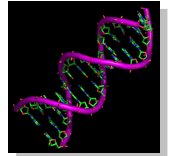


Aplicación de GPUs en Bioinformática: Alineamiento de Secuencias



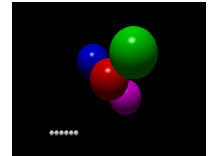
- **Bioinformática**

- Definición
- Biología molecular
- Alineamiento de Secuencias y Aplicaciones



- **GPGPU**

- Introducción
- Ventajas



- **Procesamiento en GPU**

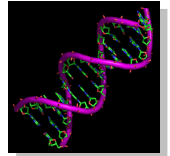
- Alineamiento de Secuencias Smith-Waterman
- Alineamiento de Secuencias por Identidad



Artwork: Jane Ailes, NHGRI

- **Conclusiones**

Definiciones



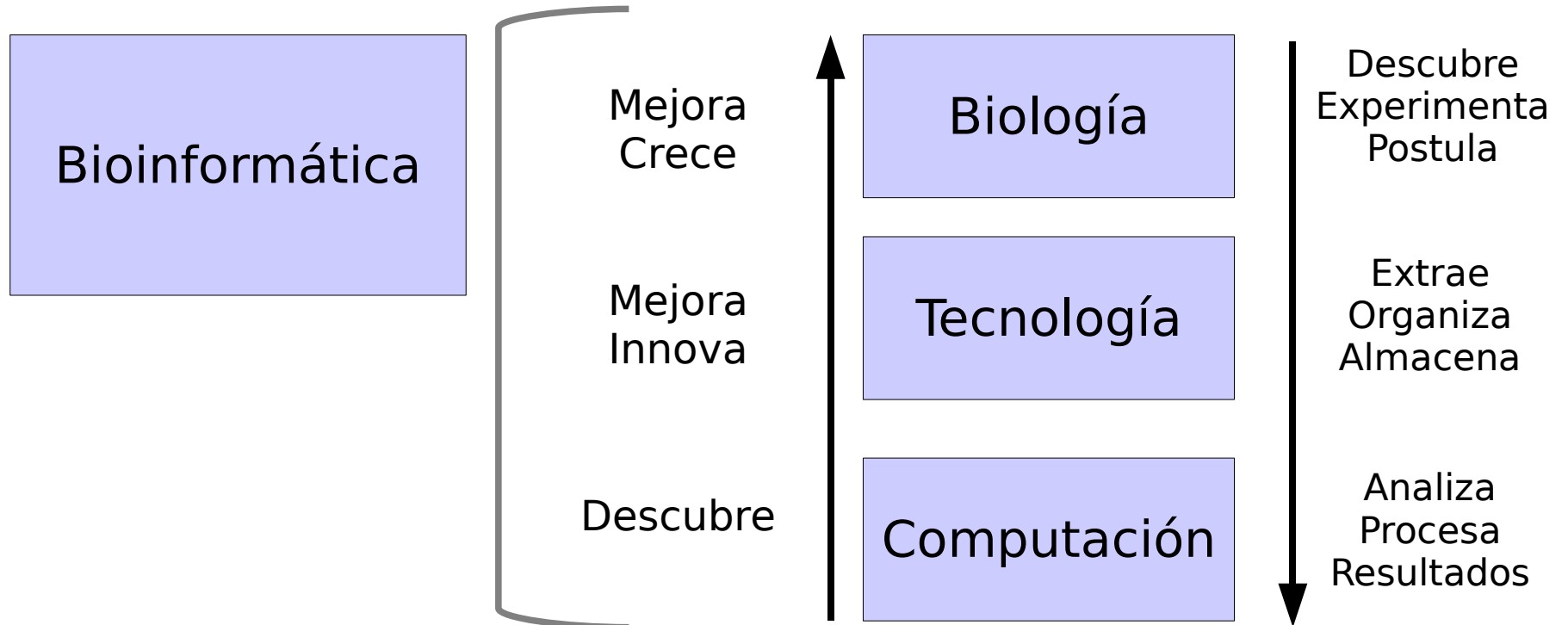
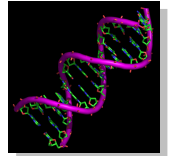
Rama de las Ciencias de la Computación que estudia sistemas de cómputo y tratamiento de la información para el análisis de datos experimentales de sistemas biológicos, así como la simulación de los mismos.

Wikipedia

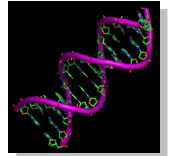
Aplicación de la computación y las matemáticas que permite la administración, análisis y comprensión de datos para resolver preguntas biológicas. (con conexiones a medi-, quimio-, neuro-, etc. informática).

Wikipedia II

Conceptos



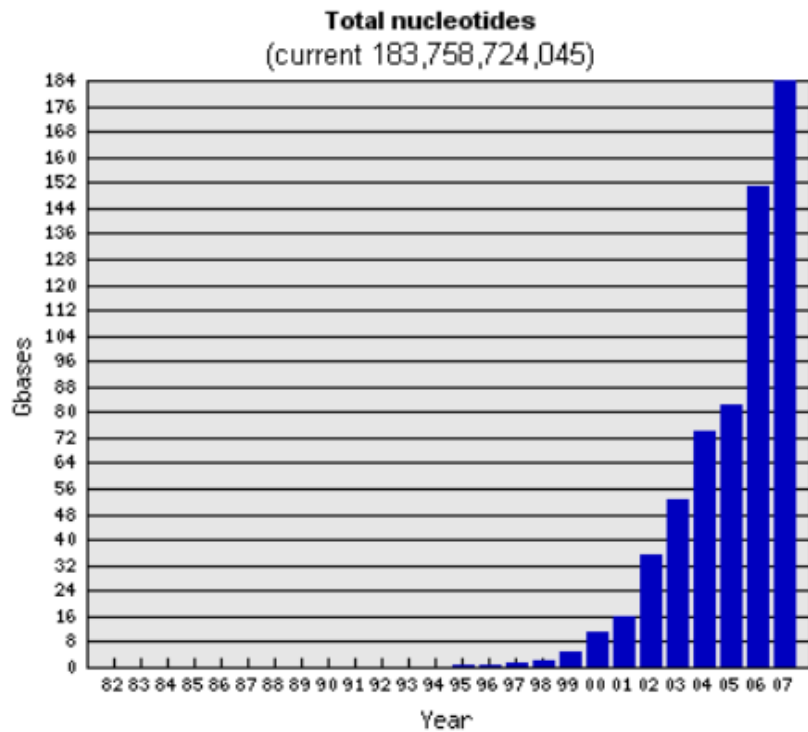
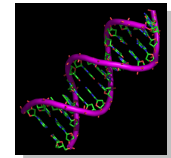
Crecimiento de Datos



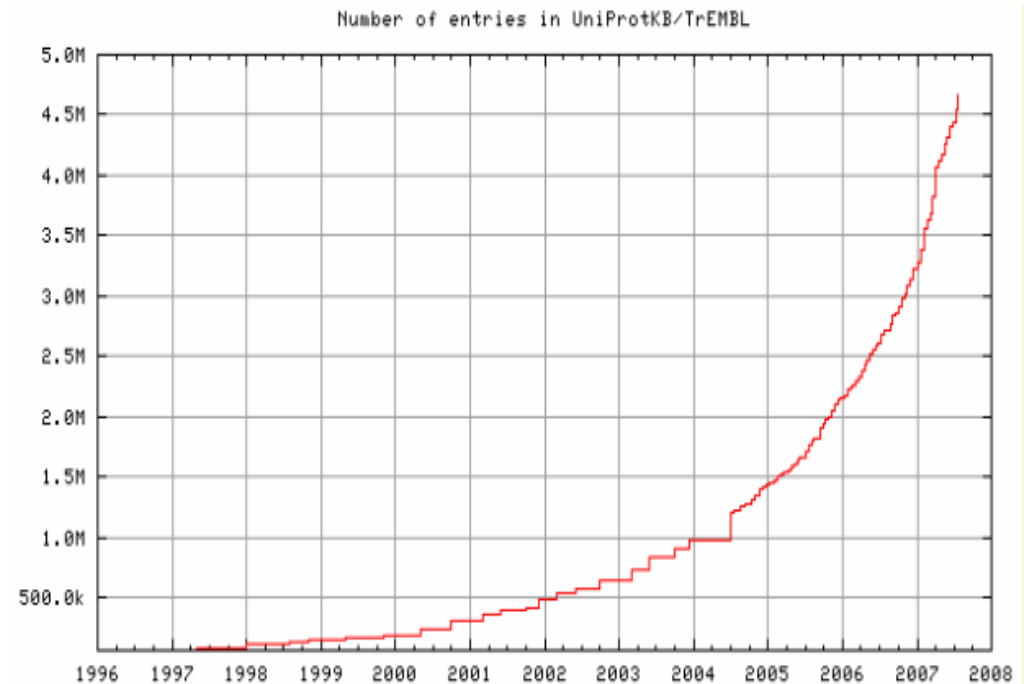
El principal problema es la explosión de información a analizar.

