

Relatório busca exaustiva GPU (Thrust) e OpenMP.

O código utilizado para construir o paralelismo multi core foi uma refatoração do código utilizado para busca exaustiva paralela somente para OpenMP na parte de geração de subsequências. Ao invés de utilizar uma função recursiva para gerar as subsequências foram utilizados um três loops que garantiram a geração das subsequências.

```
while (dnas_first_seq_cpu_flexible.size() > 0) {
    while(k > 0) {
        host_vector<char> next_subsequence;
        for (int j = 0; j < k; j++) {
            next_subsequence.push_back(dnas_first_seq_cpu_flexible[j]);
        }
        k = k - 1;

        subsequences_a.push_back(next_subsequence);
        next_subsequence.clear();
    }
    dnas_first_seq_cpu_flexible.erase(dnas_first_seq_cpu_flexible.begin());
    k = dnas_first_seq_cpu_flexible.size();
}
```

Como observado no código acima criou-se um loop entre zero e a quantidade de elementos de um vetor de dnas. O primeiro elemento é removido toda vez que é gerada todas as subsequências possíveis com este primeiro elemento.

Com a execução do código foi possível perceber um ganho muito grande com relação à implementação paralela (com o OpenMP) e a serial (sem o OpenMP) sendo que é possível rodar para 2000 elementos em 200 segundos, o mesmo tempo que a implementação serial rodou para entrada de 60 elementos.