

# Smith-Waterman Algorithm X10 implementation

---

## List of files (source code)

- SmithWaterman.x10 (The simple sequential version)
- SmithWatermanParallalBlockwise.x10 (The by-row blockwise distributed parallel version)
- SmithWatermanParallalTaskDAG.x10 (The DAG dependency using X10 activity parallel version)
- SmithWatermanParallalTaskDAGBlockwise.x10 (The DAG dependency and block combined parallel version)

## Steps of testing

### Compile

```
$ x10c++ -o SW SmithWaterman.x10
$ x10c++ -o SWP1 SmithWatermanParallalBlockwise.x10
$ x10c++ -o SWP2 SmithWatermanParallalTaskDAG.x10
$ x10c++ -o SWP3 SmithWatermanParallalTaskDAGBlockwise.x10
```

### Setting up environment

```
$ export X10_MAX_THREADS=10000
```

Setting the **X10\_MAX\_THREADS** environment variable to 10000 (default 1000) enable X10 to spawn more user threads (activities). To **SmithWatermanParallalTaskDAG** with larger data requires this change.

### Run

The program do not take in any parameters, but it will prompt the user to enter the file names, and penalty scores.

```
$ ./SW
Input the FASTA_FILE_1 FASTA_FILE_2 MATCH_FILE GAP_OPENING_PANALTY
GAP_EXTENSION_PANALTY
2k1 2k2 BLOSUM62 2 1
Identity: 635/2598 (0.244418783679754)
Gaps: 1225/2598 (0.471516551193226)
Score: 2217
.....
(More output omitted)
```

Note that the user input above was **2k1 2k2 BLOSUM62 2 1**, which are the first fasta file, second fasta file, blosum file, gap open penalty and gap entention penalty.

## Test with random sample data given

```
$ time ./SW < samplein1k
```

The sample test cases are named with `sampleinXk`. `X` is the length of the sequence in fasta files (in thousands).

Size of `X.Y k` is encoded as `X_Y`. For example, a sample run with `1.25k` (1250) length sequence will be provided in `samplein1_25k`.

The sample data varies from 1k to 100k. Note that not all sample tests are runnable on all platforms, due to memory constraints and program limitation. `SmithWatermanParallelTaskDAG` may only be able to run sample up to size 2k.

To time the program, use `time` or other linux command.

## Test with different number of threads

```
$ export X10_NTHREADS=8
```

Change `8` to other numbers to change the number of X10 threads.

## Additional note

The fasta file need to have a empty line at the end before EOF. Otherwise the scanning might have problem.

## List of available sample tests

- samplein1k
- samplein1\_25k
- samplein1\_5k
- samplein1\_75k
- samplein2k
- samplein4k
- samplein6k
- samplein8k
- samplein10k
- samplein12k
- samplein50k
- samplein75k
- samplein100k