OPSCAN (Version 1.0)

Presentation

Opscan is intended to scan a set of protein sequences (the database, supposed to be a set of ordered proteins from contiguous genes on a genome) with another set of query sequences (the query, supposed to be a set of an operon protein sequences) in order to find genomic sequences similar to the operon sequences and localised in the same area on the genome.

Throughout this document, the database sequences will be called "genomic genes", and the query sequences will be called "operon genes".

The default strategy used in Opscan is°:
-1- scan the genomic genes database with the fastp algorithm in order to find for each operon gene a set of K most similar genomic genes (K is a user defined parameter, it defaults to 6).

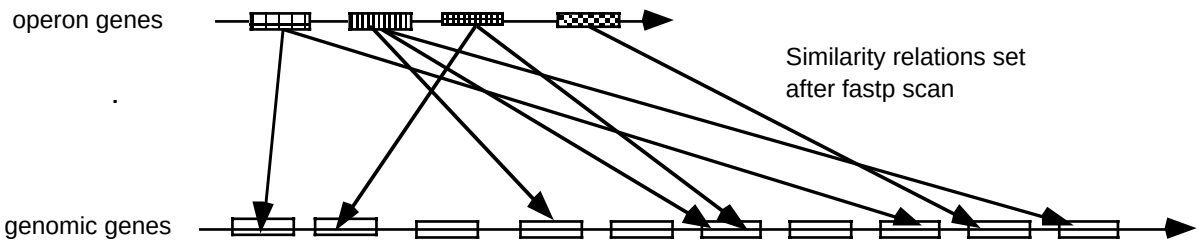
-2- refines this similarity search by performing a dynamic programming alignment (NWS - global alignment) or 'Bestfit' search (zero cost end gaps) on each operon gene with its K most similar genomic genes.

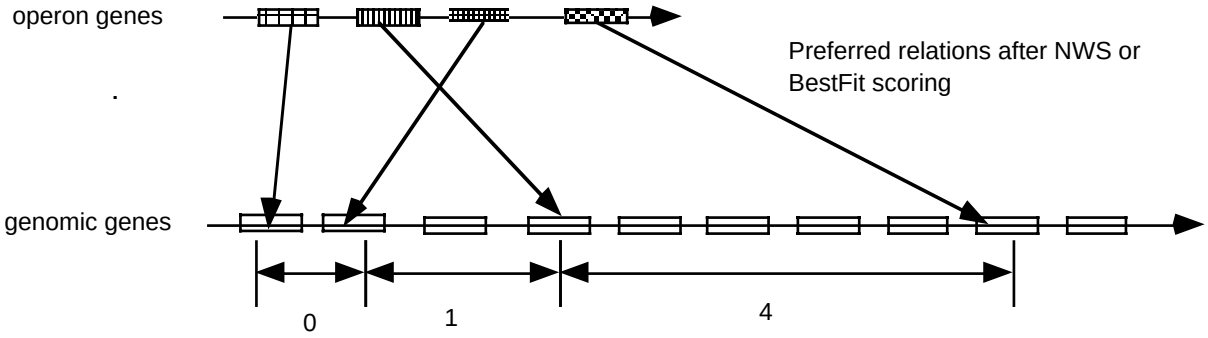
-3- then, for each operon gene, Opscan finds - if it exists - its "preferred" genomic gene within its K most similar genomic genes. Here, "preferred" means a bijective relation between a genomic gene - G_j - and an operon gene - O_i - that is established when the most similar database genomic gene with operon gene O_i is G_j, and the most similar query operon gene with genomic gene G_j is O_i (in that case, the homology between these genomic and operon genes is more probable).

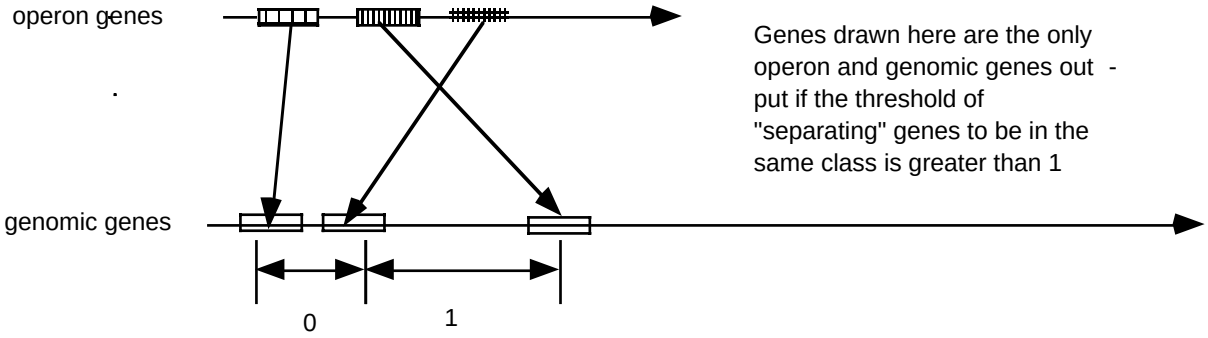
-4- outputs all the subsets of genomic genes "preferred" by operon genes, which lie in the same region (the "region" is user defined, see below).

Notes:
- The first step can be omitted, and then, the similarity between genomic and operon genes will be established only by NWS or BestFit algorithms (the program will then run slower as dynamic programming alignment algorithm are more time consuming as fastp similarity searches).
- In the fourth step, one can choose to output all the most similar genomic genes whatever their relative positions on the genome are.

