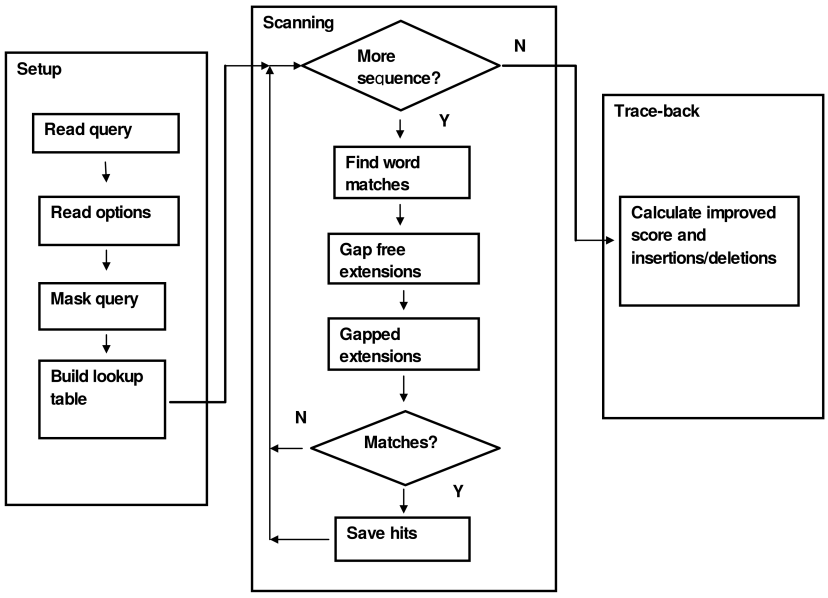
December 16, 2016

Abstract

1 Introduction

Basic Local Alignment Search Tool (**BLAST**) algorithm was introduced in early 1990 and compares pairs of words of length k matching a threshold T. If matching occurs, an extension step is then performed. The algorithm includes 3 steps:

1. **Setup**: prepares a lookup table from a query sequence
2. **Scanning**: word matching with ungapped and gapped extensions
3. **Trace-back**: produces final alignment and calculates e-values



BLAST is highly parametrizable such as the value of k but in pratice, default parameters will suit most of the applications. Setting up a BLAST search involve rather formatting of the output. Execution speed is proportional to the product of query length and the database searched and also depends of T (higher values increase speed but may miss weaker similarities).

Terminology

- Query: input sequence to be checked
- Database: set of sequences against which the query is scanned
- Subject: a sequence in the database with matching words in the query
- Entry: name of a sequence in the database
- HSP: High-scoring segment pairs, aggregated segments with maximal score which cannot be improved by further extension
- e-value: expected matching value of words in the database
- Hitscore: depends on %overlap and identity of the matching

2 Installation of BLAST+

The blast+ Suite is available on NCBI ftp server. On Linux environments, a simple call with `sudo apt-get install ncbi-blast+` will do the necessary. For windows users, download the executable from NCBI ftp server and an environment variable should be set so that Blast programs may be called from any directory (without adding the full path). Instructions are found at `https://www.ncbi.nlm.nih.gov/books/NBK52637`. Because we didn't have easy access to FTP for today, please download the `.exe` file at `http://exotic.univ-tours.fr/im7yhzl5`.

Every program in the /bin folder has its own documentation which can be accessed with the `-help` or `-h` argument. For example:

```
makeblastdb -help
```

All default values are indicated in the full help.

3 How to use BLAST+ to find homologies in a local database?

1st step: prepare your own database Database formatting is the first and obligate step before lauching Blast analyses. The `makeblastdb` tool will convert fasta files into `nucl` or `prot` databases. The argument `-parse_seqids` will be useful to retrieve target sequences in the database.

```
makeblastdb -dbtype nucl -in test.fa -parse_seqids
```

2nd step: your query (Nucleotide vs nucleotide) Once you've formatted your database, you have to store your query into a file. Here we will paste a sequence into a file named `seq_to_blast.fa`.

```
blastn -db test.fa -query seq_to_blast.fa
```

This command will display all hits above the default threshold. The display is not very convenient as it is directly written to standard output. An output file (`-out` argument) may be indicated to permanently store the results.

```
blastn -db test.fa -query seq_to_blast.fa -out blast_results
```

Ouput may be formatted in several ways, eg. tabular, xml, etc. using the `-outfmt` argument. This argument takes an integer corresponding to a given output format (eg. 6 for a tabular format).

```
blastn -db test.fa -query seq_to_blast.fa -outfmt 6
       -out blast_results_table
```

If too many hits are found, you may either increase the e-value threshold (`-evalue`) and/or display only the best hit for each query sequence (`-max_target_seqs`).

```
blastn -db test.fa -query seq_to_blast.fa -outfmt 6 -evalue 1e-20
       -max_target_seqs 1 -out blast_results_table
```

If not hit found, then your query was probably set too stringent. By default, blastn executes a `megablast` task which is dedicated to highly homologous sequences. You therefore may run a `blastn` search by setting the `-task` parameter:

```
blastn -db test.fa -query seq_to_blast.fa -outfmt 6 -evalue 1e-20
       -max_target_seqs 1 -out blast_results_table -task blastn
```

3rd step: work with subject sequences Once you've found homologous sequences in your database, it may be interesting to retrieve the corresponding hit. **Blast+** tool `blastdbcmd` is dedicated to the controlled retrieval of one or more target sequences.

```
blastdbcmd -db test.fa -entry seq_to_search
```

If several hits have to be retrieved, you may either add them in the command line with a comma separator or store the identifier into a file.

```
blastdbcmd -db test.fa -entry seq_to_search1,seq_to_search2,seq_to_search3
blastdbcmd -db test.fa -entry_batch file_containing_ids
```

When querying against a genome sequence, you might also be interesting in getting only part of the scaffold matching with the query (for example, the 1000 pb before start codon...). Use the `-range` argument to extract a particular range from a sequence in the database.

```
blastdbcmd -db test.fa -entry seq_to_search1 -range 1000-1500
```