Los avances de la Bioinformática han sido mayores a medida que se han ido acumulando grandes volúmenes de información obtenidos por nuevas tecnologías, pero, paralelamente, también ha crecido en complejidad la tarea de procesar, analizar y almacenar toda esta información, lo cual supone un cuello de botella en las tareas de investigación, ya que el ritmo de producción de datos es mucho mayor que la funcionalidad de las técnicas de análisis existentes. Esto ha creado la necesidad inmediata de desarrollos en campos de tecnología de la información orientados a la creación de herramientas eficientes especializadas para el análisis y comprensión de datos biológicos.

Crecimiento de Datos



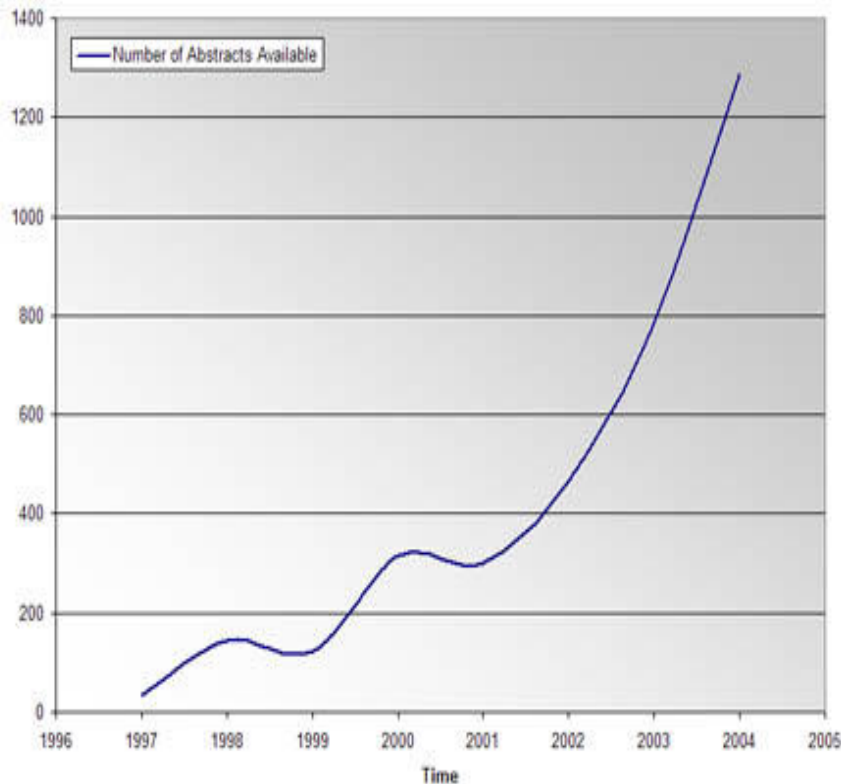
Base de Datos de Secuencias de Nucleótidos (ADN y ARN)



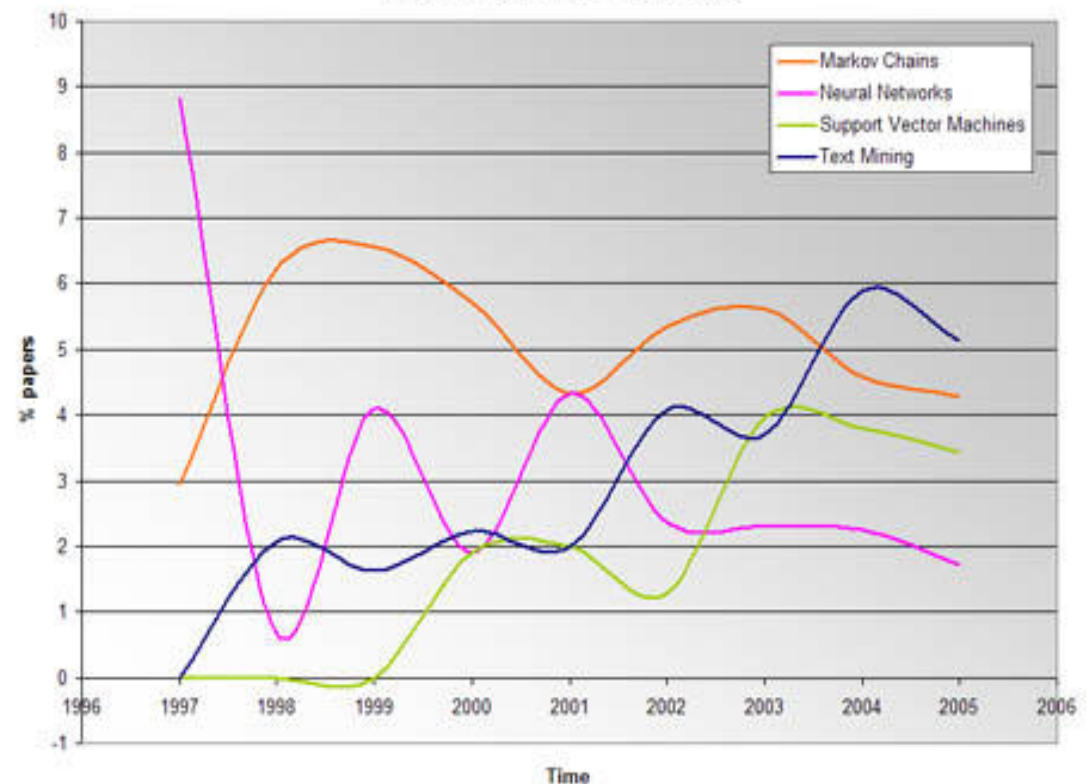
Base de Datos de Secuencias de Proteínas