Here is a scheme of the different stages in opscan°:







Similarity scanning with the fastp algorithm

In Opscan, the first similarity scan is usually (and by default) made with fastp algorithm. In the fastp algorithm, the similarity score is those of the alignment obtained from of the "best shift" of the query sequence in respect to the database sequence. This "best shift" is the shift between the two sequences that gave the maximum number of matching amino-acids k-uple (with no gaps, i.e. sequence are like slid one over the other). A given shift is called a "diagonal". Following shows a shift of +2 positions of the database sequence with respect to the query sequence°:

```
query:              PLHIMMPNGNVM
database:             HIMMPNNVM
matching 2-uples      12345
```

There is five 2-uple matches when applying this shift to the database sequence with respect to the query sequence. This shift gave the maximum (5) 2-uple matches over all possible

shifts. The fastp similarity score is computed as the similarity score of all aligned pairs of amino-acids in the sequence (i.e. sequence "free ends" are not taken into account in computation of the similarity score).
In the example, here is the following alignment corresponding to the best shift (-2):

```
query:                HIMMPNGNV
database:             HIMMPNNVM
```

Note: alignment scores are computed as the sum of the score of amino-acid pair in the alignment and the amino-acid pair scores are these found in the substitution score matrix used. PAM250 or BLOSUM62 are such substitution score matrices. If no substitution score matrix is specified, identity matrix is used, and the score is then the percentage of amino-acid identity of the partial alignment. This score is normalized in the 0-100 range.
K-uples length˚: The k parameter which fixes the length of the k-uples can be changed (see the -k option, below). The k parameter defaults to 2 (as usual for protein sequences). Sensitivity is increased with decreasing the value of k.
When "-x" option is on, scores are computed as the number of matching k-uples for the best shift. In that case, the score is more "rough" as it does not count resemblance between similar amino-acid or single amino-acid matches appearing "outside" matching k-uples. In the above example, if the -x option would be set, the score would have a value of 5.
Diagonal integration˚: when option "-x" is on, in order to get a score which could take into account gaps in some sense (i.e. diagonals that could represent matching regions shifted from the best one by one or more gaps), one can add to the number of matching k-uples obtained for the best diagonal, the number of matching k-uple of neighbour diagonals (see the -i option below).

```
query:                      PLHIMMPNGNVM
best database shift           HIMMPNNVM       (5 matches)
best database shift -1:        HIMMPNNVM      (2 matches)
best database shift +1:      HIMMPNNVM        (0 matches)
```

In the first shift, there are 5 matching 2-uples, HI, IM, MM, MP and PN. In the second shift, there are 2 matching 2-uples, NV and VM. If no integration is done, the score will be of 5 matches as the first diagonal is the maximum 2-uple matching shift. If diagonal integration of 1 is done, the score will be of 7 matches, leading to a more sensitive similarity search as some gaps (here, one gap) could be taken into account in the score as shown by the following alignment of these 2 sequences˚:

```
query:               PLHIMMPNGNVM
database:              HIMMPN-NVM
```

In Opscan, fastp scores are normalised between 0 and 100˚: 100 would be the score of the longest sequence with itself and is then the best feasible score.
As K most similar sequences will be kept for each operon gene for subsequent NWS or BestFit scoring (K is the maximum number of genomic sequences kept for one operon gene and default to 6), a lower threshold score can be set to limits subsequent scoring to only those genomic sequences that resemble a minimum to the operon sequence (default fastp threshold score is 5).

NWS/BestFit scoring

At this step, if fastp search has been accomplished (i.e. if the "-N" option is not set), each operon sequence has at most K most similar genomic genes. Then, for one operon sequence,

its K kept most similar genomic sequence are scored by NWS or BestFit alignment with the operon sequence. A NWS (global) alignment scoring is done if the "-g" option is set, else - or if the "-l" option is set - a BestFit alignment scoring is done.

If the "-N" option was set, no fastp scan has been done, and each operon gene is scored against all genomic genes, and the K best scoring genomic genes are kept. Scoring is done as above by NWS ("-l" option) or BestFit ("-l" option or default) algorithms.

As in the case of fastp scores, each NWS/BestFit score is normalised between 0 and 100˚: 100 would be the score of the longest sequence with itself and is then the highest possible score.

If the "-S" option is set, Opscan stops there and outputs for each operon gene its K most similar genomic genes. It can output less than K genomic genes for one operon if their alignment scores are lesser than the threshold score fixed for NWS/BestFit ("-t" option) or if the length of the genomic sequence is not related with those of operon genes. The NWS/BestFit score threshold is set by the "-t" option and defaults to 40.

If the "-S" option is not set, Opscan looks for couples of "preferred" operon/genomic genes. For each operon gene, O_i, Opscan take its best NWS/BestFit scoring genomic gene, G_j. Then Opscan compares by NWS/BestFit algorithm this G_j genomic gene with all operon genes˚: if the best scoring operon gene is O_i, the couple (O_i,G_j) is considered as a "preferred" couple (i.e. a possibly homologous couple), else operon gene O_i is not paired with a genomic gene. One can note that a given operon gene cannot belong to more than one "preferred couple", and one genomic gene cannot belong to more than one "preferred couple".

As mentioned above, NWS algorithm leads to a global alignment of two sequences, while BestFit algorithm does not penalise "end gaps".
One can see an optimal alignment of two sequences as the lowest cost alignment. An alignment is the pairing of the similar amino-acids in the two sequences, together with the gaps introduced - in one sequence or in the other - to reduce mismatches between them. In both NWS or BestFit algorithms, the "scoring scheme" is important, and using different scoring schemes lead usually to different alignments. The scoring scheme resumes to the scoring of each of the amino-acids pairs that can be aligned (i.e. the similarity matrix used - see below), and to the cost of opening gaps, extension gaps and "end gaps" ("end gaps" are the gaps that are put at the end of one sequence).In the BestFit algorithm, "end gaps" cost none.

