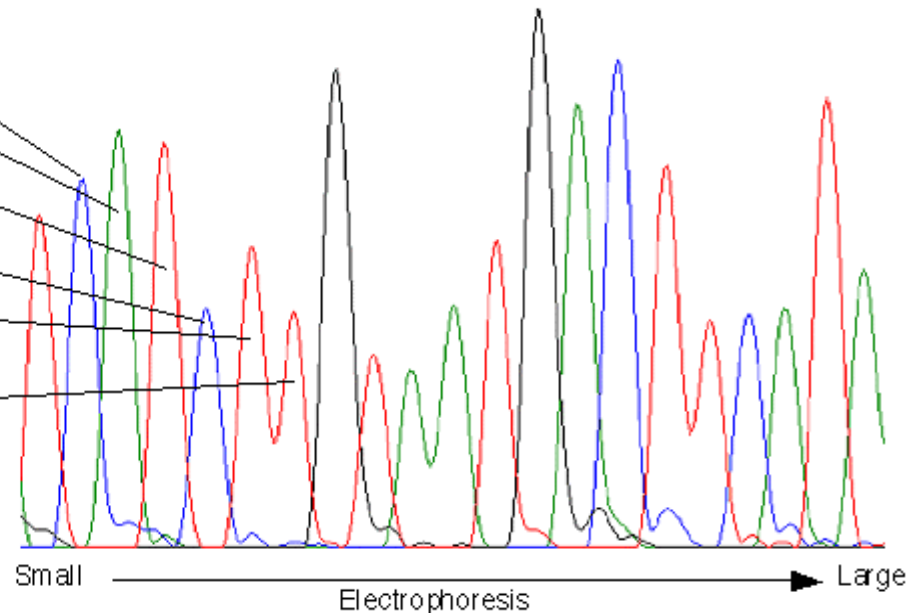
# Sanger sequencing method



# Secuenciamiento - Producto final

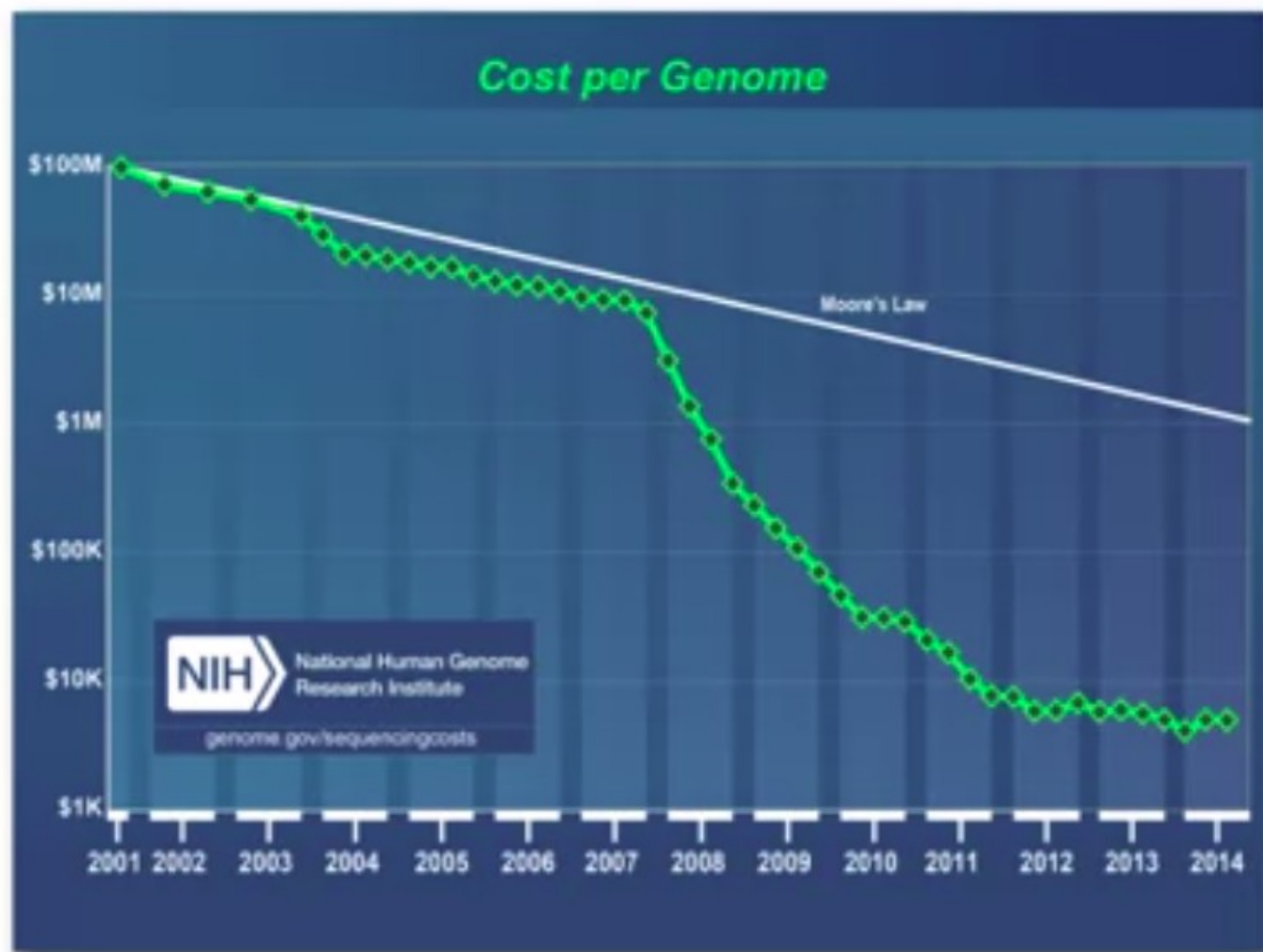


More typically now, sequencing reactions are denatured and the products are separated in a single gel lane or a single capillary tube. The products of the four reactions are labeled with a different fluorescent dye, and a single detector at the bottom of the apparatus detects the fluors as they emerge. The sequence can be read (automatically) from left to right.





## Second generation and beyond



### Input DNA

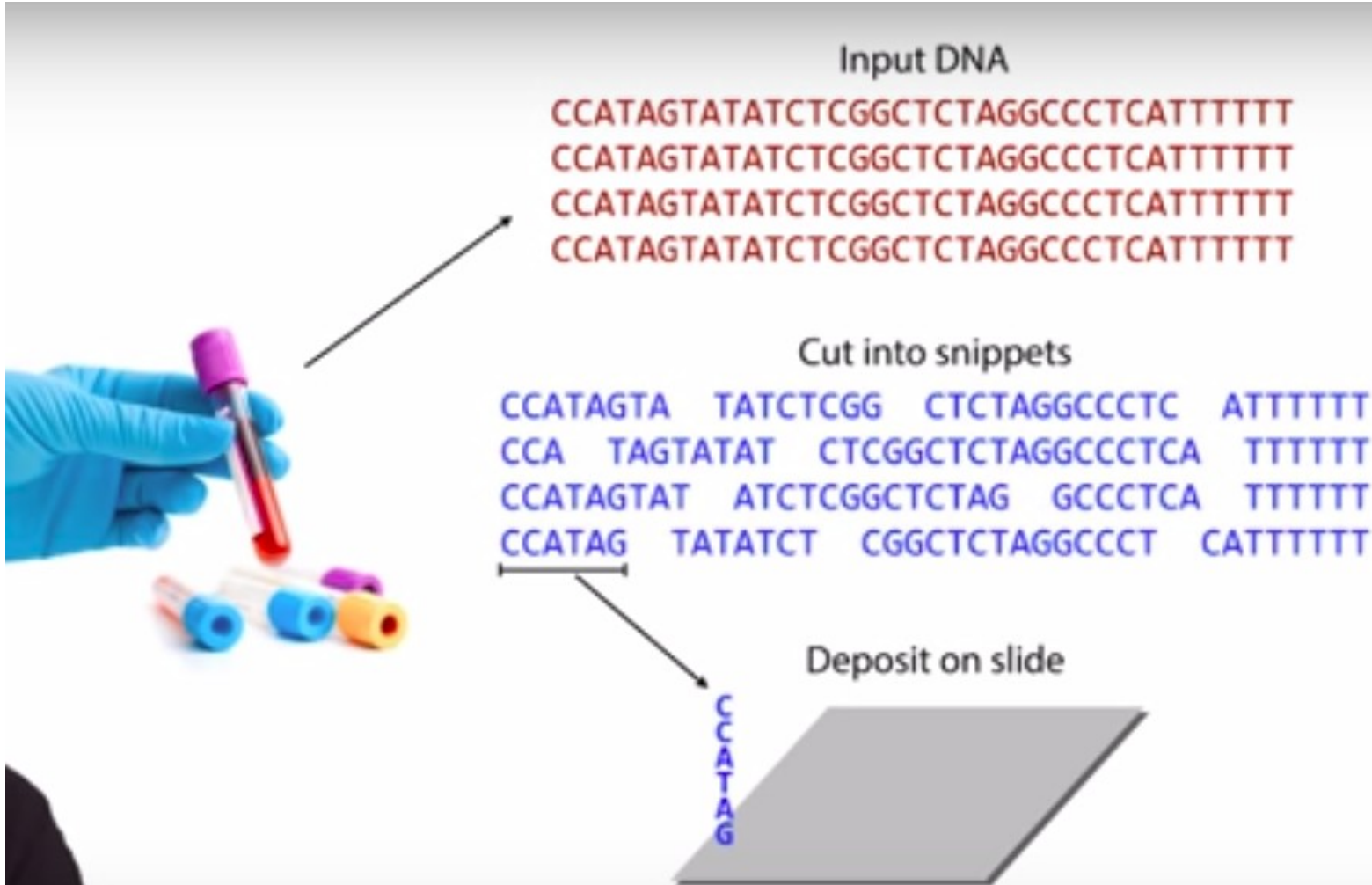
CCATAGTATATCTCGGCTCTAGGCCCTCATT  
CCATAGTATATCTCGGCTCTAGGCCCTCATT  
CCATAGTATATCTCGGCTCTAGGCCCTCATT  
CCATAGTATATCTCGGCTCTAGGCCCTCATT