Crecimiento de Datos: Publicaciones

Abstracts Available vs Time



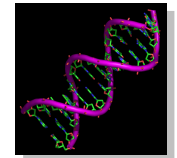
Machine Learning



Estudio sobre “abstract” de 2.920 artículos en PubMed

<http://www.ghastlyfop.com/blog/2005/10/bioinformatics-zeitgeist-05.html>

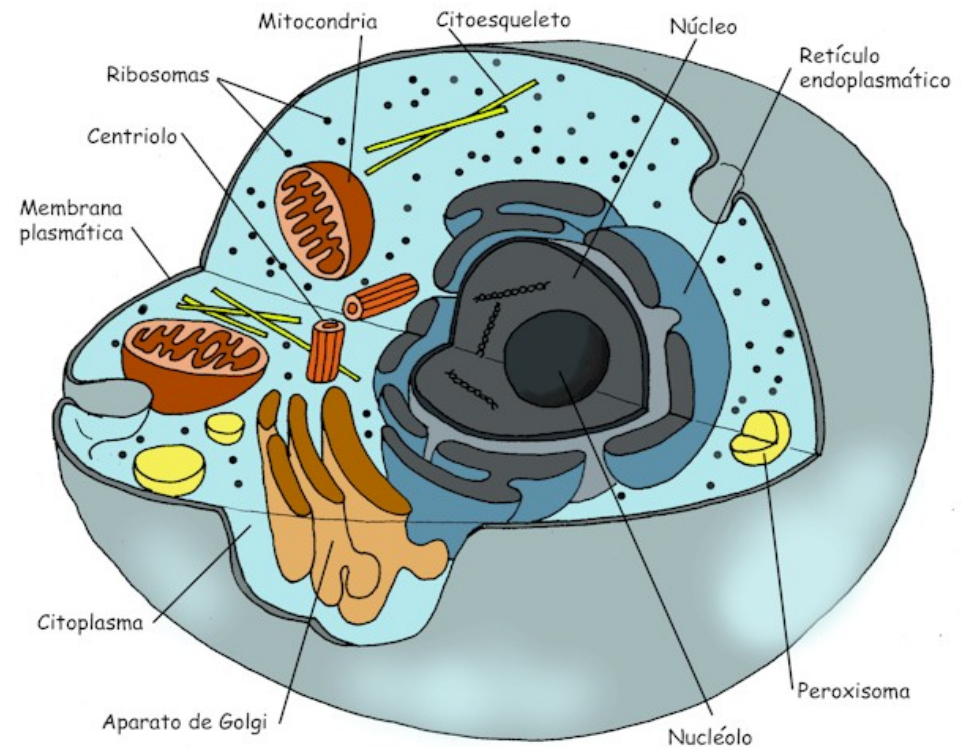
Biología Molecular



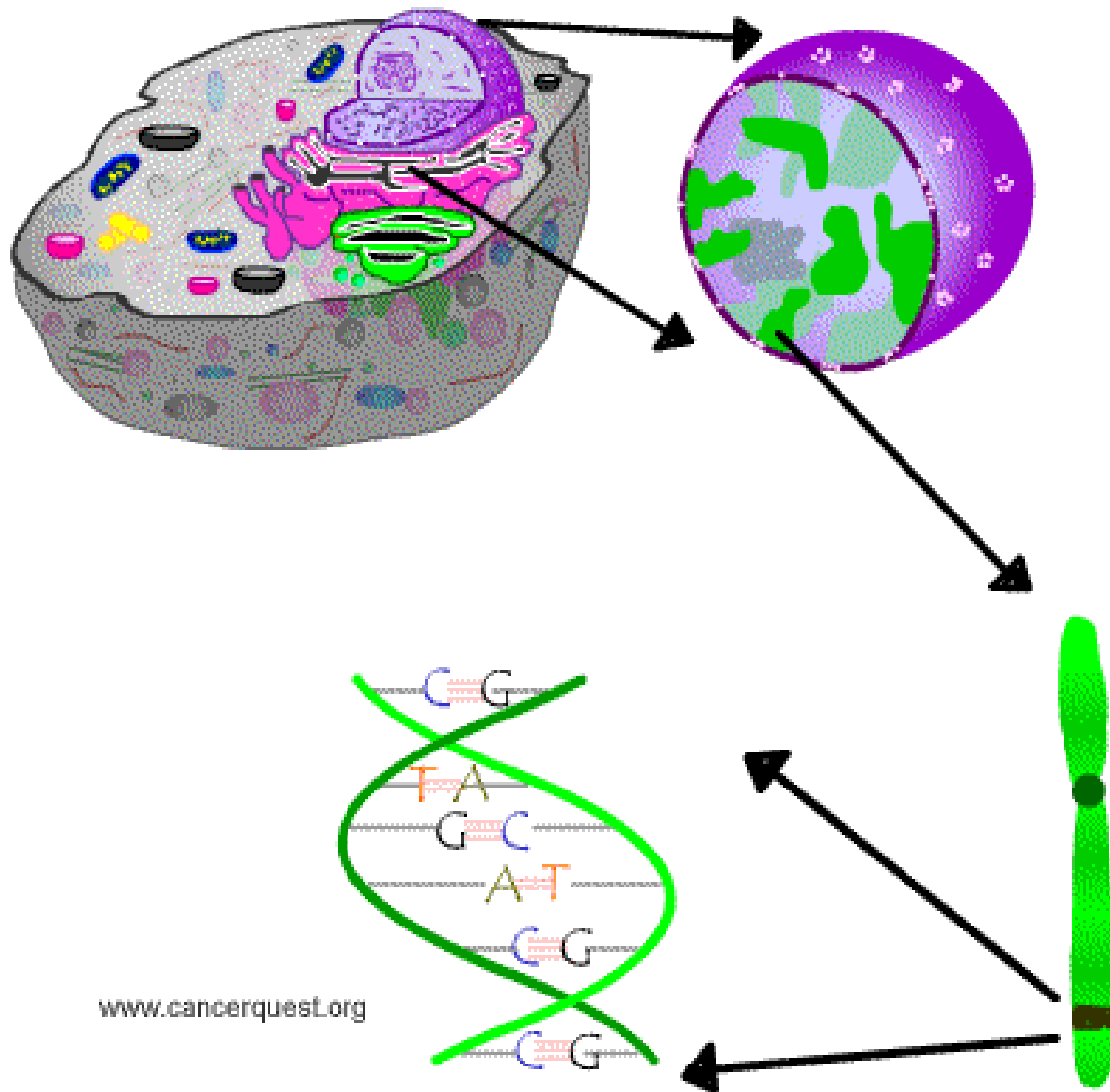
Células:

La teoría celular establece que todos los seres vivos están formados por células, que son las unidades básicas de vida, y que cada célula proviene de otra célula.

En organismos eucariotas el núcleo contiene una o más moléculas de ADN de doble hebra organizadas en forma de cromosomas



Biología Molecular



www.cancerquest.org

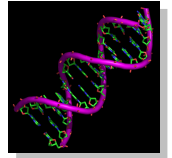
Analogías:

Núcleo	= Biblioteca
Cromosomas	= Estanterías
Genes	= Libros

Casi todas las células de nuestro cuerpo tiene las mismas bibliotecas y libros.

En los libros está toda la información para que cada célula desarrolle sus funciones.

Biología Molecular



ADN:

Contiene la información de como funcionan las células

ARN:

Transfiere información del ADN a otras partes de la célula.

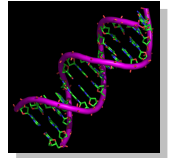
Gracias al ARN se pueden sintetizar las proteínas.

Proteínas:

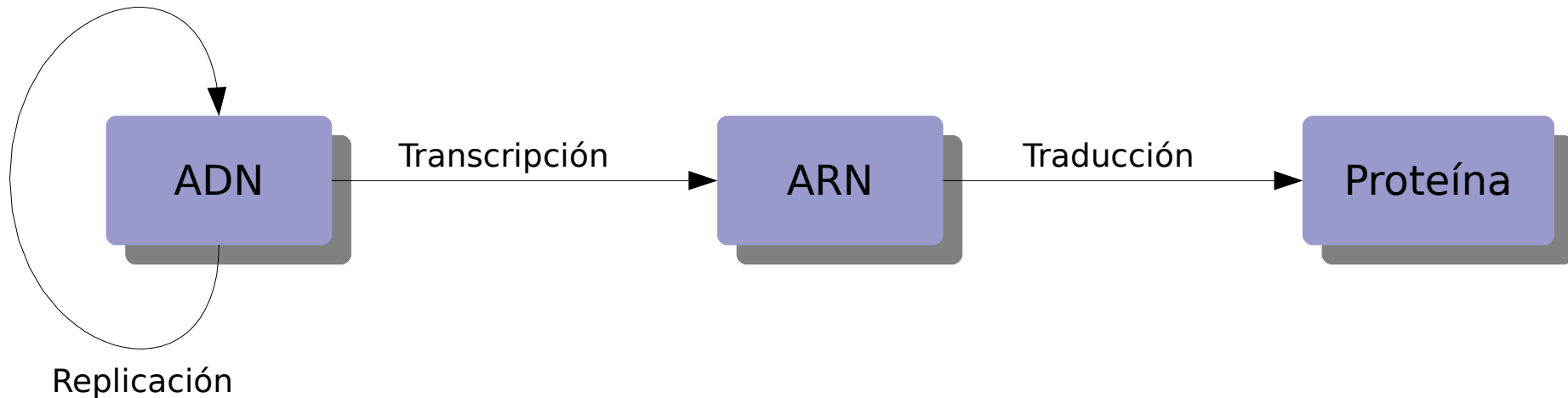
Estamos hechos de proteínas.

Forma enzimas que envían señales a otras células regulan la actividad de otros genes.