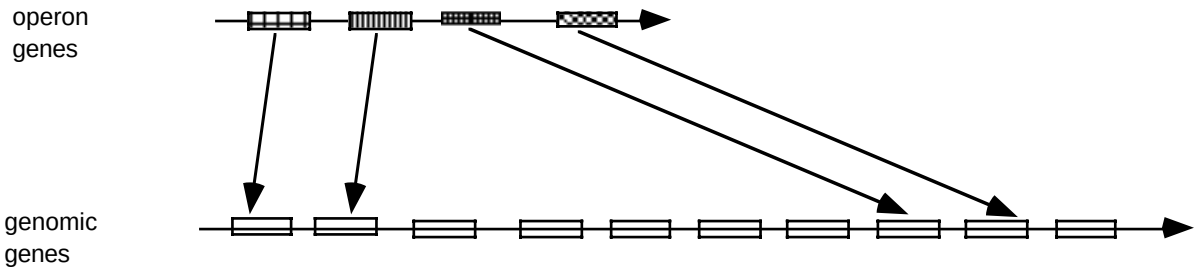
When two amino-acids match in the alignment, no cost is counted, when they differs, a certain cost is counted (mismatch cost). Also, introducing a gap in one sequence or the other cost some value ("opening gap value"). Introducing a neighbour gap in the border of another gap cost an "extension gap" value (which can be equal or less than the "opening gap" cost).
The scoring scheme for amino-acids pairing used in Opscan is set by the scoring matrix used. The chosen similarity matrix can be set using the "-M" option. By default, the scoring matrix used is the identity one˚: in this matrix, pairing of one amino-acid with itself cost none (say 0), and mismatch (pairing with a different amino-acid) cost maximum value (say 100). One can use a finer scoring scheme for amino-acid pairing in using a scoring matrix based upon known alignment of homologous protein sequences as one of the Dayhoff PAM matrices, or one of the BLAST related matrix based on similar protein alignments BLOCKS database. In these matrix, the cost for pairing different but related amino-acids does not cost always a maximum value, but instead the cost is less if the two paired amino-acids "resembles" to each other in a certain sense (see bioinformatics text books for information on alignment scoring matrices).

The scoring scheme for gaps is closely related with the scoring for amino-acids pairs, and is expressed as a relative value with respect to the maximum cost value of the similarity matrix. In Opscan, by default, gap opening cost is 1.2 times the maximum cost value for amino-acids mismatch, and gap extension cost is 0.8 the maximum cost value for amino-acids mismatch. The "—o" option sets the cost value for gap opening (example: "—o 1.7" sets the opening gap cost to 1.7 times the maximum amino-acids mismatch cost), and the "-e" sets the cost value for gap extension (example: "—e 0.5" sets the opening gap cost to 0.5 times the maximum amino-acid mismatch cost).

More details about NWS or BestFit alignment (dynamic programming algorithms) can be found in text books.

Classes of putative operon within genomic genes

At this step, some operon genes have a "preferred" genomic gene. In the usual case (if the "-Q" option is not set, see below), all operon genes are considered as genes of a single operon, and a set of "preferred" genomic gene is made of the preferred genes of all operon genes.
This set is scanned in order to find subsets of "connected" preferred genes. A "connected gene subset" is made of all the preferred genes which are "connected" with at least another gene in the subset˚: gene A is said to be "connected" to gene B if there is no more than a given number of gene lying between them in the genome. This number is the maximum separating gene number threshold which is set by the "-n" option. One can note that the order of the genes is not taken into account in the subset definition.
Such a connected subset of connected genes can be thought as a putative operon (similar to the operon query or to one of its subset). Note˚: one operon query can then lead to several different sub-operons, by splitting the operon query as explained in the figure below˚:



Here, the two groups of homolog genomic genes are too far from each other, and the operon genes are then split into two "operons".

If the "-Q" option is set, the query is considered as constituted by more than one operon. In that case, the operon are then defined in this set as subsets of query genes obeying the following conditions:

- each operon genes owes a "preferred" genomic gene
- each operon genes is separated by at most "n" genes from one of the others operon genes in the operon ("n" is given in the -n option, and has value 5 by default).

Installation

Content of distribution archive

The software should be downloaded as a compressed tar archive called opscan.tar.Z

Uncompress and untar the archive by issuing :

```
zcat opscan.tar.Z | tar xf -
```

This should produce a directory called **opscan** containing followinf files and directories :

	INSTALLATION_COMPILATION	
matrices	residus	tests
Makefile	misc	samples
README.opscan	bin	src

- ¥ bin: directory containing the binary executable opscan. (ELF binary for IRIX system 5 or 6 by default).
(see recompilation section for information on how to recompile if needed)
- ¥ Makefile˚: is the makefile for all sources, tests, etc
- ¥˚matrices: contains substitution matrix text files
- ¥ src contains all C sources
- ¥ tests contains short tests for verifying the opscan program.
- ¥ samples contains sequences and patterns samples (see the "Sample Session" section in this document)
- ¥ doc contains this documentation
- ¥ README.opscan: readme for opscan
- ¥ misc:
- ¥ INSTALLATION_COMPILATION: text file which gives explanations for recompilation and installation for opscan.

You should then add the **opscan/bin** directory into your path either manually or automatically by editing your .cshrc/.profile shell configuration file. e.g. with csh issue :

```
setenv path ($path /whatever-the-location/opscan/bin)
```

Recompilation

Normally the distribution contains ready-to-use binaries for the appropriate system, so you may skip this paragraph. However if you need to recompile the sources, go into the hspot/src directory and issue the make command :

```
cd opscan/src
make
```

__Warning__: the Makefile has been designed to operate with IRIX OS, for other platforms (e.g. SUN), you should first edit the **Makefile** file in opscan/src and follow the indications inside.

Overview

The opscan/bin directory contains the main executable file **opscan** together with several shell utilities. This part of the documentation mostly focuses on the usage of the opscan main executable.

The typical usage is :

```
opscan [options] -O operon_query [genomic_database]
```

¥ **operon_query**: name of a file containing the operon sequences to be searched for.

¥ **genomic_database**: name of a file containing the genomic sequences.

The file formats are described below.

¥ [**options**] : are program options of the form -key [value]
key: one letter key
[value] optional integer (nn) or float (ff) value)

to get a summary of all available options (see annex 1), issue:

```
opscan -h
```

Sequence(s) Input File Format

The genomic and operon sequence databases can be in a Fasta like format (called "mnemo format") or in standard Fasta format. The difference in the two format resides principally in the numbering of the genes, and in the first line description for each gene. In the standard fasta format, the gene number — according to its placing on the chromosome - is not given (only a short name followed by a short description are given), while in the mnemo format, the number of the gene is part of the format of the first line.