### Cut into snippets

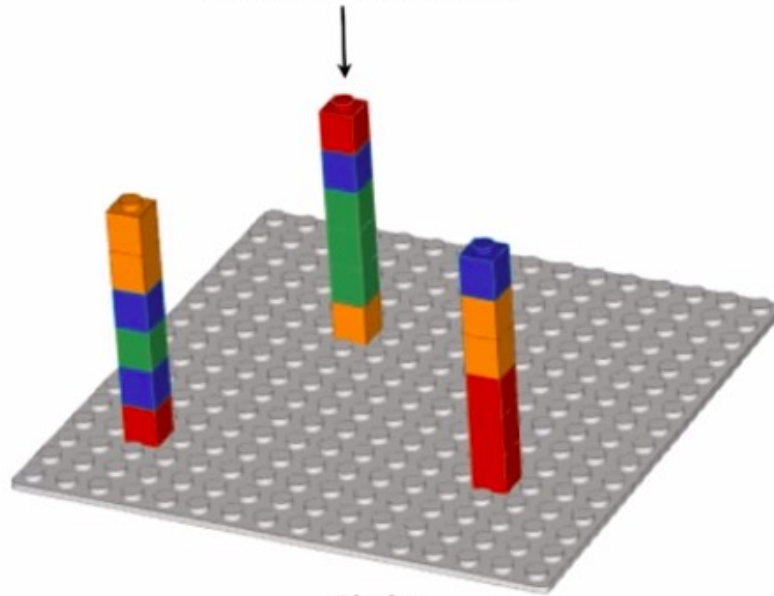
CCATAGTA TATCTCGG CTCTAGGCCCTC ATTTTTT  
CCA TAGTATAT CTCGGCTCTAGGCCCTCA TTTTTT  
CCATAGTAT ATCTCGGCTCTAG GCCCTCA TTTTTT  
CCATAG TATATCT CGGCTCTAGGCCCT CATTTTTT

### Deposit on slide

C  
C  
A  
T  
A  
G



Template  
(billions of them!)



Slide


A




T



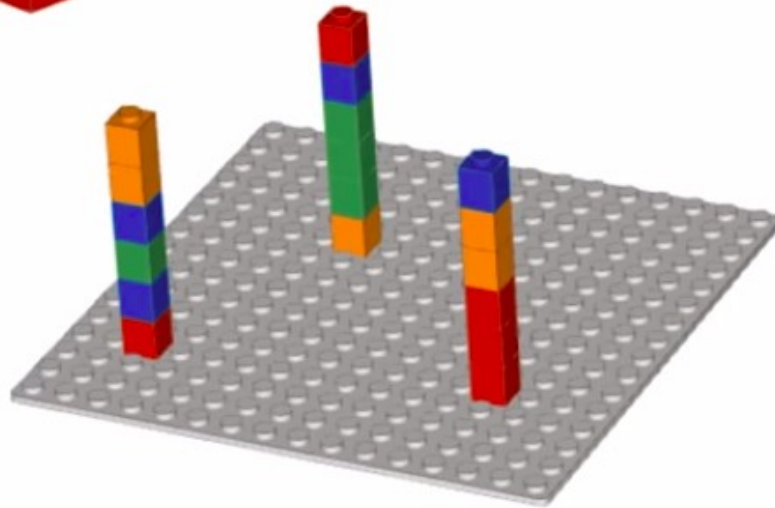
C

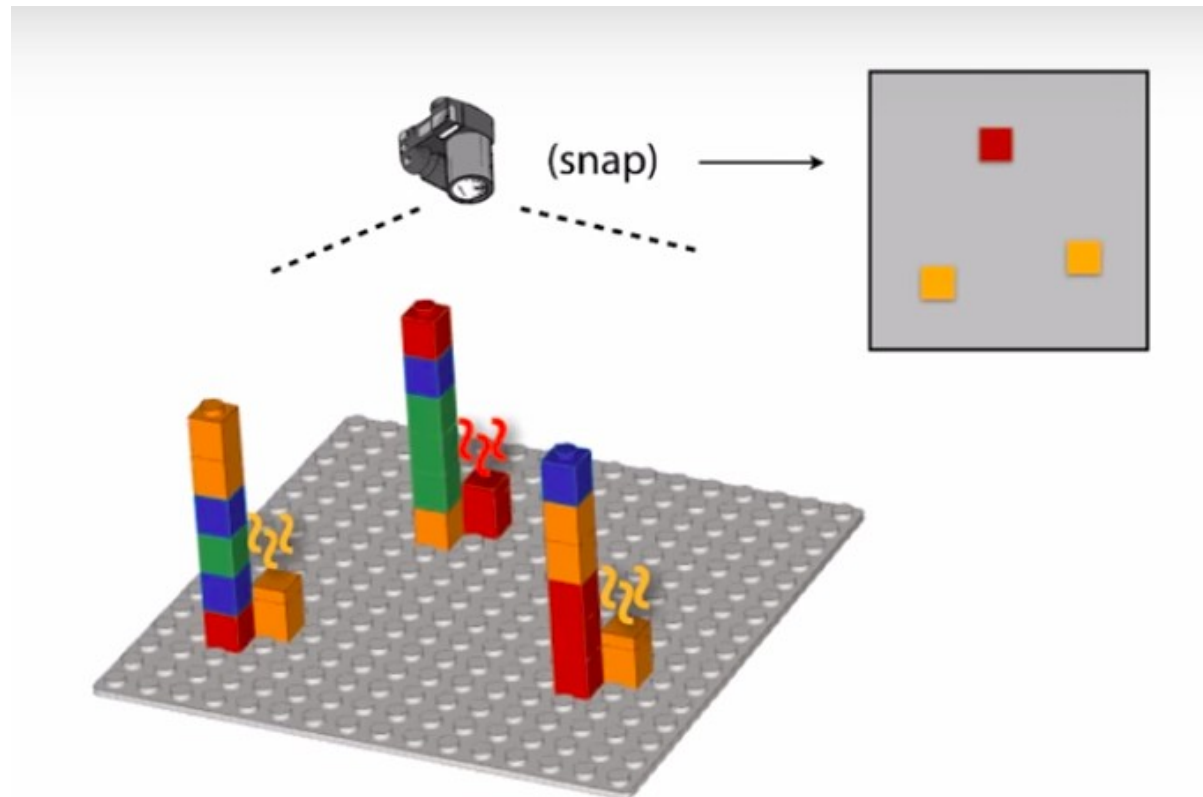


G

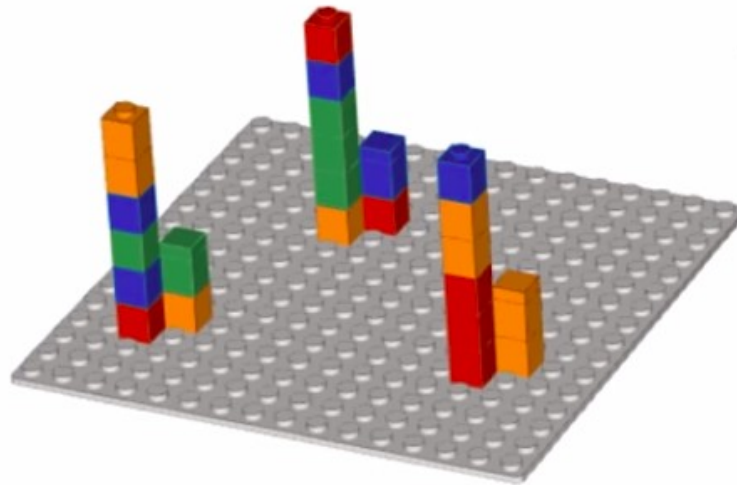
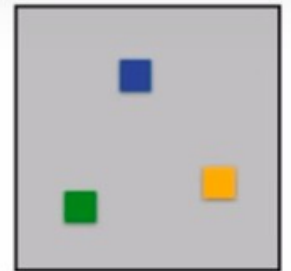
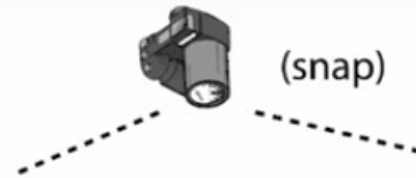
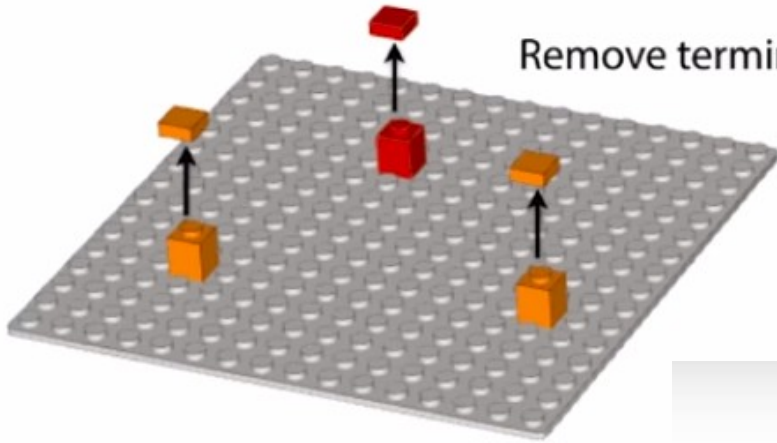


