

Tarea Programada MPI

Análisis de Algoritmos

1 Introducción

En esta tarea ud. debe implementar el algoritmo de Needleman-Wunsch para alineamiento de secuencias en dos versiones, una versión secuencia que corre en un solo procesador, y otra versión que corre en varios procesos utilizando el Message Passing Interface (MPI). Opcionalmente puede implementar una versión en paralelo que calcule y devuelva el alineamiento óptimo.

2 Needleman-Wunsch

El algoritmo de Needleman-Wunsch se puede encontrar en:

http://en.wikipedia.org/wiki/Needleman-Wunsch_algorithm, aquí les presentamos un breve resumen.

2.1 especificación

El algoritmo recibe como entrada:

- dos hileras h_1 y h_2 tal que el largo de h_1 es $n = |h_1|$ y el largo de h_2 es $m = |h_2|$, estas hileras se obtienen de un alfabeto que en nuestro caso asumiremos que es $\Sigma = \{A, C, G, T\}$, alfabeto típico para el secuenciamiento de genes.
- una tabla S de tamaño $|\Sigma| \times |\Sigma|$ que contiene en la entrada S_{ij} el costo de sustituir el carcter i por el carcter j .
- una penalización d para cuando un caracter se inserta o elimina.

El algoritmo propiamente hace lo siguiente:

```

for i=0 to length(h1)
    F[i,0] = d*i
for j=0 to length(h2)
    F[0,j] = d*j
for i=1 to length(h1)
    for j=1 to length(h2)
    {
        Match = F[i-1,j-1] + S[h1[i], h2[j]]
        Delete = F[i-1, j] + d
        Insert = F[i, j-1] + d
        F[i,j] = max(Match, Insert, Delete)
    }

```

Para nuestro caso, puede asumir que la penalización por insertar o eliminar es -1, que el costo de comparar valores distintos es -1 también y que el premio de acertar una alineación (comparar G con G por ejemplo) es de 1.

Si es así, el algoritmo sólo recibe como entrada las dos hileras y queda el siguiente código

```

for i=0 to length(h1)
    F[i,0] = -i
for j=0 to length(h2)
    F[0,j] = -j
for i=1 to length(h1)
    for j=1 to length(h2)
    {
        Match = F[i-1,j-1] + (h1[i] == h2[j] ? 1 : -1)
        Delete = F[i-1, j] - 1
        Insert = F[i, j-1] - 1
        F[i,j] = max(Match, Insert, Delete)
    }

```

3 Problema

Usted debe escribir un programa en C que utilice MPI y resuelva este problema para hileras arbitrarias, asuma que la entrada del programa son dos archivos donde el primero contiene la hilera h1 y el segundo la hilera h2, asuma también que las hileras solo contienen los valores A, C, G o T. La

última suposición que debe hacer es que la cantidad de procesos que ejecuta el MPI divide tanto el largo de la hilera h1 como el largo de la hilera h2.

Así si convoca su programa con:

```
> ./mpiexec -n 8 needleman-wunsch h1.txt h2.txt
```

Entonces 8 debe dividir el largo de el texto contenido en h1, y debe dividir el largo del texto contenido en h2.

Si sus hileras miden 800 caracteres h1 y 1000 caracteres h2, entonces debe dividir su matriz $F[800,1000]$ en bloques de tamaño 100x125.

El resultado de su programa solo debe ser el mejor puntaje obtenido de la comparación entre h1 y h2.

4 Entregables

Código del programa con comentarios, la tarea se puede hacer en grupos de no más de 3 personas, se evaluará en citas de revisión, martes y miércoles por la mañana.