"Mnemo" format

It is defined as follows˚:

```
><mnemo> <gene_number> <from> <to> <strand>
sequence (on_several_lines)
```

where˚:

> : the leading ">" is mandatory

<mnemo>: is the gene name or id (alphanumeric characters except space) . It can be a simple name or a complex name as :
<gene_ID>@<org_ID>@<contig_ID>
where˚:
<gene_ID> is the gene name
<org_ID> is the organism name
<contig_ID> is the contig identifier

<gene_number>˚:
this field has a different meaning in operon database and in genome database˚:

- In operon database, if <gene_number> is 1, the gene is to be searched as the part of the putative operon, it can also be a relative number (as in the genome database, see below), if <gene_number> is 0: the gene is ignored in the search.
- In genome database, <gene_number> is the relative gene number on the chromosome.

Note: If the genome.db does not have <gene_number> (this is the case when genome.db is a standard fasta file), the genes are supposed to be in the order of their appearing on the chromosome. The same remark holds for the operon database.
If opscan do not retrieve '@' characters in mnemo fields, it tries to read <gene_number>, <from>, <to> and <strand> fields, if even this fails, it continues assuming simple fasta format (then the numbering of the genes is done according to their rank in the database file).

<from>°: this field gives the position on the chromosome where the gene begins (integer). This field is relevant in the genome database, and is ignored in the operon database.

<to>°: this field gives the position on the chromosome where the gene ends (integer). This field is relevant in the genome database, and is ignored in the operon database.

<strand>°: this field gives the strand where gene resides: 'D' for direct strand, 'C' for complementary strand

Note: **<mnemo>, <from>, <to>** and **<strand>** are only user informations, they are not used by opscan, while the **<gene_number>** information is used by opscan, as it determines the number of possible operons (in the operon database and in the genome database).

Here is an example of entries in mnemo format:

```
>pyrK@pig@contig101 1 500 1400 C
MNRPSWSTAFNIGGGFPIQWYGIIVSIGIIFAILMFVFKLIYCYKLQDNS
FYFFIFIAVLTMVLGARLWSFVIGDSNFANNNFFDFRNGGLAIQGGILLT
SIVGVIYFNFFLNSKTNKTKTIAELLNNKNEIKAVYVERNISVLVMLDLI
APCVLIGQAIGRWGNFFNQEVYGFALAGTMNDPQALANTQWGFLKILMPK
VWDGMWIDGQFRIPLFLIESFFNTIFFVLIYFVMDFIRGVKSGTIGFSYF
LATGIIRLILENFRDQTFYFQTSITTSILFIVVGILGIFYCQFIHVKLRN
>pyrL@pig@contig102 1 1600 2500 D
MNRPSWSTAFNIGGGFPIQWYGIIVSIGIIFAILMFVFKLIYCYKL
FYFFIFIAVLTMVLGARLWSFVIGDSNFANNNFFDFRNGGLAIQGGILLT
SIVGVIYFNFFLNSKTNKTKTIAELLNNKNEIKAVYVER
APCVLIGQAIGRWGNFFNQEVYGFALAGTMNDPQALANTQWGFLKILMPK
VWDGMWIDGQFRIPLFLIESFFNTIFFVLIYFVMDFIRGTSYF
LATGIIRLILENFRDQTFYFQTSITTSILIHVKLRN
```

Standard fasta format°:

It is defined as follows°:

```
>name_of_sequence [optional comments]
sequence (on_several_lines)
```

> : is mandatory

name : should be a one word name (less than 64 chars). This is the name that will be printed out in the result file.

comments: are optional, any number of words on that line are considered to as a comment string (including the spaces)

sequence: the sequence itself, either in lower or upper case characters. The sequence may be split over any number of lines.
Each line may have any number of nucleotides (providing that a line does not exceed 8192 characters).
The sequence should only contains letters in the range [a-z] and [A-Z], (**No numbers nor spaces !**)

Here is an example of a an entry in standard fasta format:

```
>MYGE_000910 | prolipoprotein diacylglyceryl transferase
MNRPSWSTAFNIGGGFPIQWYGIIVSIGIIFAILMFVFKLIYCYKLQDNS
FYFFIFIAVLTMVLGARLWSFVIGDSNFANNNFFDFRNGGLAIQGGILLT
SIVGVIYFNFFLNSKTNKTKTIAELLNNKNEIKAVYVERNISVLVMLDLI
APCVLIGQAIGRWGNFFNQEVYGFALAGTMNDPQALANTQWGFLKILMPK
VWDGMWIDGQFRIPLFLIESFFNTIFFVLIYFVMDFIRGVKSGTIGFSYF
LATGIIRLILENFRDQTFYFQTSITTSILFIVVGILGIFYCQFIHVKLRN
>MYGE_000911 | what is this protein ?
MNRPSWSTAFNIGGGFPIQWYGIIVSIGIIFAILMFVQDNS
FYFFIFIAVLTMVLGARLWSFVIGDSNFANNNFFDFRNGGLAIQGGILLT
SIVGVIYFNFFLLNNKNEIKAVYVLVMLDLI
APCVLIGQAIGRWGNFFNQEVYGFALAGTMNDPQALAGFLKILMPK
VWDGMWIDGQFRIFYCQFIHVKLRN
```

Options

Options are divided into general options, fastp options, bestfit/NWS options and class options.

General options°:

Getting help (-h)

type **opscan -h** to get a summary of all available options. (See Annex 1)

No Fastp option (-N)

When -N option is set, no fastp, fastp is not run as a prefilter i.e. only bestfit will be run between all operon sequences and all genome sequences. This option is not set by default and fastp is run as a prefilter for similarity.
Note: This option slows the run.