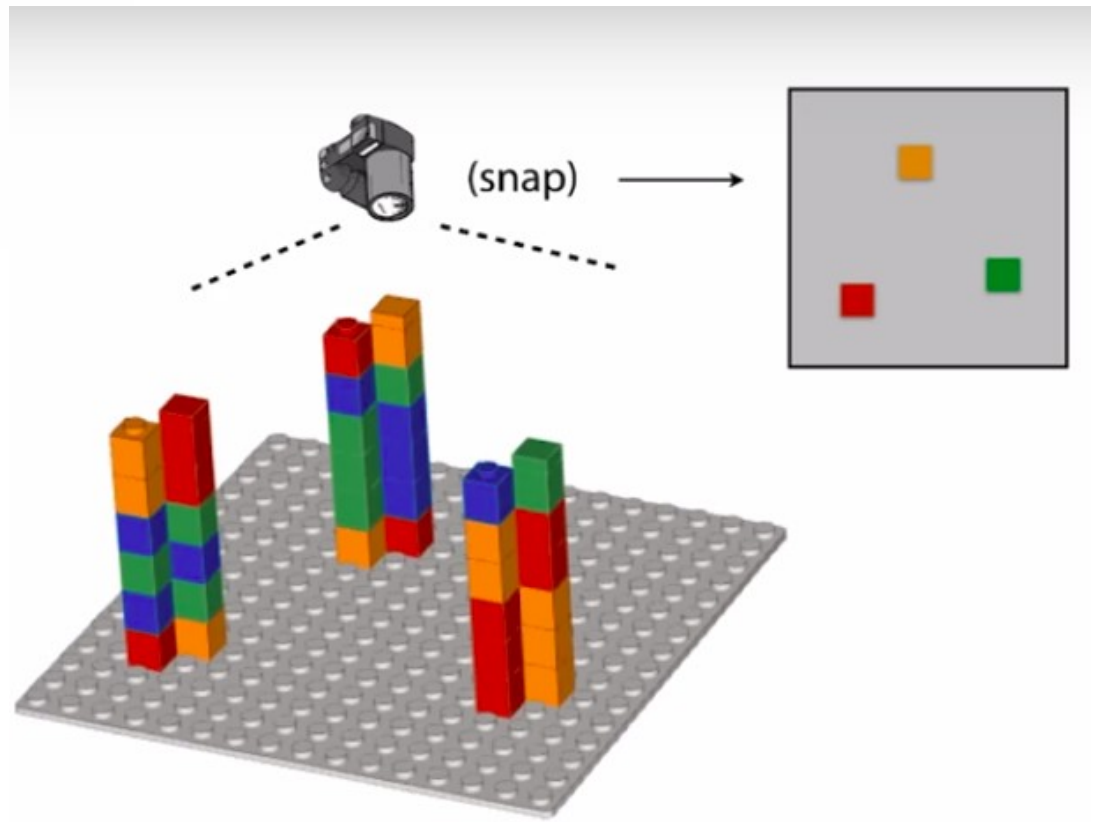
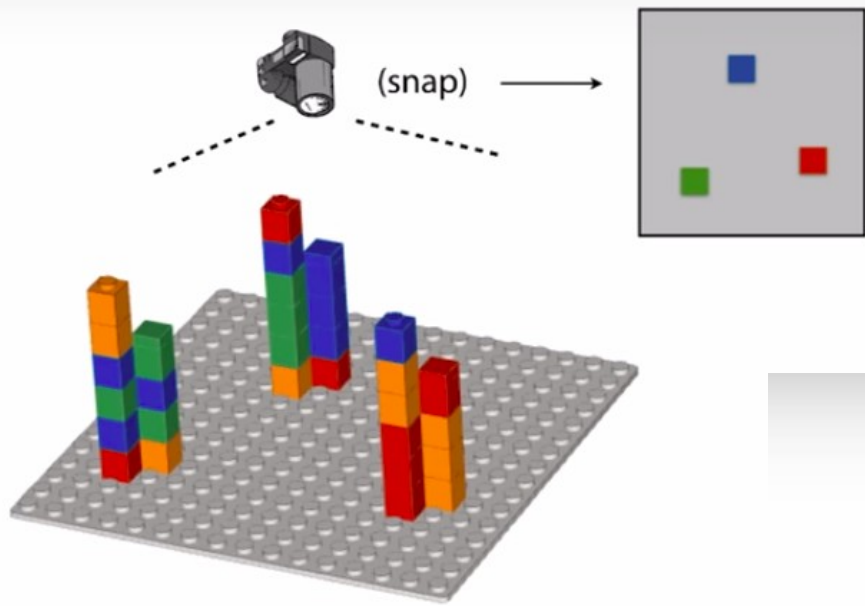
DNA polymerase