Biología Molecular



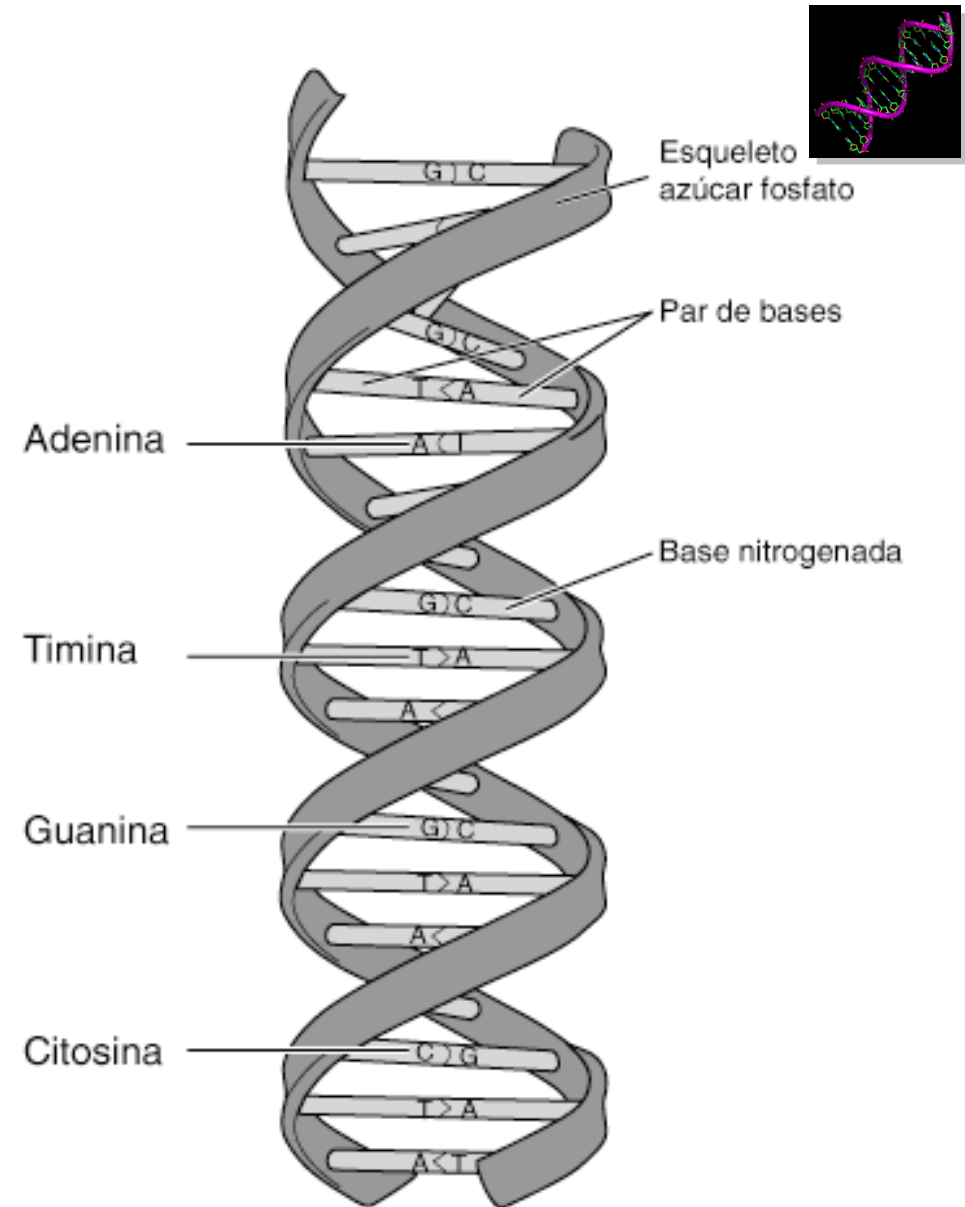
Dogma Central de la Biología Molecular



Biología Molecular

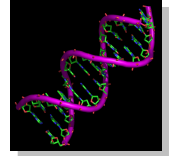
Alfabeto del ADN A-C-G-T

Código genético de todos los organismos formado por cuatro nucleótidos diferentes: Adenina (A), Guanina (G), Citosina (C) y Timina (T) las cuales también pueden aparecer en pares complementarios formados por A-T y C-G

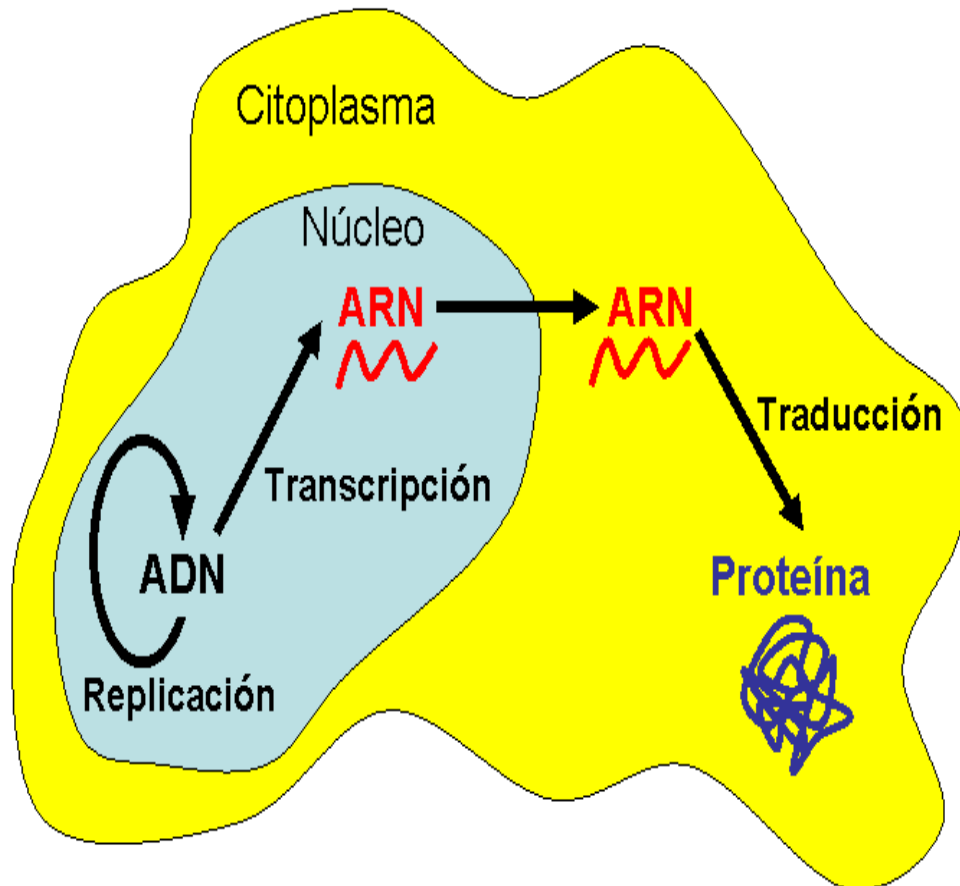


Biología Molecular

ARN Mensajero



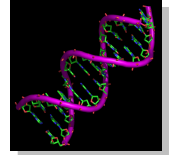
Flujo de la información genética en eucariontes



- Se traduce, se copia en la transcripción de ADN a ARNm.
- El alfabeto es el mismo pero se cambia T por U (Uracilo)
- El ARN es generalmente de cadena simple.

Biología Molecular

Proteínas

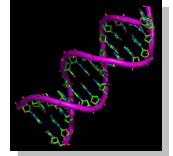


		Segunda base				
		U	C	A	G	
P r i m e r a b a s e	U	UUU } Fen UUC } UUA } Leu UUG }	UCU } Ser UCC } UCA } UCG }	UAU } Tir UAC } UAA Alto UAG Alto	UGU } Cis UGC } UGA Alto UGG Trp	U C A G
	C	CUU } Leu CUC } CUA } CUG }	CCU } Pro CCC } CCA } CCG }	CAU } His CAC } CAA } Glu CAG }	CGU } Arg CGC } CGA } CGG }	U C A G
	A	AUU } Ile AUC } AUA } AUG Met inicio	ACU } Tre ACC } ACA } ACG }	AAU } Asn AAC } AAA } Lys AAG }	AGU } Ser AGC } AGA } Arg AGG }	U C A G
	G	GUU } Val GUC } GUA } GUG }	GCU } Ala GCC } GCA } GCG }	GAU } Asp GAC } GAA } Glu GAG }	GGU } Gli GGC } GGA } GGG }	U C A G

- Se decodifica el ARN y se construye una cadena de aminoácidos (la proteína).
- Cuatro bases posibles en cada posición (A, C, G, U)
- Cada aminoácido se codifica con 3 nucleótidos (codón).
- $4 \times 4 \times 4 = 64$ secuencias de 3
- Codones de inicio (1) y de parada (3)

- Quedan 61 codones para 20 aminoácidos = Hay redundancia.
- Un mismo aminoácido puede ser codificado por varios codones (Serina)

Biología Molecular Resumiendo



----- ADN -----
ATGGAAGTATTTAAAGCGCCACCTATTGGGATATAAG...
----- Codones -----
ATG GAA GTA TTT AAA GCG CCA CCT ATT GGG ATA TAA G...
----- Proteína -----
M E V F K A P P I G I stop

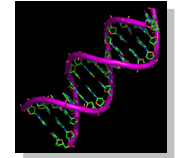
La proteína es una secuencia de aminoácidos codificada en ADN

Las funciones de una proteína se pueden inferir tanto de su estructura como de su secuencia.

Proteínas similares suelen tener funciones similares. Patrones.

	Número Cromosomas	Tamaño Genoma	Genes
Homo Sapiens	23	3.200.000.000	30.000
Mus Musculus	22	2.600.000.000	30.000
Dro. Melanogaster	4	180.000.000	18.000

Alineamiento de Secuencias



Panorama

Cada vez se secuencia más rápido y mayor cantidad de especies e individuos.

Hay muchas base de datos diferentes con un crecimiento exponencial:

- De secuencias (ADN, ARN, Aminoácidos)
- Grafos (Redes de interacción, árboles filogenéticos,...)
- Patrones y Modelos matemáticos.
- Texto Libre

Bases datos públicas de fácil acceso.

Herramientas de libre acceso.

Código abierto (BIOPerl, etc)

Super Computadores.