Fasta format option (-E and -F)

When -F option is set, genomic sequences are to be given in standard fasta format (not in "mnemo format" (see input file format paragraph)) and the genomic genes are supposed to be contiguous and given according their order on the chromosome.
When -E option is set, operonic sequences are to be given in standard fasta format (not in "mnemo format" (see input file format paragraph)) and the operonic genes are supposed to be contiguous and given according their order on the chromosome.

Ratio length threshold option (-r <real>)

When this option is set, the real following —r is taken as the ratio length threshold between operon and genomic gene to be candidate to be "homolog", i.e. a genomic gene is retained for

comparison with an operon gene, if the ratio of the longest sequence length divided by the shortest one is less than this "ratio length threshold". In other words, if one sequence is greater than r times the other, or lesser than 1/r times the other, they won't be compared, and they never can be "homolog"
Example: -r 2 means that genomic gene g and operon gene o won't be compared if length of g is greater than twice or lesser than 0.5 times length of o (no threshold when r is less than 1.0). By default, this ratio is 1.3

Maximum separating genes for 2 genes to be in the same class (-n <integer>)

When this option is set, the integer following —n is taken as the maximum number of genes separating two genes to be considered in the same class (i.e. possibly in the same operon). By default, this number is 5.

Not bijective option (-b)

When this option is set, two or more operon genes can be "homolog" with the same genomic gene. This option is not set by default, and their must exist a one to one similarity relation between operon gene and genomic gene in order to consider them as "homolog", i.e. the genomic gene must be the most similar to the operon gene AND the operon gene must be the most similar to the genomic gene

Only similarity option (-S)

When this option is set, no classes are computed, only similarity between operon genes and genome genes are output. This option is not set by default.

Singleton option (-s)

Do not output [s]ingletons. The default is Off. Single couples operon/genomic genes are output by default. A singleton is an "homolog" pair which does not belong to a class of several operon/gene pairs and is not considered as a class.

Circular genome option (-c)

When this option is set, the genomic gene set is considered as circular, i.e. the last gene is considered as a neighbor of the first gene. This option is not set by default.

Circular operon option (-U)

When this option is set, the operon gene set is considered as circular, i.e. the last gene is considered as a neighbor of the first gene. This option is not set by default.

Operon class option (-Q)

When this option is set, the operon genes in the operon database are not considered as belonging to the same operon, and "sub-operons" are computed within the operon database with the same criteria as those used for the genome database (Note˚: the —Q option does not ouput these intermediate operon classes, see the —s option).

Operon class output option (-C)

When this option is set, the —Q option is also set, so the operon genes in the operon database are not considered as belonging to the same operon, and "sub-operons" are computed within the operon database with the same criteria as those used for the genome database (as in the —Q

option, see below). In that case, the output contains also these intermediate operon classes (the —Q option does not ouput the intermediate operon classes).

Similar kept genes option (-K <integer>)

When this option is set, the integer following K is the number of best similar genomic genes to keep in the first run (fastp or bestfit) for each operon entry. By default, 6 best similar genome genes are kept for each operon gene. When this number is large, opscan takes longer time to achieve, but sensibility of the similarity search is increased, as the best similar genomic gene of an operon gene can occasionally have a worse fastp score than a less similar one. When running in only bestfit/NWS mode (-N option), this number is irrelevant as best similar genomic genes are choosen on the bestfit/NWS score.

Replace unknown symbol option (-R <character>)

When this option is set, opscan replaces unknown characters in the sequences (as '*', '@' or 'Z') by the character following —R option. By default, the replacing character is 'X'.

Delete unknown symbol option (-D)

When this option is set, opscan deletes unknown characters in the sequences (as '*', '@' or 'Z').

Statistic scores option (-Z)

When this option is set, statistics are computed on shuffled genomic sequences. If fastp is to be done (i.e. option —N is not set), statistics are computed on fastp scores of shuffled genomic sequences. If only bestfit is done, (i.e. option —N is set) statistics are computed on bestfit scores of shuffled genomic sequences. Note˚: this option doubles running time.

Store besfit shuffled scores option (-T)

When the statistic option is set, this option permits to store the shuffled bestfit score for an operon gene, this option takes huge memory, but speeds up the process.
Note: this option is only valuable when the statistic option is on (option -Z), otherwise it has no effects.

Combinatoric experimental search (-a)

This option permits a combinatoric search of operons based upon a systematic search of all potential operons (even if genomic genes are not the bijective "homologs" of the operon genes, but are only in the list of kept genes of the operon genes. Very huge output.

Verbose option (-v) or very verbose option (-V)

When scanning a long database or a long sequence and when the output is redirected into a file one may wish to get an idea of the completion status of the program. The -v option will print regularly a status line to indicate where the process is. This is printed on <**stderr**> and therefore will not appear in the result file. The —V (very verbose option) give an ouput of best similar genome genes for each operon gene after both fastp and bestfit runs.

Fastp options

For the fastp run, there are "scoring options" for fastp (k,L,x,I) and "report options" (-m,-n).

Kuple size option (-k <integer>)

When this option is set, the integer following the —k option is the size of the Kuple used by fastp algorithm (see fastp explanation parapgraph above). The default for k-uple size is 2.

Lower score threshold (-L <integer: 0-100>):

This option gives the similarity lower score threshold in fastp for subsequent exclusion in bestfit search. The default is a threshold 5. Note: the score, in fastp or bestfit/NWS are normalized in the 0-100 range (0 means no similarity and 100 means complete similarity)

k-uple score option (-x):

When this option is set, Fastp score is computed as the number of kuples which match for the best shift between the two sequences. By default, this option is off, and the Fastp score is computed based on the alignment score of the best shift alignment (see fastp explanation paragraph).

Diagonal integration option (-i <integer>):

This option is only useful when the —x option is set. The integer gives the diagonal integration in fastp computations (see fastp explanation paragraph). By default: -i 2 when -x option, else no integration is done.

Report fastp alignments option (-p <filename>):

The name following the —p option will be the filename of the file where all fastp alignments will be output (this option is not useful and is not yet implemented).

Bestfit/NWS options°:

Local alignment option (-l)

When this option is set, the alignment is a "bestfit" alignment (no gaps are counted in the ends of sequences). By default, this option is on.