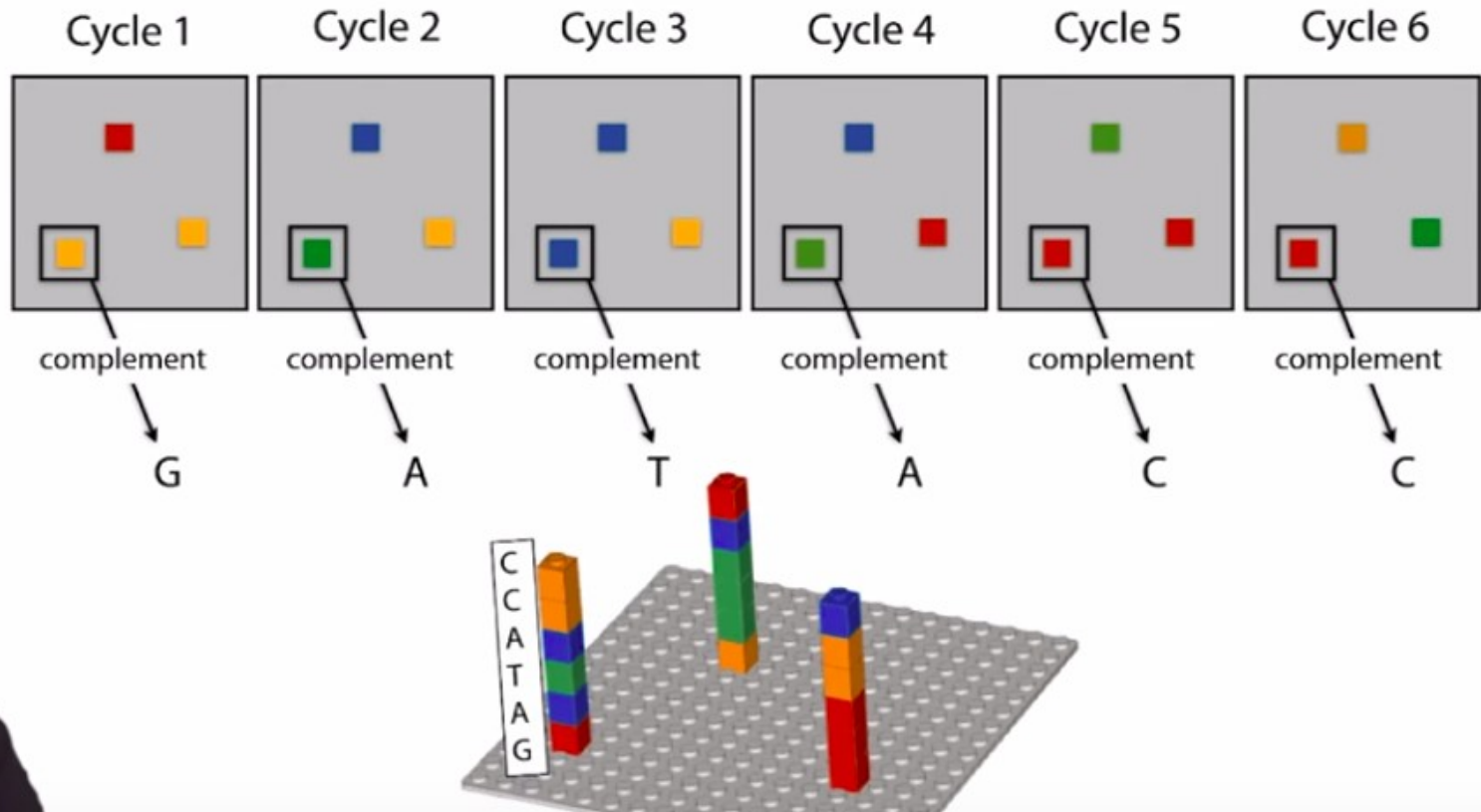


Remove terminators





# Sequencing by synthesis



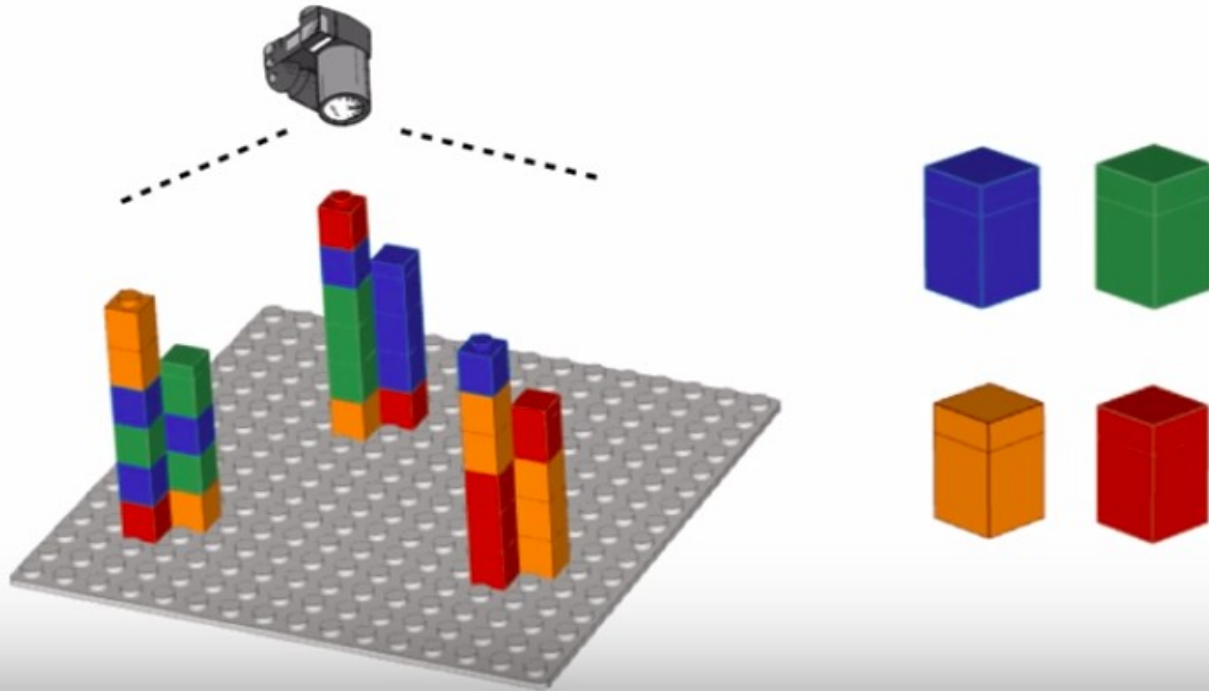


# Sequencing by synthesis

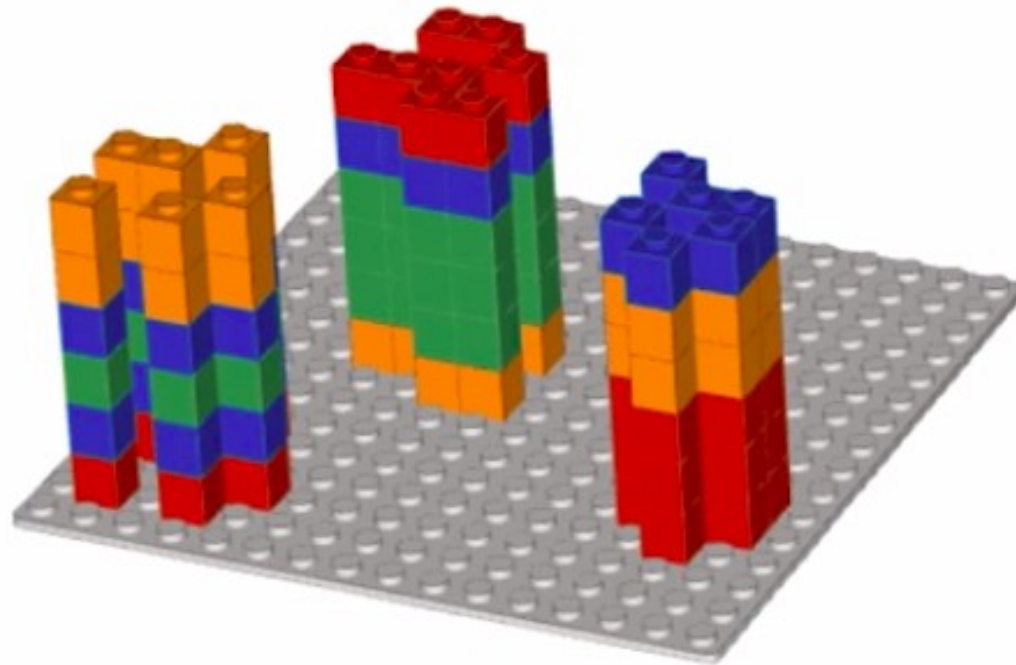
Billions of templates on a slide

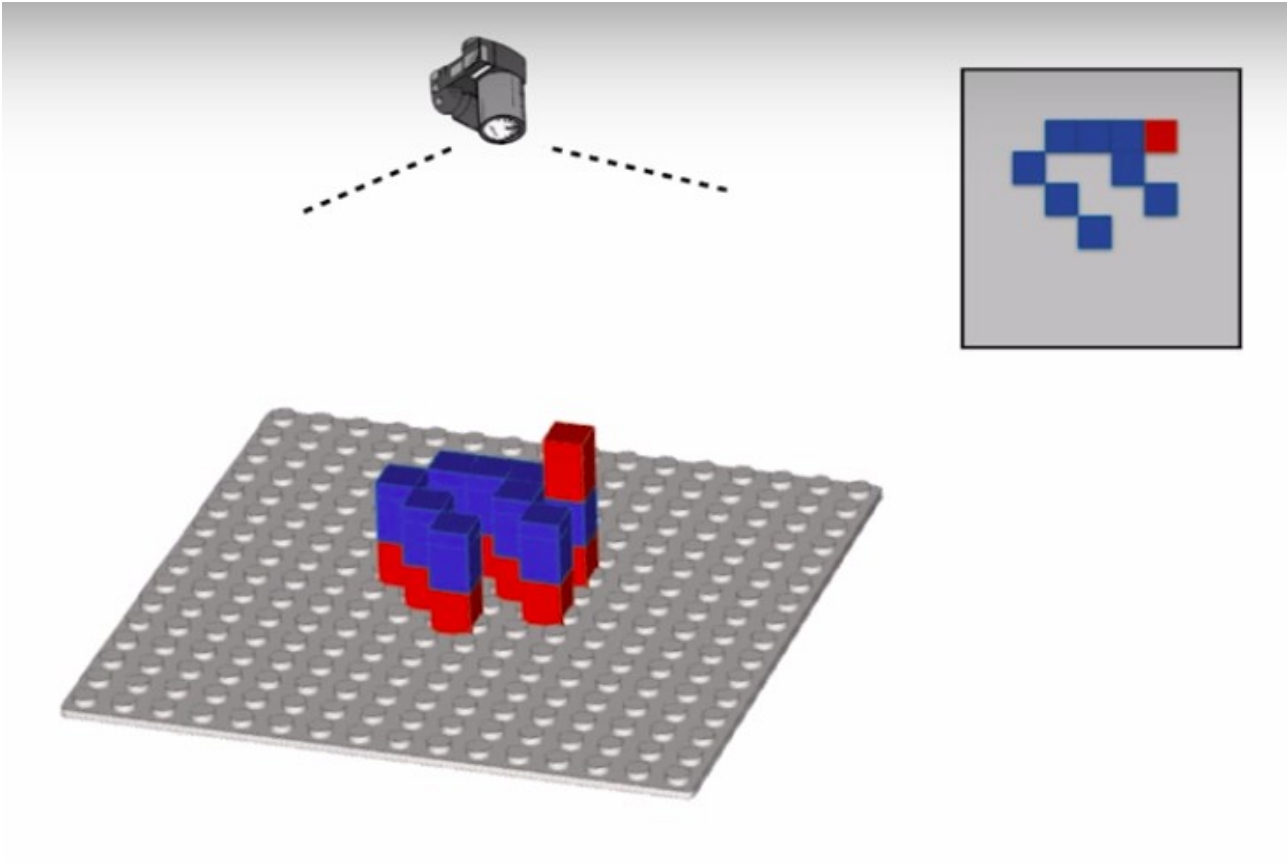
Massively parallel: photograph captures all templates simultaneously

