

KlastRunner

Use KLAST from the command-line

KlastRunner is a software designed to the run KLAST high-performance sequence comparison engine from the command-line. The software is well-suited to run on cluster infrastructures.

KlastRunner main features

KlastRunner performs:

- KLAST sequence comparisons
- data filtering using user-defined constraints
- sequence data integration (i.e. feature tables, biological classifications, etc.) within KLAST results, i.e. producing fully annotated KLAST results.

KlastRunner directly works on Fasta files. However, we provide our *Databank Manager* to facilitate the installation of sequence databanks from public and personal databanks.

Available as a command-line tool, KlastRunner can be integrated within any existing workflow execution systems. It is fully compatible with major batch-queueing systems for distributed resource management.

```
usage: where options include:
-a <arg>          Number of cores to use (max given by
                  license)
-annot <arg>      Enable integration of annotation within
                  results. Accepted value: bco or none
                  (default)
-b <arg>          bandwidth for small gapped extension
-d <arg>          Subject database file name (Fasta)
-dbairror <arg>   File name to the KDMS dbairror
                  configuration file
-e <arg>          Cost to extend a gap
-f <arg>          Expectation value threshold
-g <arg>          Apply a low-complexity filter on query
                  sequence (SEG or DUST)
-G <arg>          Cost to open a gap
-g <arg>          threshold for small gapped extension
-H,--max-hsp-per-hit <arg> Maximum alignments per hit. 0 value will
                           dump all hits (default)
-h,--help          print this help message
-i <arg>          Query database file name (Fasta) or KDMS
                  databank name
-M <arg>          Score matrix (BLOSUM62 or BLOSUM50)
-m,--max-database-size <arg> Maximum allowed size (in bytes) for a
                              database. If greater, database is
                              segmented (see comment, below).
-n <arg>          Size of neighbourhood performing ungapped
                  extension
-o <arg>          KLAST report output file name
-p,--PROGRAM <arg> Program name. Accepted value: plastp,
                  plastn, tplastn, plastx or tplastx.
-Q,--max-hit-per-query <arg> Maximum hits per query. 0 value will dump
                              all hits (default)
-q <arg>          Penalty for a nucleotide mismatch (plastn)
-r <arg>          Reward for a nucleotide match (plastn)
-S,--seeds-use-ratio <arg> Ratio of seeds to be used (see comment,
                              below). [1..100], default is 100. Decrease
                              value to speedup algorithm with little
                              loss of quality.
-s <arg>          Ungapped threshold trigger a small gapped
                  extension (see comment, below). [25..127],
                  default is 28 and 55 for protein-based and
                  nucleic-based comparisons, respectively.
                  Increase value to speedup algorithm with
                  little loss of quality.
-stats <arg>      KLAST statistics Output File
-strand <arg>     Strands for plastn: 'plus', 'minus' or
                  'both' (default)
-X <arg>          X dropoff value for gapped alignment (in
                  bits) (zero invokes default behavior)
-xmlfilter <arg>  File name of KLAST filter.
-Z <arg>          X dropoff value for final gapped alignment
                  in bits (0.0 invokes default behavior)

Klast specific parameters to speed-up comparison:
"-seeds-use-ratio", "-s" and "-max-database-size"
for more information, go to : http://www.korilog.com/index.php/KLAST-algorithm.html
```

KlastRunner scalability

The KLAST algorithm relies on a unique software architecture taking full advantage of recent multi-core computers. When running on a cluster infrastructure, KLAST can be finely tuned to benefit from the available computing power.

Comparison of 22.8k cDNA and protein sequences from *Bos taurus* and *Xenopus tropicalis* on a Dell PowerEdge 16-core Intel Xeon computer. Upper table illustrates CPU usage. Lower table displays the running time (seconds) of a single KLASTp job on 8 and 16 cores vs. 8 and 16 KLAST jobs, each of them running on a single core (unit-KLASTp).

Cores	Blastx	KLASTx
1	100%	100%
2	94%	98%
4	64%	95%
8	32%	89%
16	16%	75%

Cores	KLASTp	unit-KLASTp
8	468	511
16	280	290

Benchmark data courtesy of Pierre Pericard and Christophe Caron, ABIMS platform/CNRS-UPMC, Station Biologique de Roscoff (France)

System Requirements

- Starter configuration:
- desktop computer
 - Intel 64-bit quad-core
 - 8 Gb RAM
 - Linux or MacOS X (64-bit required)
 - Java Runtime 1.6 (64-bit required)

Contact us at klast@korilog.com for more information

KoriLog BIOINFORMATICS Solutions
4, rue Gustave Eiffel – 56230 Questembert - France
Tél : +33 960 368 038
www.korilog.com - klast@korilog.com

KLAST software development by