Global alignment option (-g)

When this option is set, the alignment is a "NWS" alignment (gaps are also penalized in the ends of sequences). By default, this option is off.

Bestfit score threshold option (-t <integer>)

```
This option gives the bestfit score threshold (in the 0-100 range) for a
couple operon gene / genomic gene to be considered as "homolog". This
threshold is 40 by default.
```

Gap and Weights options (-M <matrix_file_name>, -o <real>, -e <real>)

-M option is followed by a filename, which indicates the similarity scoring matrix where to take scoring pair values (like a PAM or BLOSUM matrix) for the alignment. Note: this matrix will be used also for scoring fastp alignments. By default, the identity matrix will be used for similarity, i.e. all matches cost nothing, and all mismatches cost worst value (for example 1.0). Following the -o option is a real number, which serves to compute the "opening gap penalty" applied in the alignment procedure. This real number is a factor, which multiply the worst (100) similarity score to give the gap penalty. For example, if distance score pairs in the

distance matrix (which is computed on the base on the similarity matrix, option -M) are in the 0 (best)-1(worst) range, and option —o 1.2 is given, a gap penalty will cost 1.2 (to be compared with the worst matching pair which have a value of 1.0). By default, this value is 1.2.
The -e option is followed by a real number, which serves to compute the "extension gap penalty" applied in the alignment procedure. This real number is a factor, which multiply the worst (100) similarity score to give the gap extension. For example, if distance score pairs in the distance matrix (which is computed on the base on the similarity matrix, option -M) are in the 0 (best)-1(worst) range, and option —e 0.8 is given, a gap extension will cost 1.2 (this value is to be compared with the worst matching pair which have a value of 1.0). By default, this real value is 0.8.

Bestfit alignment report option (-P)

When this option is set, opscan outputs the bestfit alignements for the "homolog" operon/genome gene pairs . By default, alignments are not reported.

Output File and Output format options

The output is normally send to <**stdout**> (i.e. to the screen), one has to redirect it into a file if needed. e.g. with C-shell :

```
opscan -O -O my_op_query my_genomic_db > ! my_result
```

The output file is an ascii file.
Its general structure is :
KEY<space><space>VALUE

where KEY is a two-letter-code keyword
and VALUE is anything relevant to this key.

Header record (HD)

The header record starts the output file. Its associated keyword is **HD**. The HEADER record basically resumes all pertinent parameters used in the opscan run : general settings, fastp settings, bestfit settings and class settings. Then there are four header parts:

- the general header which includes the shell command, the name of the genomic database, the number of sequences in operon database and genomic database),
- the Fastp header: giving the parameters related to the fastp part of opscan,
- the Bestfit header: giving the parameters related to the NWS or Betsfit part of opscan,
- the Class header, giving the parameters related to the class part of opscan.

```
HD  /----------------------------------------
HD  / Opscan Version 0.9 Fev. 2000
HD  /----------------------------------------
HD  / running cmd  : opscan -v -r 1.5 -L 1.0 -O operon_sample.db
genome_sample.db
HD  / database              : genome_sample.db
HD  / number of operon seq  : 6
HD  / number of genome seq  : 13
HD  /----------------------------------------
HD  /----------------------------------------
HD  / Fastp part:
```

```
HD  /----------------------------------------
HD  / alphabet               : ABCDEFGHIJKLMNOPQRSTUVWXYZ
HD  / kuple                  : 2
HD  / fastp diag integ.      : 2
HD  / fastp lower threshold  : 1
HD  / unknown symbol : replace by X
HD
     /…/
HD  /----------------------------------------
HD  /   Bestfit part
HD  /----------------------------------------
HD  / # of operon genes: 6
HD  / Max # of genome genes by operon: 6
HD  / algorithm      : bestfit (local)
HD  / scoring matrix : <internal> Identity
HD  / gap scheme     : 1.2 + (L-1) * 0.8
HD
/…/
HD  /----------------------------------------
HD  /   Class part
HD  /----------------------------------------
HD  / Bestfit score threshold for "homologs": 40
HD  / length ratio threshold : 1.500000
HD  / Maximum number of genes separating two genes
HD  /   to be in the same class:5
HD  / One to one relation: YES
HD  / Singletons are output
HD  / Operon set is linear
HD  / Chromosome is linear
HD  / Compute also classes within operon genes:NO
HD  / Do not  output of intra-operonic classes (see README.opscan)
```

COMPLETION records

They are associated the "CO" keyword. CO records indicate the completion of the process (i.e. number of sequence read until now, etc). These records are printed on **stderr**.
Example of output when —V option is set.

```
CO  List of similar genes after Fastp:

CO  List of similar genes with operon MYGE_000900:
CO      gene URUR_000740 (score  17.97  dlen ratio  1.01)
CO      gene URUR_006390 (score   6.59  dlen ratio  1.07)
CO      gene URUR_003680 (score   6.53  dlen ratio  1.19)
CO      gene URUR_002220 (score   6.53  dlen ratio  1.26)
CO      gene URUR_005560 (score   6.47  dlen ratio  1.10)
CO      gene URUR_005050 (score   6.47  dlen ratio  1.24)

CO  List of similar genes with operon MYGE_000910:
CO      gene URUR_000730 (score  10.21  dlen ratio  1.14)
CO      gene URUR_005880 (score   6.25  dlen ratio  1.13)
CO      gene URUR_006020 (score   6.06  dlen ratio  1.15)
CO      gene URUR_006070 (score   6.05  dlen ratio  1.29)
CO      gene URUR_005550 (score   5.73  dlen ratio  1.09)
CO      gene URUR_003240 (score   5.73  dlen ratio  1.21)
```

Here, the partial listing gives intermediate results of kept genomic genes for each operon genes°: score is the fastp score, and "dlen ratio" is the sequence length ratio (i.e. the length of the longest sequence divided by the length of the shortest (see —L option).

HELP records

They are associated the HE keyword. Help records are displayed when the —h option is set, or if opscan can not run due to incorrect settings.