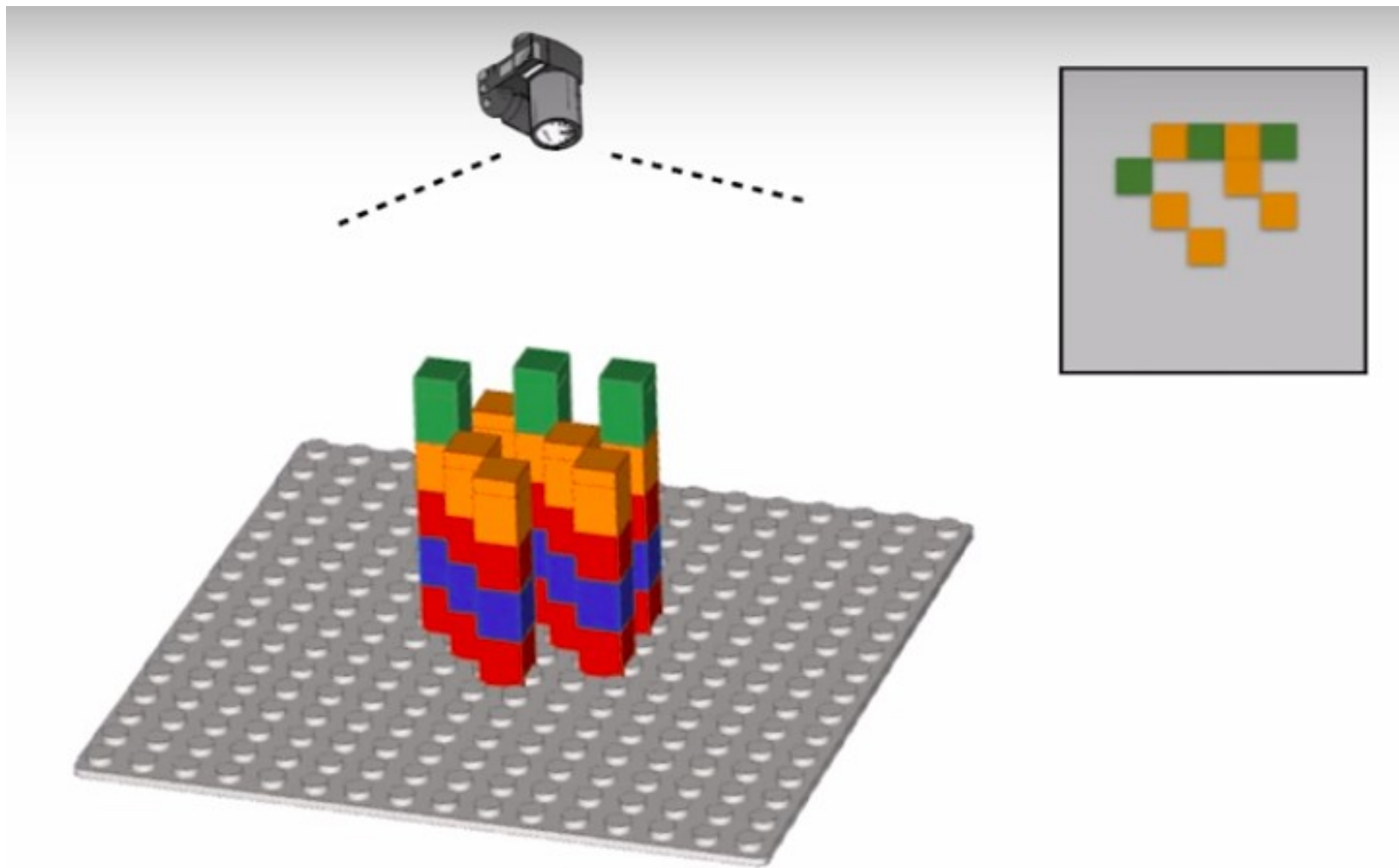
Terminators are “speed bumps,” keeping reactions in sync



Cluster of clones



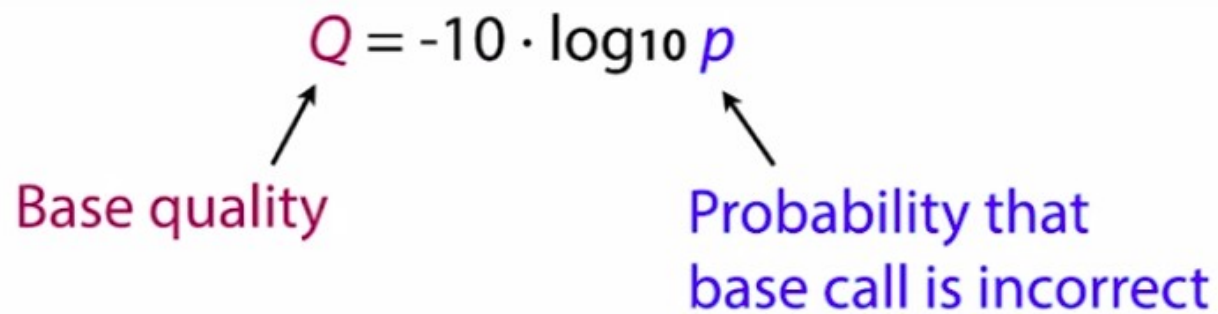




$$Q = -10 \cdot \log_{10} p$$

Base quality

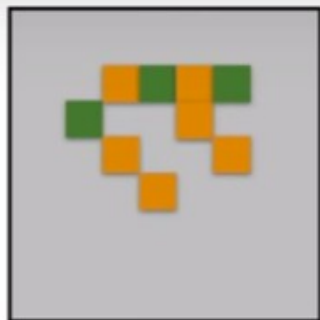
Probability that  
base call is incorrect



$Q = 10 \rightarrow 1$  in 10 chance call is incorrect

$Q = 20 \rightarrow 1$  in 100

$Q = 30 \rightarrow 1$  in 1,000



Call: orange (C)

Estimate  $p$ , probability incorrect:  
non-orange light / total light

$$p = 3 \text{ green} / 9 \text{ total} = 1/3$$

$$Q = -10 \log_{10} 1/3 = 4.77$$

# FASTQ

[illegible]

# Base qualities

Bases and qualities line up:

```
AGCTCTGGTGACCCATGGGCAGCTGCTAGGGA
| | | | | | | | | | | | | | | | | | |
HHHHHHHHHHHHHHHHGCGC5FEFFFGHHHHHH
```

Base quality is ASCII-encoded version of  $Q = -10 \log_{10} p$



# Base qualities

Usual ASCII encoding is "Phred+33":

take Q, rounded to integer, add 33, convert to character

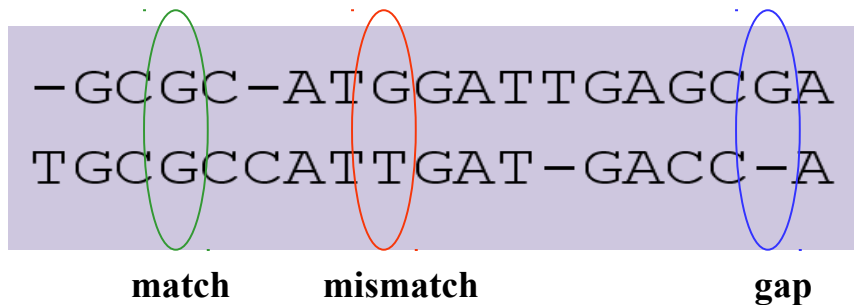
```
def QtoPhred33(Q):  
    """ Turn Q into Phred+33 ASCII-encoded quality """  
    return chr(Q + 33)  
           ↖  
           (converts character to integer according to ASCII table)
```

```
def phred33ToQ(qual):  
    """ Turn Phred+33 ASCII-encoded quality into Q """  
    return ord(qual)-33  
           ↖  
           (converts integer to character according to ASCII table)
```

# Alineamiento – de Secuencias

**Comparar secuencias  $\longrightarrow$  consiste en encontrar que partes de las secuencias son similares o parecidas y cuales difieren**

- Alineamiento de 2 secuencias  $\longrightarrow$  es el camino por el cual se coloca una secuencia debajo de la otra de manera de visualizar la correspondencia entre caracteres o espacios en la primer secuencia y caracteres o espacios en la segunda secuencia. No alinear espacio-espacio.**



## **Nucleótidos**

- **Comparar sec muy parecidas, diferencias de sólo 1 o 2 nucleótidos (estudios filogenéticos de poblaciones)**
- **Identificar genes: zonas exónicas (Codifican Genes) son mas conservadas que las intrónicas (No codificantes)**
- **Comparar secuencias no codificantes**
- **Seleccionar primers (Secuencias cortas de inicio en el proceso de replicación de ADN)**

## **Aminoácidos**

- **Búsqueda de homólogos (Existencia de ancestros comunes)**
- **Identificar regiones o dominios importantes de proteínas**

**Alineamiento Global** → alinear las secuencias enteras, alineando tantos caracteres como sean posibles, incluyendo los extremos. Inserción de espacios en posiciones arbitrarias en las secuencias de forma de tengan la misma medida