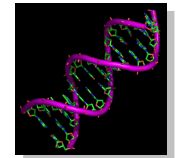


Alineamiento de Secuencias

Secuencias

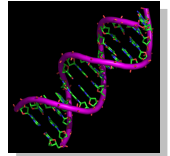
Se puede acceder a las base de datos con millones de secuencias (DNA, RNA y Proteínas):

- Desde la Web
- Usando Herramientas (WS, Cliente-Servidor, etc)
- Descargando las BD en local



```
>P05067|A4_HUMAN Amyloid beta A4 protein - Homo sapiens (Human).  
MLPGLALLLLAAWTARALEVPTDGNAGLLAEPQIAMFCGRLNMHMNVQNGKWSDPSGTK  
TCIDTKEGILQYCQEVPELQITNVVEANQPVTIQNWCKRGRKQCKTHPHFVIPYRCLVG  
EFVSDALLVPDKCKFLHQERMDVCETHLHWHTVAKETCSEKSTNLHDYGMLLPCGIDKFR  
GVEFVCCPLAEESDNVDSADAEEDSDVWGGADTDYADGSEDKVVEVAEEEEVAEVEEE  
EADDEDDEDGDEVEEEAEPEYEEATERTTSIATTTTTTTESVEEVVREVCSEQAETGPC  
RAMISRWFYFDVTEGKCAPFFYGGCGGNRNNFDTEEYCMAVCGSAMSQSLLKTTQEPLARD  
PVKLPPTAASTPDAVDKYLETPGDENEHAHFQKAKERLEAKHRERMSQVMREWEAERQA  
KNLPAKADKKAIVQHFQEKVESLEQEAANERQQLVETHMARVEAMLNDRRRLALENYITAL  
QAVPPRPRHVFNMLKKYVRAEQKDRQHTLKHFEHVRMVDPKKAAQIRSQVMTHLRVIYER  
MNQSL SLLYNPVAVAEEIQDEVDELLQKEQNYSDVLANMISEPRISYGNDALMPSLTET  
KTTVELLPVNGEFSLLDLQPWHSFGADSV PANTENEVEPVDARPAADRGLTTRPGSGLTN  
IKTEEISEVKMDAEFRHDSGYEVHHQKLVFFAEDVGSNKGAIIGLMVGGVV IATVIVITL  
VMLKKKQYTSIHHGVVEVDAAVTPEERHLSKMQQNGYENPTYKFFEQMQN
```

Alineamiento de Secuencias



Secuencia:

Cadena lineal finita y ordenada de símbolos pertenecientes a un alfabeto.

Alfabeto:

Conjunto de símbolos.

ADN $A = \{A, C, G, T\}$

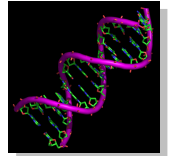
ARN $A = \{A, C, G, U\}$

Proteínas $A = \{a, c, d, e, f, g, h, i, k, l, m, n, p, q, r, s, t, v, w, y\}$

Comparación:

Encuentra posición relativa entre dos secuencias que maximice su parecido. O entre una y muchas.

Alineamiento de Secuencias



Problemas Típicos a Resolver

Alinear dos secuencias (pairwise alignment).

Alinear una secuencia contra muchas en base de datos.

Alineamiento múltiple.

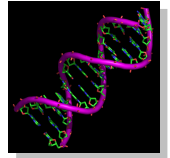
Alineamiento Global: Alinear toda la secuencia A con la B

Ejemplo: Alinear proteínas homólogas de diferentes organismos, identificar regiones de grandes inserciones o borrados, identificar mutaciones en las secuencias...

Alineamiento Local: Buscar las subsecuencias de A y B cuyo alineamiento sea el mejor posible.

Ejemplo: Encontrar un gen en el genoma.

Alineamiento de Secuencias



GAPS

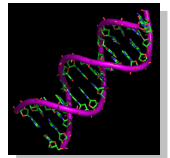
Son aquellas posiciones en las cuales una letra es alineada con un espacio.

Tienen una interpretación evolutiva.

En los métodos de alineamiento se tiene en cuenta que pueden aparecer Gaps, pero se penaliza su aparición.

```
C A T - T C A - C  
C - T C G C A G C
```

Alineamiento de Secuencias



Ejemplo

```
AAB24882 TYHMCQFHCRCRYVNNHSGEKLYECNERSKAFSCPSHLQCHKRRQIGEKTHEHNQCGKAFPT 60
AAB24881 -----YECNQC GKAF AQHSSLKCHYRTHIGEKPYECNQC GKAFSK 40
          ****: .***: * *:** * :****.:* *****..

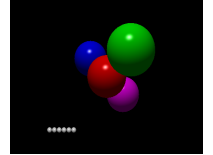
AAB24882 PSHLQYHERTHTGKPYECHQCGQAFKKCSLLQRHKRTHHTGKPYE-CNQC GKAF AQ- 116
AAB24881 HSHLQCHKRTHHTGKPYECNQC GKAF SQHGLLQRHKRTHHTGKPYMNVINMVKPLHNS 98
          **** *:*****:***:*. : .*****          : *.: :
```

Fuente: Wikipedia -

<http://upload.wikimedia.org/wikipedia/commons/8/86/Zinc-finger-seq-alignment2.png>



Definición



General-Purpose computing on Graphics Processor Units

Uso de las GPU como un procesador auxiliar con un alto grado de paralelismo para resolver problemas tradicionalmente resueltos por una o varias CPUs.

Modelos de Programación (Encapsulan la parte gráfica)

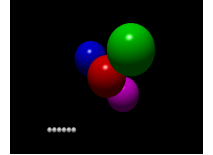
CUDA de nVIDIA
CTM de AMD.

Propietarios y no-abiertos

Glift
GPU++

Abierto. Mundo Académico.
<http://blog.beef.de/thesis/>

Aplicación



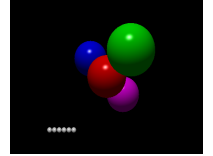
Típicamente algoritmos paralelizables con estructura “la misma acción(kernel) contra diferentes datos” y de forma masiva.

```
int x = 1000
int y = 2000

int* array de x por y

for (i=0;i<x;i++)
{
    for (j=0;j<y;j++)
    {
        ejecutar_accion(i, j, array); // Se ejecuta x * y :: PARALELIZAR
    }
}
```

Definición



CUDA :: Compute Unified Device Architecture

Trabaja con diferentes dispositivos de la marca nVIDIA.
Representación interna de tipos de datos enteros (int).

La GPU se muestra como un dispositivo de computación utilizable para aplicaciones de procesamiento de datos en paralelo.

Acceso a la memoria del dispositivo de forma directa (random access memory device).

Posibilidad de ejecutar un alto número de hilos en paralelo.

CUDA

Los hilos se agrupan en bloques.

Muchos bloques se agrupan en un *grid*.

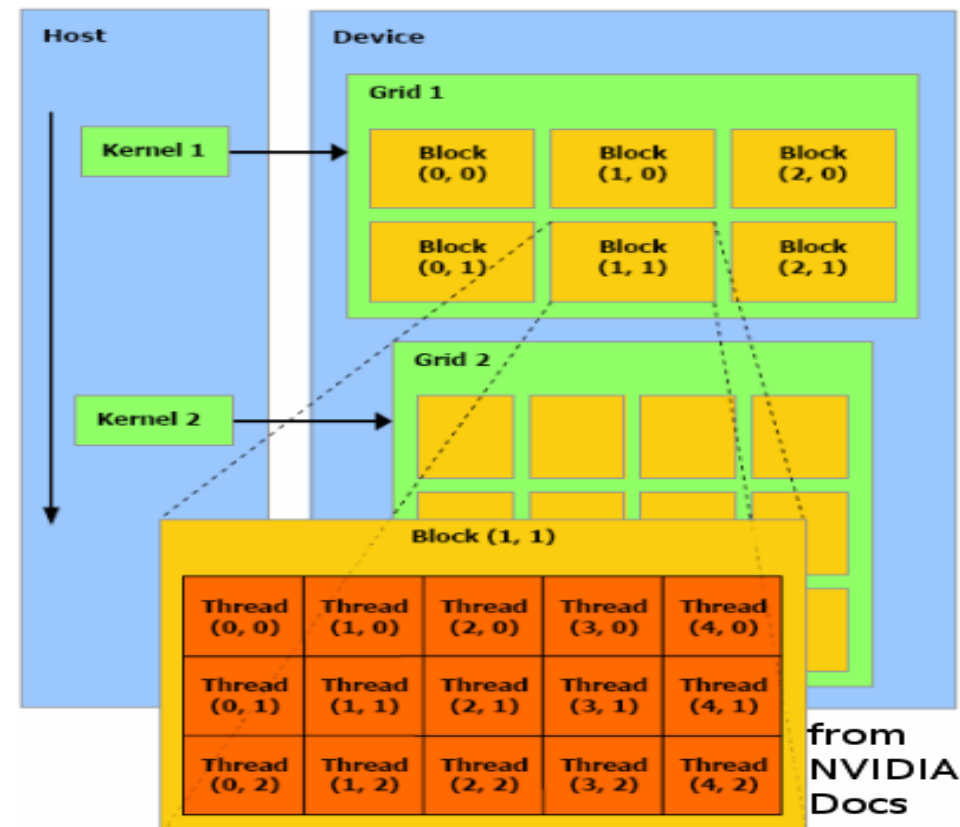
Todos los hilos de un *grid* ejecutan el mismo kernel.