WARNING records

They are associated the WA keyword, and ERROR records, the ER keywords. Warnings are displayed for example when two contradictory options are set, and opscan choose one over the other. Errors records are displayed when opscan can not run with current settings, can not open a file or can not choose an option set.
Example:

```
WA  *Warning* [0] Output of intra-operonic classes (option -C) needs -Q
option (operon classes)
WA  *Warning* [0] Setting the -Q option...
```

REMARK records

They are associated the "RE" keyword. They are displayed when non fatal errors are encountered as for example, when a gene has a too long name, or when opscan read an input file in fasta format while the input fasta format options (-E or —F) are not set.
Example°1:

```
RE  OperonParseInfo:: error parsing D ATG TAG 111790 112725 Valid MG085 936
1 111790 112725 | hpr(ser) kinase, putative . (translation)
should be int, int, int, char
RE   Continue reading assuming standard fasta format...
```

CLASS records (putative operons)

Opscan outputs classes, which can be considered as the putative operons it has found. The class records are associated several keywords ("CN", "CL", "CD", and "SG"). The "CN" record indicates the number of "homolog" classes (putative operons). For each operon gene-genomic gene "homolog" pair, the "CL" records indicate the class it belongs to and the names of the two genes. The "CD" records contain the description of both genes in the pair depicted in the preceding "CL" record.
Syntax of CL records:

```
CL <clnum> <opname> <opnum> <opclass> <gname> <gnum> <score> <pr> <lratio>
```

where°:

- <clnum> is the number of the class and is in the 0-(n-1) range if there are n classes,
- < opname > is the name or "mnemo" of operon gene of the "homolog" pair,
- <opnum> is the number of the operon gene (i.e. it is its relative number in the operon file, or its number if it was specified (see input formats)),
- <opclass> is the number of the class of the operon gene in the operon file (if —E option is on), else this field is "-1" which means "no class",
- <gname> is the name or "mnemo" of genomic gene of the "homolog" pair,

- <gnum> is the number of the operon gene (i.e. it is its relative number in the genome file, or its number if it was specified (see input formats)),
- <score> is the score in the Bestfit/NWS alignment for this pair,
- <pr> is the probability for the bestfit alignment of this pair,
- <lratio> is the ratio of the longest sequence length of the pair divided by the shortest one.

See example below where the first class (class 0) contains MYGE_000910/ URUR_000730 and MYGE_000900/ URUR_000740 "homolog" pairs˚:

```
CN  Number of genomic classes: 4
CP  Number of operonic classes: 1
CL     0 MYGE_000910    2    0 URUR_000730   73  48.4 -1.000  1.14
CD         MYGE_000910: prolipoprotein diacylglyceryl transferase (lgt)
CD         URUR_000730: prolipoprotein diacylglyceryl transferase .
CL     0 MYGE_000900    1    0 URUR_000740   74  54.7 -1.000  1.01
CD         MYGE_000900: hpr(ser) kinase, putative . (translation)
CD         URUR_000740: hpr serine/threonine protein kinase . (translation)
CL     1 MYGE_000940    5    1 URUR_005630  529  77.5 -1.000  1.00
CD         MYGE_000940: elongation factor G (fus) . (translation)
CD         URUR_005630: translation elongation factor G . (translation)
CL     1 MYGE_000930    4    1 URUR_005640  530  76.0 -1.000  1.00
CD         MYGE_000930: ribosomal protein S7 (rpS7) . (translation)
CD         URUR_005640: ribosomal protein S7 . (translation)
CL     1 MYGE_000920    3    1 URUR_005650  531  82.2 -1.000  1.01
CD         MYGE_000920: ribosomal protein S12 (rpS12) . (translation)
CD         URUR_005650: ribosomal protein S12 . (translation)
CL     2 MYGE_000990   10    2 URUR_005910  556  42.4 -1.000  1.09
CD         MYGE_000990: replicative DNA helicase (dnaB) . (translation)
CD         URUR_005910: replicative DNA helicase . (translation)
CL     2 MYGE_000980    9    2 URUR_005920  557  52.9 -1.000  1.03
CD         MYGE_000980: ribosomal protein L9 (rpL9) . (translation)
CD         URUR_005920: ribosomal protein L9 . (translation)
CL     2 MYGE_000960    7    2 URUR_005940  559  42.0 -1.000  1.04
CD         MYGE_000960: single-stranded DNA-binding protein (ssb) .
CD         URUR_005940: single-strand binding protein . (translation)
CL     2 MYGE_000950    6    2 URUR_005950  560  48.8 -1.000  1.30
CD         MYGE_000950: ribosomal protein S6 (rpS6) . (translation)
CD         URUR_005950: ribosomal protein S6 . (translation)
```

The "SN", "SG" and "SD" keywords depict the singleton pairs, i.e. the pairs of "homolog" operon/genomic genes which do not belong to a class (see above for a definition of an "homolog" pair, and for the definition of a class). The syntax for SN, SG and SD records is respectively the same as for the CN, CL and CD records, but the class number of every gene is —3 (meaning no class).

Example˚:

```
SN  Single operon/genomic gene couples: 1
SG  MYGE_000970    8   -3 URUR_001700  164  -2  39.4 -1.000  1.03
SD         MYGE_000970: ribosomal protein S18 (rpS18) . (translation)
SD         URUR_001700: unique hypothetical . (translation)
```

"intra-operonic classes" records

They are associated the "ON" and "OC" keyword. The "ON" record gives the number of intra-operonic classes. These records are output when the —C option is set, i.e. when the user asks for finding class in the operon sequence file˚: the operon genes are not considered as belonging to only one operon (default), but to several possible operons (determined by the class parameters) These operons can be viewed as potential "sub-operons" in the operon

sequence file, as, if all genes of a "sub-operon" have their homologs in the same region (see discussion in the option —n paragraph) they can lead to an effective operon. The "OC" records show how operonic genes are ordered in these "sub-operons". If the class number is —1, then the corresponding operonic gene is not in a possible operon, i.e. its distance to other operonic genes is greater than the limit given.
The "OC" records syntax is˚:

R1 <opename> <openum> **-> class** <class_number> ([does|dose not] have homolog)

where˚:

- <opename> is the name of the operon gene
- <openum> is the relative number of the operon gene in the operon sequence file
- <class_number> is the number of the "sub-operon" which operon gene belongs to (this number is in the 0-(N-1) range if N is the number of sub-operons, or —1 if the operon gene does not belong to a "sub-operon".
- Followed by "does have homolog" or "does not have homolog", as the operon gene which does not have homolog can not belong to an operon.