LGPSSKQTGKG--SRIWDN

LN-ITKSAGKGAIMRLGDA

- se utiliza para secuencias similares de aproximadamente la misma medida

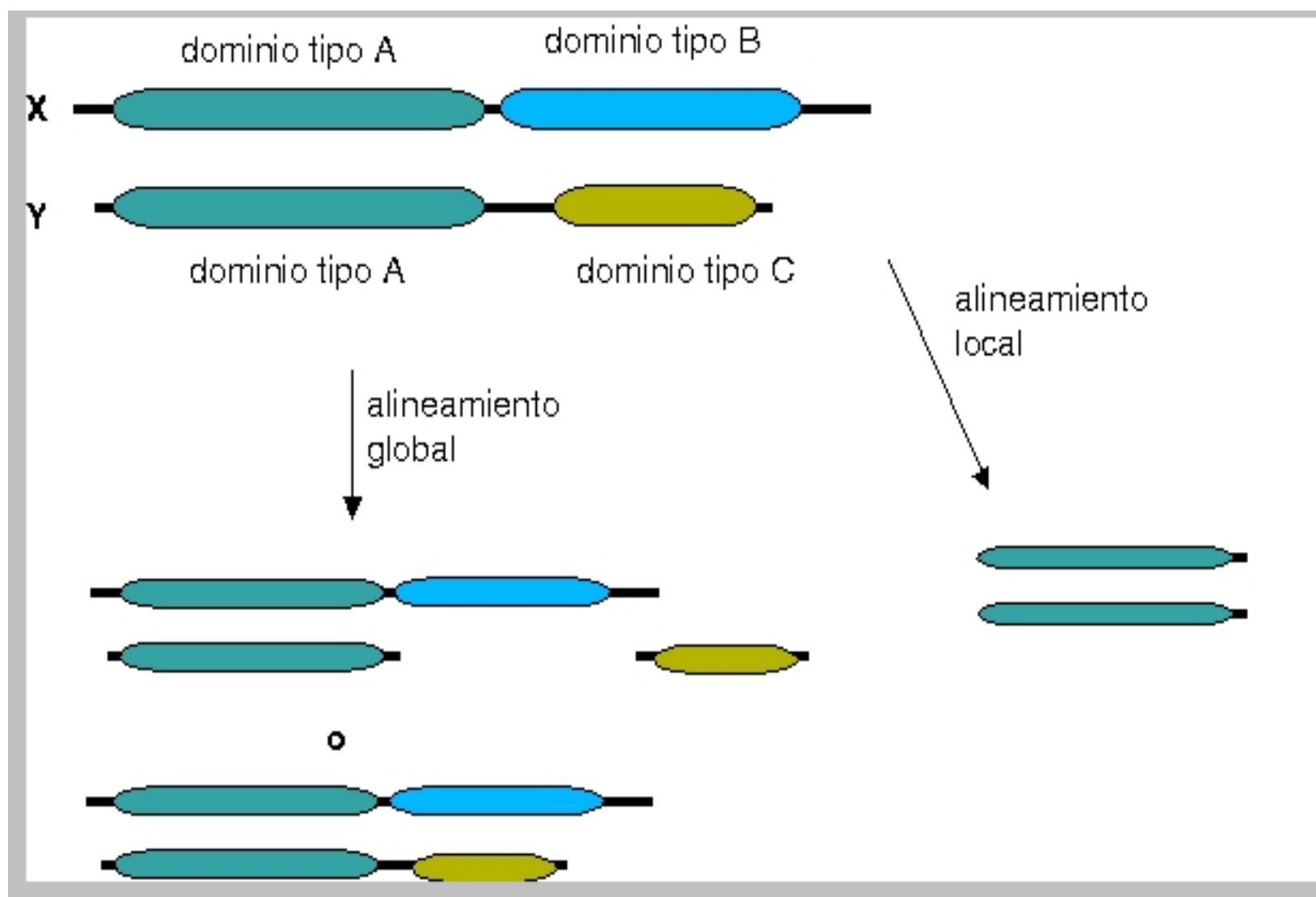
**Alineamiento Local** —————→ es un alineamiento entre una subregion o región local de la secuencia 1 y una subregion de la secuencia 2, dando un patrón de uniones. El alineamiento de detiene cuando finaliza la región de identidad.



The diagram shows a local alignment between two DNA sequences. The top sequence is represented by a series of dashes followed by the bold red text 'TGKG' and more dashes. The bottom sequence is represented by a series of dashes followed by the bold red text 'AGKG' and more dashes. The 'GKG' portion of both sequences is aligned vertically, indicating a region of identity.

- - - - - **TGKG** - - - - -  
- - - - - **AGKG** - - - - -

- se utiliza para encontrar patrones de nucleótidos conservados, secuencias de ADN o patrones de aa en sec. de proteínas



**Alineamiento semi-global → ignorar espacios  
en los extremos de las secuencias (aparecen antes o  
después de los últimos caracteres)**

**CAGCA - CTTGGATTCTCGG**  
**- - - CAGCGTGG - - - - -**



# Alineamientos

- Descubrir o buscar homólogos.
- Encontrar patrones de conservación.
- Construir taxonomías.
- Estudios filogenéticos
- Inferir los eventos del proceso evolutivo.
- Identificar proteínas. similares: predecir función y estructura.
- Identificar genes, determinar su función
- Identificar secuencias. repetidas
- Identificar reg. funcionales: orig replicación, sec anclaje, etc.
- Identificar mutaciones
- Localizar sec ADN solapadas: ensamble



**Esta información, a su vez, tiene miles de aplicaciones, por lo que los alineamientos son la herramienta base de toda la bioinformática.**

¿Cómo alineamos dos  
secuencias?

# Algunas formas de comparar secuencias

- A mano
  - Comparacion por identidades
- Se pueden aplicar los siguientes métodos informáticos:
  - Análisis de Dot Plot.
  - Algoritmos **Óptimos** de Programación Dinámica.
  - Heurísticas (FASTA y BLAST).

# Comparación por identidades

Desplazar una secuencia sobre otra y determinar cual es la superposicion (alineamiento) con mayor numero de identidades

Ejemplo: ACGTT Vs. CATTG

ACGTT CATTG	(0)	ACGTT 	(3)	ACGTT 	(1)
ACGTT CATTG	(0)	ACGTT 	(1)	ACGTT CATTG	(0)
ACGTT 	(1)	ACGTT 	(1)	ACGTT CATTG	(0)

# Desventajas

- No tiene en cuenta sustituciones, deleciones e inserciones de caracteres (nucleotidos o aminoacidos)
- No tiene en cuenta la frecuencia de caracteres en las sustituciones

# **Soluciones informáticas al problema del Alineamiento de Secuencias**

# Terminología

- **String**: secuencia ordenada de caracteres (TGATG) de un alfabeto {G,A,T,C}.
- **Prefijo**: letras consecutivas del comienzo (vacío,T,TG,TGAT).
- **Sufijo**: letras consecutivas del final (vacío, G, ATG).
- **Substring**: letras consecutivas (GA, ATG, G).
- **Subsecuencia**: ordenadas, no necesariamente consecutivas (TT, AG, TAG).

# Conceptos

- **Algoritmo**: Método no ambiguo. Secuencia de pasos bien definidos (instrucciones).
- **Complejidad**: Indica que tan eficiente es un algoritmo.  $O(n)$
- **Instrucciones**:
  - Para**  $i = 1$  **hasta**  $n$
  - Para**  $j = 1$  **hasta**  $m$
  - Si**  $i = n$  **hacer**
  - $A[i, j] = 12$
  - Devolver**  $i$
- **Función**: Algoritmo que puede ser invocado mediante un nombre (suma(3,2), Alinear(S,T)).



# Complejidad Algorítmica

- **Complejidad Temporal:** Es un indicador del tiempo necesario para ejecutar el algoritmo.
- **Complejidad Espacial:** Es un indicador del espacio en memoria que ocupan los datos (matrices, secuencias, etc.) del algoritmo.
- La complejidad depende del **tamaño de entrada**.

- $O(n)$ : depende **linealmente** del tamaño de la entrada.
- $O(1)$ : el algoritmo es **constante**, siempre tarda lo mismo, no depende del tamaño de la entrada.
- $O(n^2)$ : depende **cuadráticamente**.
- $O(p(n))$ ,  $p$ : funcion polinómica: depende **polinomialmente**.
- $O(c^n)$ ,  $c > 1$  : depende **exponencialmente**.

- Generalmente se usan las siguientes simplificaciones para el cálculo de la complejidad de una función  $f(x)$ :
- Si  $f(x)$  es la suma de varios términos. Si existe uno de los términos con la mayor tasa de crecimiento, éste se puede conservar. Todos los demás pueden ser omitidos
- Si  $f(x)$  es el producto de varios factores, cualquier constante (que no dependa de  $x$ ) puede ser omitida.
- $f(x) = 6x^4 - 2x^3 + 5 \rightarrow f(x) = O(x^4)$

# Recursión

- **Funcion Recursiva:** es una función que se llama (invoca) a si misma para resolver una porción menor del problema hasta encontrar una llamada final que no requiere otra llamada (caso base).
- Se requiere de un caso base, y de que las llamadas reduzcan el problema hasta alcanzarlo.