Cada hilo tiene un ID único.

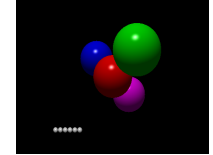
threadID

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

```
-  
float x = input[threadID];  
float y = func(x);  
output[threadID] = y;  
-
```



Ventajas del Modelo



Mucho más barato: Un supercomputador en casa.

Mucho menos complejo que un supercomputador formado por nodos distribuidos.

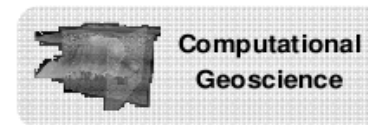
Abstracción sobre la arquitectura de las tarjetas.

“Wrapper” sobre C o quasi-C. Curva de aprendizaje suave.

El programador se concentra en en los algoritmos paralelos.

Sumamos potencia de CPU y GPU.

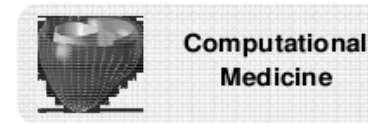
GPU Computing Example Markets



Computational
Geoscience



Computational
Chemistry



Computational
Medicine



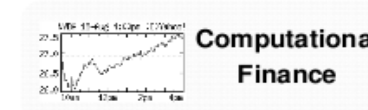
Computational
Modeling



Computational
Science



Computational
Biology



Computational
Finance



Image
Processing

Introducción al Algoritmo

Sirve para realizar alineamiento de secuencias local.

En lugar de mirar a la secuencia completa, compara segmentos de todas las longitudes posibles y optimiza la similitud.

Programación Dinámica: Garantiza que se va a encontrar el alineamiento local óptimo.

Voraz consumidor de recursos de memoria y proceso. **Es lento:**

$O(mn)$ en tiempo y espacio de memoria

En la práctica se usa el algoritmo BLAST:

Más eficiente

No garantiza que se encuentre la solución óptima.



Artwork: Jane Aides, NHGRI

Introducción al Algoritmo



El número de secuencias de DNA y de Proteínas en las base de datos crece de forma exponencial.

Crecimiento mayor que la mejora de los microprocesadores que además son caros.

Los algoritmos **FASTA** (Pearson and Lipman 1988) y **BLAST** (Altschul et al. 1997) son hasta **40 veces más rápidos** que la mejor implementación conocida del algoritmo de Smith-Waterman.

Svetlin A Manavski and Giorgio Valle <http://www.biomedcentral.com/1471-2105/9/S2/S10>

“CUDA Compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment”

De 2 a 30 veces más rápido

Algoritmo

Matriz de Sustitución (W)

Contiene puntuaciones asociadas a cada pareja de elementos.

Aminoácidos (20 - Proteínas)

Nucleótidos (4 - ADN o ARN)



Artwork: Jane Ades, NHGRI

La puntuación valora el coste evolutivo de cambiar una letra por otra.

Matrices PAM : Proporcional a: $\log(F_{ab}/(F_a \times F_b))$

F_{ab} : Frecuencia del par AB en alineamiento relacionados

$F_a \times F_b$: frecuencia del par AB en alineamiento no relacionados

Probabilidad de que el alineamiento sea auténtico

$W(ij) = \frac{\text{Probabilidad de que el alineamiento sea auténtico}}{\text{Probabilidad de que el alineamiento sea aleatorio}}$

Probabilidad de que el alineamiento sea aleatorio

Algoritmo

Matriz de Sustitución (W) PAM250 Proteínas



C	12																			
S	0	2																		
T	-2	1	3																	
P	-3	1	0	6																
A	-2	1	1	1	2															
G	-3	1	0	-1	1	5														
N	-4	1	0	-1	0	0	2													
D	-5	0	0	-1	0	1	2	4												
E	-5	0	0	-1	0	0	1	3	4											
Q	-5	-1	-1	0	0	-1	1	2	2	4										
H	-3	-1	-1	0	-1	-2	2	1	1	3	6									
R	-4	0	-1	0	-2	-3	0	-1	-1	1	2	6								
K	-5	0	0	-1	-1	-2	1	0	0	1	0	3	5							
M	-5	-2	-1	-2	-1	-3	-2	-3	-2	-1	-2	0	0	6						
I	-2	-1	0	-2	-1	-3	-2	-2	-2	-2	-2	-2	-2	2	5					
L	-6	-3	-2	-3	-2	-4	-3	-4	-3	-2	-2	-3	-3	4	2	6				
V	-2	-1	0	-1	0	-1	-2	-2	-2	-2	-2	-2	-2	2	4	2	4			
F	-4	-3	-3	-5	-4	-5	-3	-6	-5	-5	-2	-4	-5	0	1	2	-1	9		
Y	0	-3	-3	-5	-3	-5	-2	-4	-4	-4	0	-4	-4	-2	-1	-1	-2	7	10	
W	-8	-2	-5	-6	-6	-7	-4	-7	-7	-5	-3	2	-3	-4	-5	-2	-6	0	0	17
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W

Algoritmo

Ejemplo de Alineamiento



	A	E	G	H	W
A	5	-1	0	-2	-3
E	-1	6	-3	0	-3
H	-2	0	-2	10	-3
P	-1	-1	-2	-2	-4
W	-3	-3	-3	-3	15

- Penalización por iniciación del gap: -8
- Penalización por extensión del gap : -8

```

HEAGAWGHE-E
      | | | |
--P-AW-HEAE
    
```

$$(-8) + (-8) + (-1) + 5 + 15 + (-8) + 10 + 6 + (-8) + 6 = 9$$

```

HEAGAWGHE-E
      | | | |
P-A--W-HEAE
    
```

$$(-2) + (-8) + 5 + (-8) + (-8) + 15 + (-8) + 10 + 6 + (-8) + 6 = 0$$

Algoritmo

El Problema:

Alinear dos secuencias. La Objetivo y una de la base de datos (N veces).

Entrada: Dos secuencias A y B (~2000 elementos)

Matriz de Sustitución (W)

Penalización por inicio de Gap (d).

Penalización por extensión de Gap.

Salida: El alineamiento óptimo de las dos secuencias.

Alineamientos Globales

posibles :

$$\binom{2n}{n} = \frac{(2n)!}{(n!)^2} \approx \frac{2^{2n}}{\sqrt{(\pi n)}}$$



Artwork: Jane Adams, NHGRI

¡Con n=20 tenemos 120 millones de alineamientos posibles!

Algoritmo

Se basa en una matriz de alineamiento H ($d = -4$).

$$H(i,j) = \max \left\{ \begin{array}{l} 0 \\ H(i-1,j-1) + W(A_i, B_j) \\ H(i-1,j) + d \\ H(i, j-1) + d \end{array} \right. \quad W = +5 \quad != -3$$



	-	G	A	A	T	T	C	A	G	T	T	A
-	0	0	0	0	0	0	0	0	0	0	0	0
G	0	5	1									
G	0											
A	0											
T	0											
C	0											
G	0											
A	0											

	-	G	A	A	T	T	C	A	G	T	T	A
-	0	0	0	0	0	0	0	0	0	0	0	0
G	0	5	1	0								
G	0											
A	0											
T	0											
C	0											
G	0											
A	0											

Algoritmo

Una vez que tenemos la matriz rellena:

Hacemos “TraceBack”