# Ejemplo de Recursión

- Factorial:

$$n! = \begin{cases} 1 & \text{Si } n \leq 1 \\ n*(n-1)! & \text{En los demás casos} \end{cases}$$

Función Factorial(n):

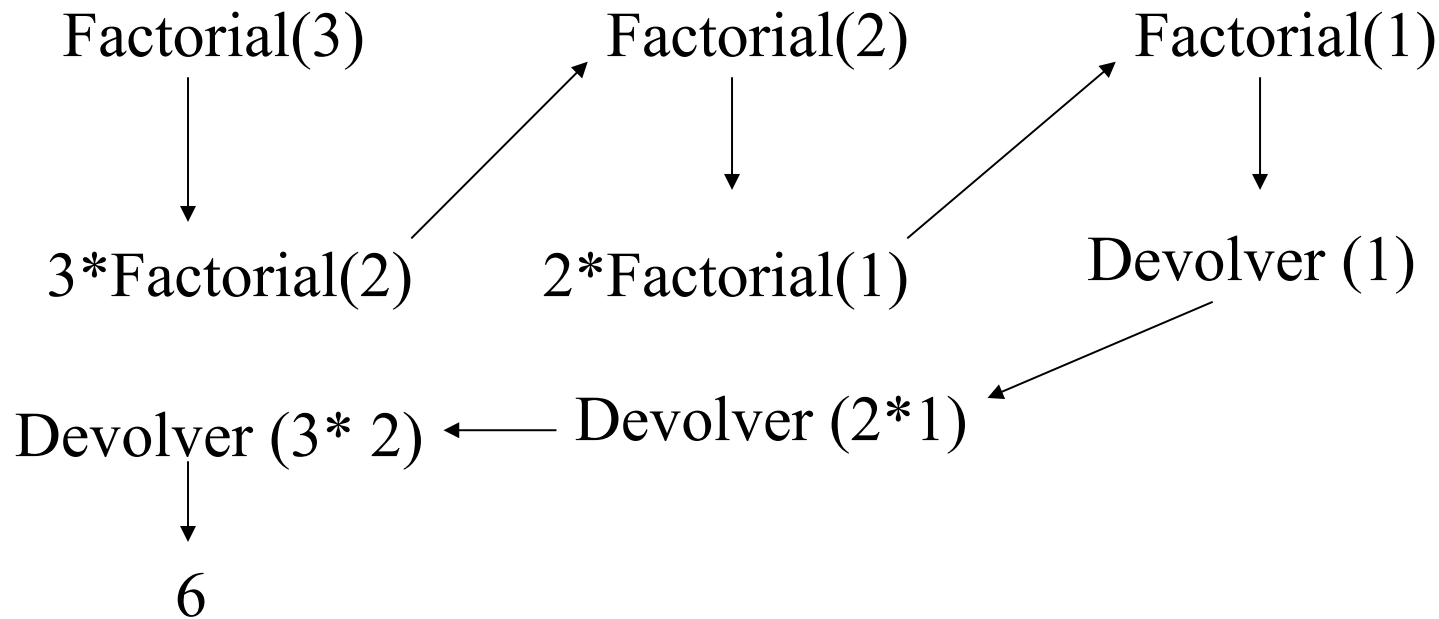
Si  $n = 1$  hacer

Devolver 1

Sino hacer

Devolver (  $n * \text{Factorial}(n-1)$  )

**Ejemplo:** Calcular el Factorial de 3:



(Dos llamadas recursivas hasta llegar al caso base)

Las “llamadas” (invocación) a la función crean un anidamiento y “esperan” el resultado devuelto por la llamada recursiva.

# Alineamiento

- Un **Alineamiento** de un String S y un String T es un par de Strings S' y T' tal que:
  - La longitud de S' y T' son iguales ( $\text{long}(S') = \text{long}(T')$ )
  - No se alinean 2 espacios juntos (G\_AC no se alinea a A\_GC).
- **Alineamiento Óptimo** de S y T es cuando tienen el mayor puntaje (score).

# Alineamiento

Estrategia de Alineamiento?



# Alineamiento de secuencias

## Ejemplos de alineamiento

Sin Gaps:(10 coincidencias)

```
a:  ATATTGCTACGTATATCAT
      |||||
b:  ATATATGCTACGTATCAT
```

Con gaps en a (14 coincidencias):

```
a:  ATAT-TGCTACGTATATCAT
      |||  |||||
b:  ATATATGCTACGTATCAT
```

Con gaps en a y b (16 coincidencias):

```
a:  ATAT-TGCTACGTATATCAT
      |||  |||||  |||||
b:  ATATATGCTACG--TATCAT
```

El objetivo del alineamiento es conseguir alinear las posiciones homólogas.

# Dot plot:

Dot plot es una representacion grafica de similaridad entre dos secuencias

Está compuesta por una matriz, cuyos ejes se forman con las dos secuencias que se desean alinear.

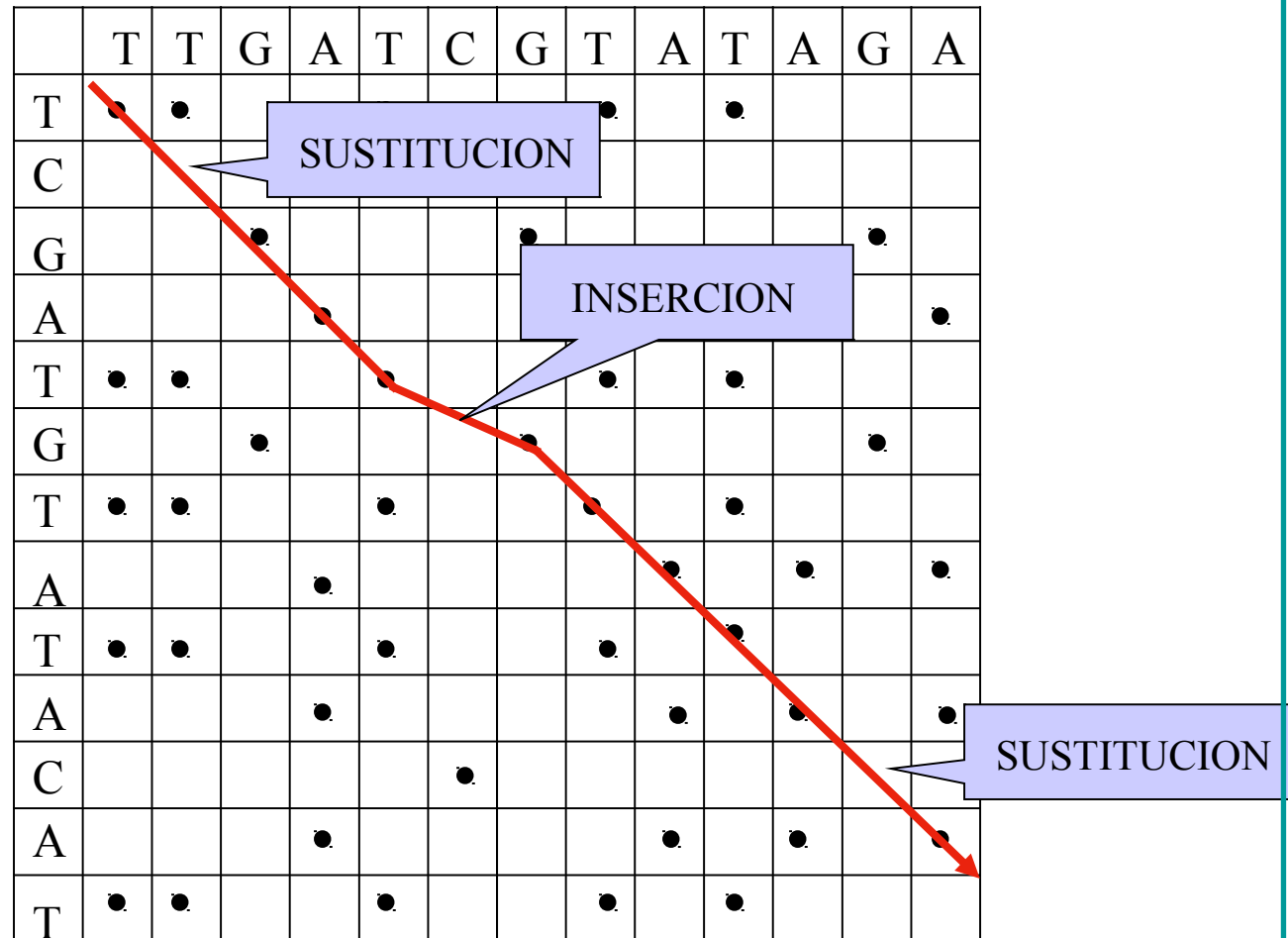
Luego se llena con un punto la intersección donde coinciden los caracteres.

Por último, se traza una línea uniendo las diagonales formadas por las regiones similares.

# Dot plot:

Secuencia 1

Secuencia 2



Secuencia 1

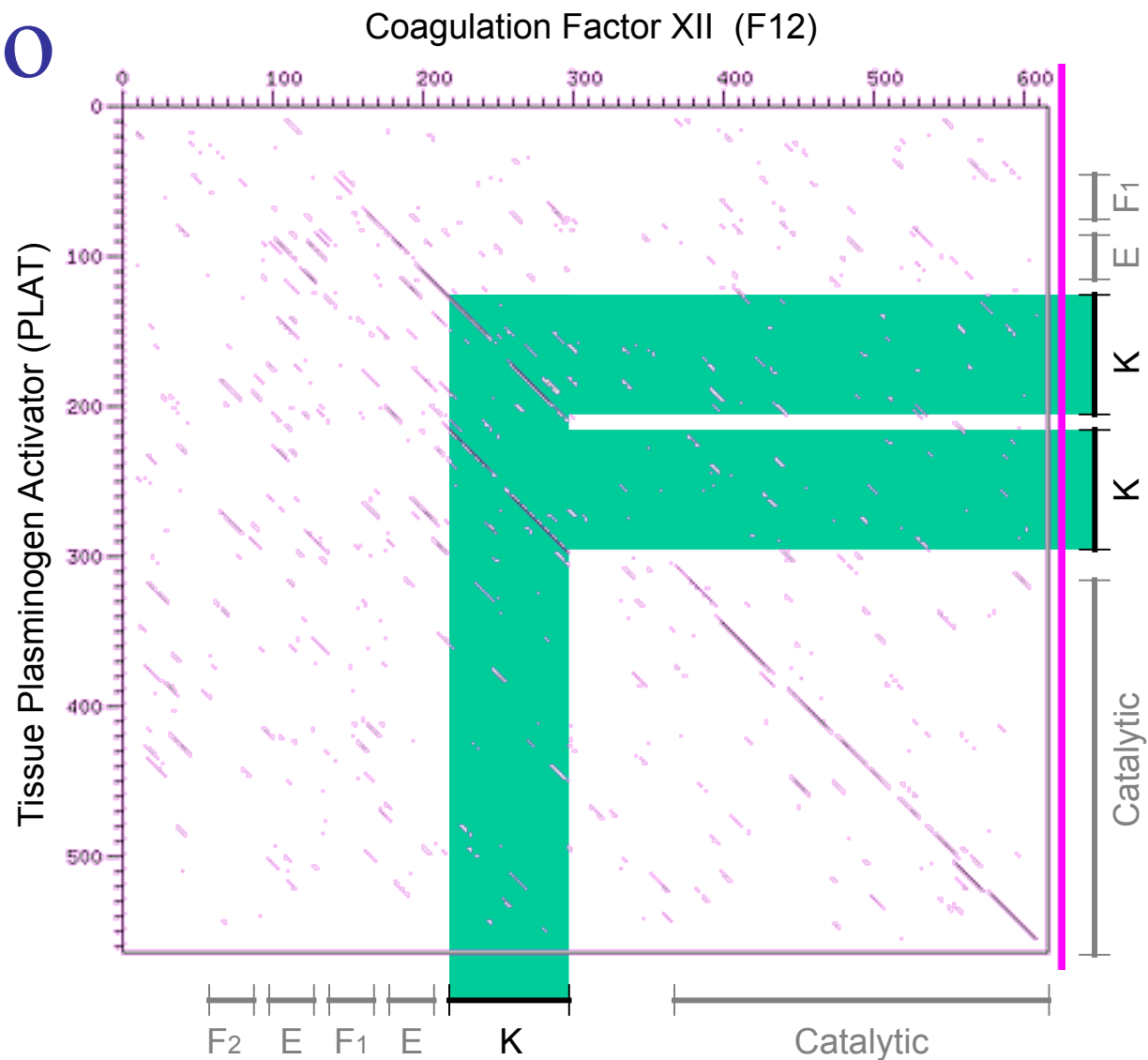
TCGAT-GTATACAT

Secuencia 2

TTGATCGTATAGA-

# Dot plot: ejemplo

**Regiones  
repetidas  
confunden a los  
algoritmos de  
alineamiento.**



# Ventajas

- Es una forma rápida y gráfica para encontrar regiones de apareamiento entre dos secuencias, dejando al investigador la elección de las regiones de interés.
- Es útil para encontrar regiones repetidas e invertidas
- Es útil como primer paso antes de aplicar algoritmos de programación dinámica.

# Desventajas

- A veces no es fácil encontrar el mejor apareamiento de forma objetiva. (Ej. regiones repetidas)
- Cuando se analiza una secuencia con una base de datos de secuencias, ¿de que forma encontrar las secuencias que mayor similitud tengan? y en tal caso ¿como poder inferir una homología?



Necesidad de una medida objetiva



?



Con dot plot no se puede ya que se debería calcular el score para todos los alineamientos posibles y este cómputo es exponencial

# Desventajas

- A veces no es fácil encontrar el mejor apareamiento de forma objetiva. (Ej. regiones repetidas)
- Cuando se analiza una secuencia con una base de datos de secuencias, ¿de que forma encontrar las secuencias que mayor similitud tengan? y en tal caso ¿como poder inferir una homología?



Necesidad de una medida objetiva



Scoring: sistema de puntaje que representa la rigurosidad del alineamiento



Con dot plot no se puede ya que se debería calcular el score para todos los alineamientos posibles y este cómputo es exponencial

# Scores

El score del alineamiento de dos secuencias dadas es una función que nos dá un valor numérico, que indica cuanto se parecen entre sí.

Esta función premia las coincidencias de caracteres (match) y penaliza los gaps y las diferencias (mismatch)

El score máximo de todos los alineamientos posibles entre un conjunto dado de secuencias se denomina **similitud**. En general, puede haber varios alineamientos distintos que tengan un score máximo.

Cuando se comparan dos alineamientos por medio de sus scores, debe tenerse en cuenta que dichos valores provengan de un mismo sistema de alineamiento.



# Scores

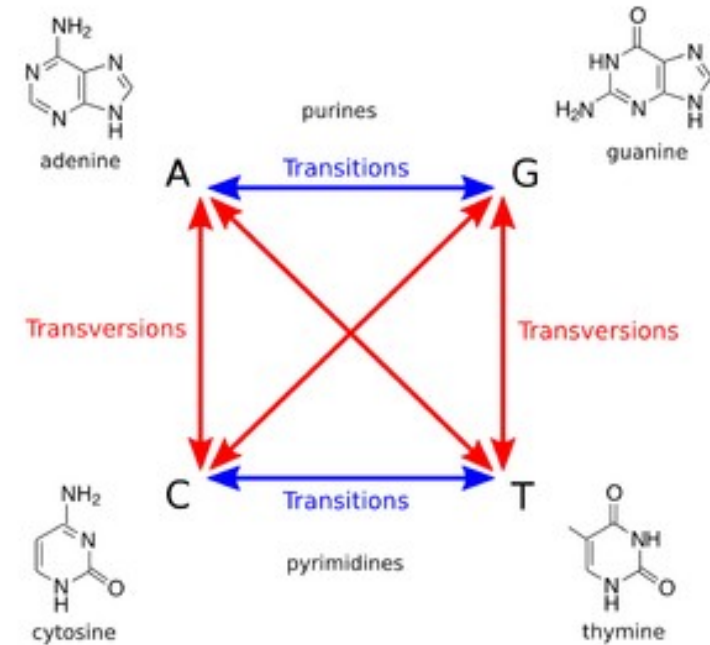
- **Criterios de valoración de sustituciones y deleciones o inserciones**

Sustitución: dependiendo de la frecuencia con que ocurra reciban distintos puntajes. Menor valor cuando menos frecuencia tenga.

- DNA: transversiones vs transiciones
- Proteínas: aminoácidos pertenecientes al mismo grupo son más frecuentes

Deleciones o inserciones (gap):

- Se penalizan más que las sustituciones
- En la mayoría de los casos se penaliza mas la aparicion de gap individuales que la extension de un gap a mayor longitud



Debido a que, a diferencia de la Transición la Transversion es una mutación que cambia la estructura química de manera drástica, las consecuencias de la Transversión tienden a ser más drásticas – y se penalizan más.

# Scores

Otro ejemplo de scoring de secuencias de Nucleótidos:

- Los nucleótidos A y G son de la familia de las purinas.
- Los nucleótidos C y T son de la familia de las pirimidinas.
- Si dos nucleótidos coinciden le asignamos el valor +3.
- Si uno de los nucleótidos es A y el otro G le asignamos el valor +1.
- Si uno de los nucleótidos es C y el otro T le asignamos el valor +1.
- Si uno de los nucleótidos es A (resp. G) y el otro es C (resp. T), le asignamos el valor -1.
- Además, a la aparición de un hueco le asignaremos el valor -2.

En este caso, la matriz de scoring  $s(A[i], B[j])$  sería la siguiente:

	A	C	G	T
A	3			
C	-1	3		
G	1	-1	3	
T	-1	1	-1	3

# Scores

Ejemplo

T	GA-CGGATTAG
S	GATCGGAATAG

match = +1

mismatch = -1

gap penalty = -2

$$\text{Score}(T, S) = 9 \times 1 + 1 \times (-1) + 1 \times (-2) = 6$$

## Ejercicio:

1. Dada la siguiente secuencia: TAATCGAATGGGC derive seis posibles secuencias de aminoácidos que se generen a partir la secuencia dada.
2. Diseñe un algoritmo que pruebe si una cadena  $p$  es una subcadena de otra cadena  $t$  – Puede diseñar el algoritmo para que corra en tiempo  $O(|p| + |t|)$ ?

Subir un doc con sus respuestas a la carpeta tareas.  
CC471-Lec02-Nombre-apellido.zip