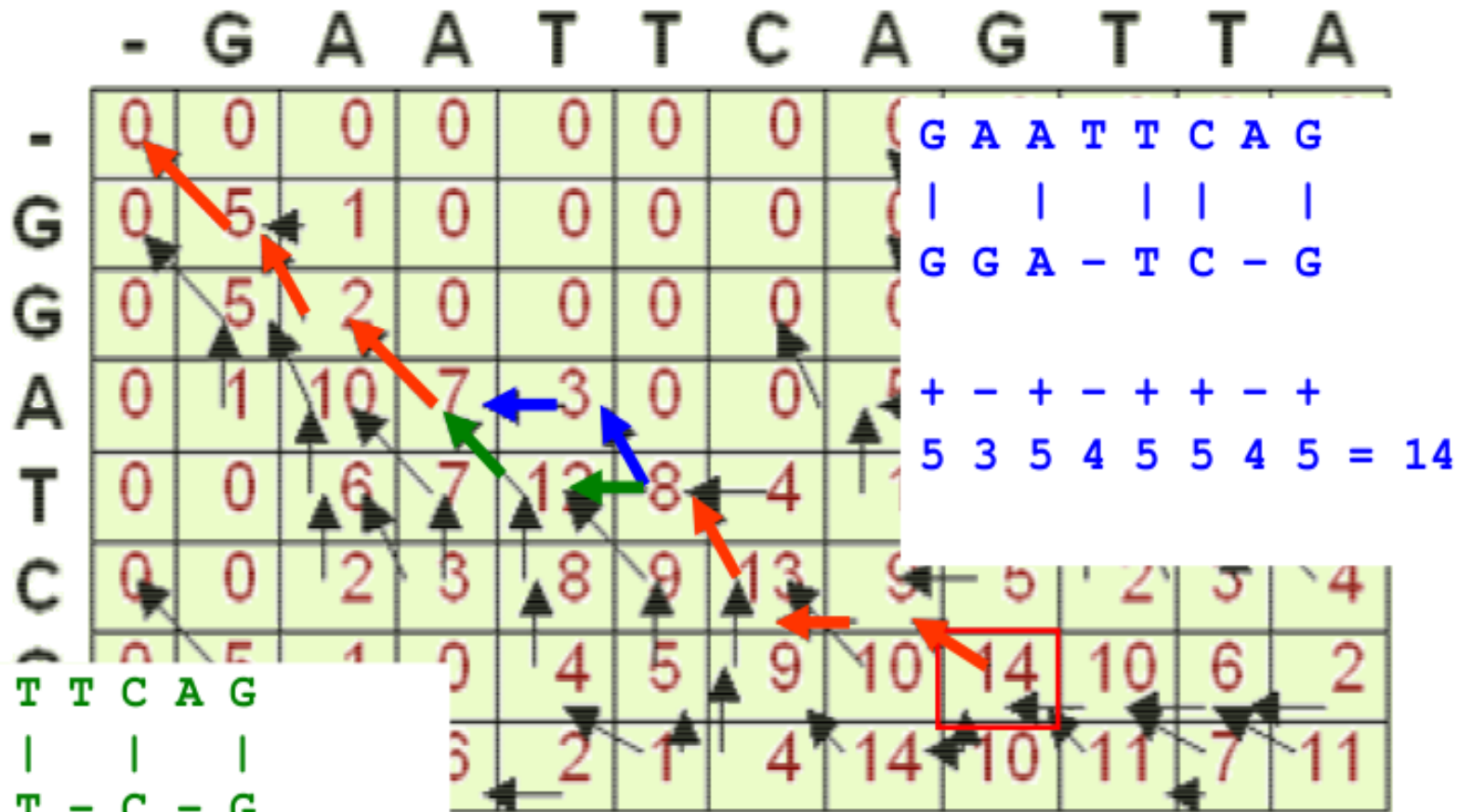


- Se comienza en la celda con mayor puntuación.
- En cada celda observamos hacia dónde movernos de acuerdo a los punteros que hemos guardado (de donde vengo).
- Cuando llegamos a una celda en la que no aparecen punteros que nos indiquen hacia dónde movernos paramos (comienzo del alineamiento).
- Puede que el máximo se encuentre en dos o más celdas. Hay más de un alineamiento que produce puntuación óptima.

Trace-Back



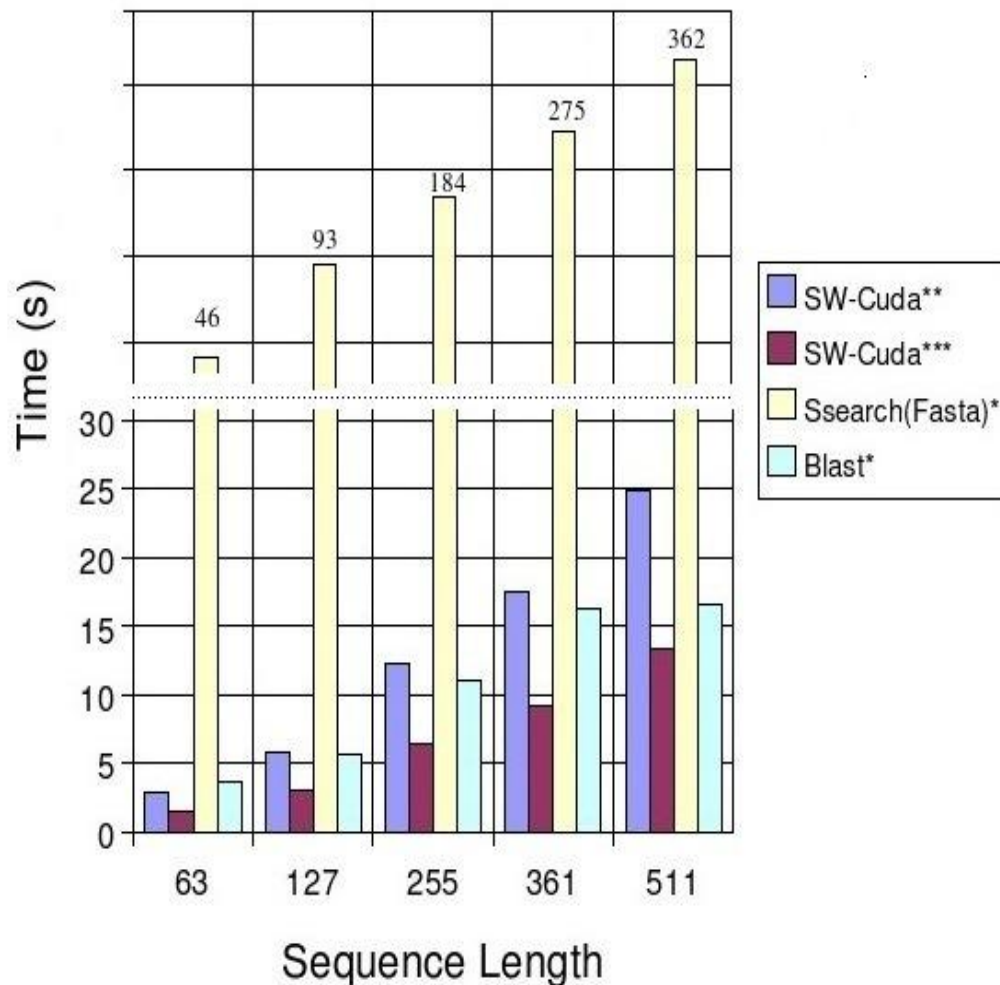
Artwork: Jane Adee, NHGRI



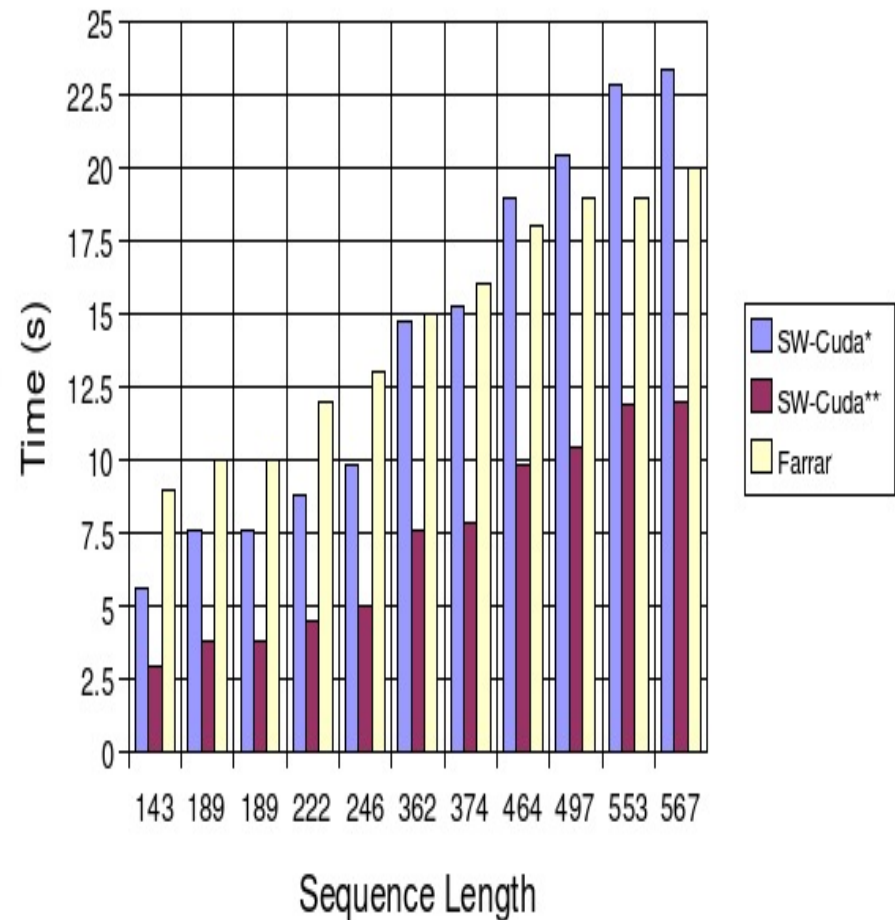
G A A T T C A G
 | | | | |
 G G A T - C - G
 + - + + - + - +
 5 3 5 5 4 5 4 5 = 14

Resultados en CUDA (Manavsky)

Time Performances



Time Performances



Introducción al Algoritmo



- Desplazamos una secuencia debajo de la otra.
- Contamos el número de coincidencias.
- Seleccionamos el máximo.

A: ACGTACGTACGTACGTACGT I=0

B: AAGGCCTTAAGGCC

A: ACGTACGTACGTACGTACGT I=1

B: AAGGCCTTAAGGCC

.....

A: ACGTACGTACGTACGTACGT I=4

B: AAGGCCTTAAGGCC

Introducción al Algoritmo

Desplazar una secuencia debajo de la otra equivale a recorrer una matriz por sus diagonales $O(N \cdot M)$.



	A	C	G	T	A
C		X			
G			X		
T				X	
A	X				X

A C G T A
C G T A

A C G T A
C G T A

A C G T A
C G T A

A C G T A
C G T A

A C G T A
C G T A

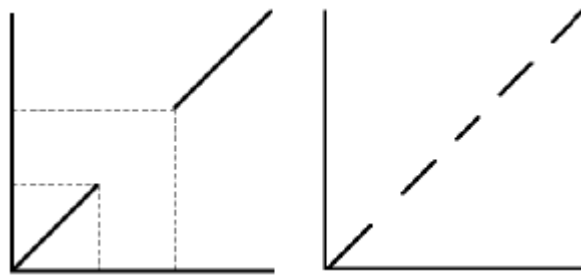
A C G T A
C G T A

Usos

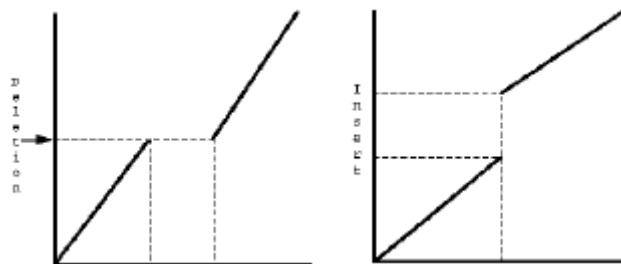
Mostramos un punto donde haya coincidencias.

Si son idénticas habrá diagonales definidas.

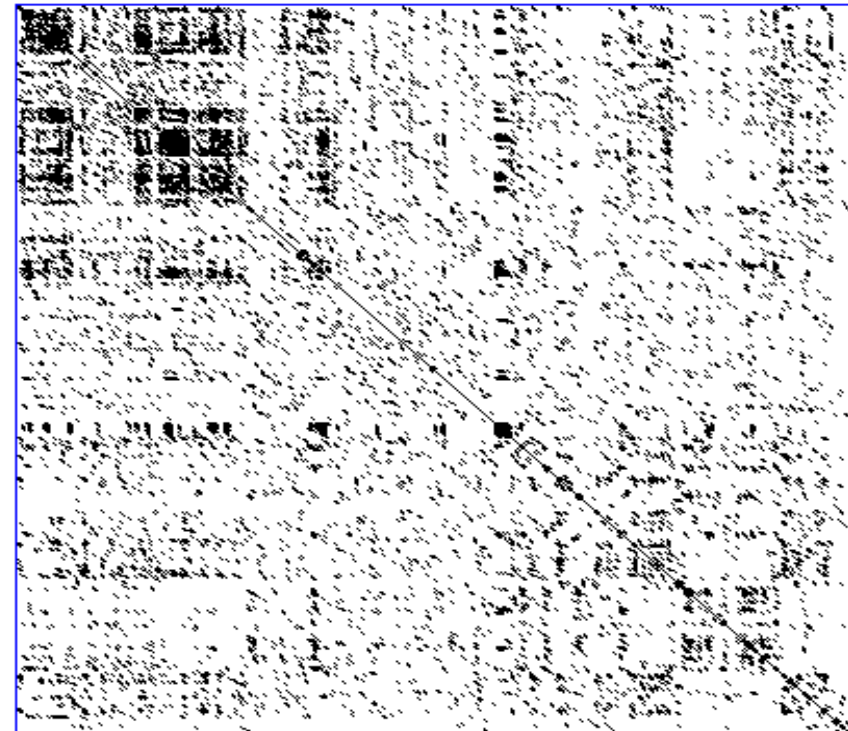
Cuanto más diferentes, más difusas serán las diagonales.



Secuencias que han divergido



Delecciones e inserciones



WikiPedia

Pseudo Código :: CPU

```
// Pueder ser n= 1024;  
//           m = 2048;  
  
cargar_secuencia(h_a, n);  
cargar_secuencia(h_b, m);  
  
crear_matriz(h_W,n,m);  
  
for (i=0;i<n;i++)  
{  
    for (j=0;j<m;j++)  
    {  
        h_W[i*j] =(h_a[i] == h_b[j]);  
    }  
}  
  
mostrar_matriz(h_W, n, m);
```



Artwork: Jane Aides, NPGRI

Pseudo Código :: GPU

```
cargar_secuencia(h_a, n);
int* d_a = 0;
cudaMalloc((void**)&d_a,n);
cudaMemcpy(d_a, h_a, n,
           cudaMemcpyHostToDevice);

cargar_secuencia(h_b, m);
int* d_b = 0;
cudaMalloc((void**)&d_b,m);
cudaMemcpy(d_a, h_a, n,
           cudaMemcpyHostToDevice);

crear_matriz(h_W,n,m);
int* d_W = 0;
cudaMalloc((void**)&h_W,n * m);
cudaMemset(d_W,0,n*m);

cal_dotplot << dimGrid, dimBlock >>
           (d_a, d_b,d_W,n,m);

synchThreads();
cudaMemcpy(h_W, d_W, n*m,
           cudaMemcpyDeviceToHost);

mostrar_matriz(h_W,n,m);
cudaFree(d_a); // y d_b y d_W....
free(h_a);    // y h_b y h_W....
```

```
dim3 dimBlock(blksiz, blksiz);
dim3 dimGrid(n / blksiz, m /blksiz);

__global__ void cal_dotplot
           (int* d_a, int* d_b, int* d_W,
            int n ,
            int m)
{
    int i = blockIdx.x * blockDim.x +
            threadIdx.x;
    int j = blockIdx.y * blockDim.y +
            threadIdx.y;
    int index = i + j * n;

    if (i < n && j < m)
        d_W[index] = (d_a[i] == d_b[j]);
}
```

La Bioinformática es un campo fértil en problemas a resolver relacionados con la computación.

- Crecimiento exponencial de datos.
- Búsquedas de secuencias y patrones.
- Representación 3D de Moléculas: Comportamiento inferido.
- Búsquedas en texto no estructurado (publicaciones y “abstracts”)

El uso de GPUs para procesamiento genérico de procesos paralelos:

- Acerca la investigación a entidades sin grandes recursos.
- Rendimientos comparativos de 5 a 10 veces más óptimos.
- Muchas publicaciones recientes.
- Nuevos modelos y herramientas de encapsulación.

Revisar y adaptar algoritmos existentes y nuevos para su procesamiento masivo paralelo.

- Solución a problemas de recursos y de complejidad de administración
- Aplicaciones inmediatas en Biología Molecular

- [1] **CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment.**
Svetlin A Manavsky and Giorgio Valle - 2008
CRIBI, University of Padova, Italy

- [2] **High-throughput sequence alignment using Graphics Processing**
Michael C Schatz, Cole Trapnell, Arthur L Delcher, Amitabh Varshney - 2007
Center for Bioinformatics and Computational Biology, University of Maryland, USA

- [3] **Sequence Alignment**
WikiPedia
http://en.wikipedia.org/wiki/Sequence_alignment

- [4] **Smith-Waterman Algorithm**
WikiPedia
http://en.wikipedia.org/wiki/Smith-Waterman_algorithm

- [5] **Fast exact string matching on the GPU**
Michael C Schatz, Cole Trapnell- 2007
CMSC 740 Computer Graphics

- [9] **Material Bioinformática: Curso y máster en investigación**
Profesor Alberto Pascual Montano
Departamento de Computadores y Automática - U.C.M.
<http://www.dacya.ucm.es/apascual/bioinformaticaDoctorado.html>

¿Preguntas?

jlmarina [at] gmail [-dot-] com