Note: "sub-operon" can contains only one operon gene.

Example of output˚:

```
ON  Intra-operon classes: 3
OC  MYGE_000900    1 class:    0 (does have homolog)
OC  MYGE_000910    2 class:    0 (does have homolog)
OC  MYGE_000920    3 class:    0 (does have homolog)
OC  MYGE_000920    4 class:   -1 (does not have homolog)
OC  MYGE_000921    5 class:   -1 (does not have homolog)
OC  MYGE_000922    6 class:   -1 (does not have homolog)
OC  MYGE_000923    7 class:   -1 (does not have homolog)
OC  MYGE_000924    8 class:   -1 (does not have homolog)
OC  MYGE_000925    9 class:   -1 (does not have homolog)
OC  MYGE_000926   10 class:    1 (does have homolog)
OC  MYGE_000927   11 class:   -1 (does not have homolog)
OC  MYGE_000928   12 class:   -1 (does not have homolog)
OC  MYGE_000929   13 class:   -1 (does not have homolog)
OC  MYGE_000930   14 class:   -1 (does not have homolog)
OC  MYGE_000935   15 class:   -1 (does not have homolog)
OC  MYGE_000940   16 class:    2 (does have homolog)
OC  MYGE_000950   17 class:    2 (does have homolog)
OC  MYGE_000960   18 class:    2 (does have homolog)
OC  MYGE_000970   19 class:    2 (does have homolog)
OC  MYGE_000980   20 class:    2 (does have homolog)
OC  MYGE_000990   21 class:    2 (does have homolog)
```

Here, there are 3 potential "sub-operons" as MYGE_000920 to MYGE_000925 and MYGE_000927 to MYGE_000935 have not homologs.

SUMMARY records

They are associated the "SU" keyword and are displayed on the standard error. They finish the opscan listing by giving information about the run, as the total numbers of operon and genomic sequences read, the number of correct length sequences eliminated after fastp, the fraction of correct sequences kept, the number of comparison performed in fatsp and besfit, the user cpu time and the memory used by the opscan run.
Example:

```
SU  /----------------------------------------
SU   Total # of operon sequences read: 10
SU   Total # of genome sequences read: 611
SU   # of correct length sequences eliminated after fastp: 556
SU   fraction of correct sequences kept: 100.0
SU   Comparisons space: 6110
SU   Total # of fastp comparisons performed: 1219
SU   Total # of bestfit alignments performed in first pass: 59
SU   Total # of bestfit alignments performed in second pass: 1
SU   User Cpu time: 1.41 s
SU   Sys  Cpu time: 0.36 s
SU   Memory used: 16224 Kb
SU   End of program
```

Time and Space considerations

Memory space: opscan is only limited by the memory amount of the computer because memory allocation is dynamically processed.
Running time˚: the fastp part permits opscan to scan quickly the genome database, as a pre-filter for possible "homologs", and then, the bestfit or NWS algorithm is run only between operon genes and their n possible homologs (n is the number of kept genome genes for each operon gene, see —K option).

Caveats and Bugs

don’t hesitate to send questions, bug reports etc... at :
Joel.Pothier@snv.jussieu.fr

Annex 1 - Options summary

```
HE  ----------------------------------------
HE   Opscan with Operon Query  0.1 (mars 2000)
HE  ----------------------------------------
HE  synopsis :
HE    Fastp then SmithWaterman (bestfit) operon search in genome
HE    database starting with an operon file
HE
HE  usage:
HE    opscan [options] -O operon_query [db_file]
HE
HE  options:
HE  (*): not yet implemented
HE  ----- General options   -----------------
HE  -N           : no fastp, fastp is not run as a prefilter
HE                 i.e. only bestfit will be run between all
HE                 operon sequences and all genome sequences
HE              (default: fastp is run as a prefilter)
HE              Note: this option slows the run
HE
HE  -F           : genomic sequences are in standard fasta format
HE                  (not in "mnemo format" (see README.opscan))
HE                  the genomic genes are supposed to be contiguous
HE                  and in the order they are placed on the chromosome
HE
HE  -E           : operon sequences are in standard fasta format
HE                  (not in "mnemo format" (see README.opscan))
HE                  the operon genes are supposed to be contiguous
HE                  and in the order they are placed on the chromosome
HE
HE  -n           : maximum number of genes separating two genes
HE                 to be considered in the same class
HE                 default: 5
HE
HE  -r <float>   : [r]atio length threshold (float)
HE                 ratio length threshold between operon and genomic
HE                 genes to be candidate to be "homolog",
HE                 i.e. genomic gen is retained for comparison with
HE                 operon ope, if ratio of the longest sequence
HE                 length divided by the shortest one is less than
HE                 "ratio length threshold"
HE                 In other words, if one sequence is greater than r
HE                 times the other, or lesser than 1/r times the other,
HE                 they won't be compared, and they never can be "homolog"
HE                 Example: -r 2 means that genomic gene g and operon
HE                 gene o won't be compared if length of g is greater
HE                 than twice or lesser than 0.5 times length of o
HE                 (no threshold when r is less than 1.0)
HE                 default: -r 1.3
HE
HE  -c           : chromosome is circular [default: Off]
HE
HE  -b           : not [b]ijective:  two or more operon genes can
HE                 be "homolog" with the same genomic gene
HE                  [default: Off, i.e. one to one      relation
HE                  between operon gene and genomic gene:
HE                  the genomic gene must be the most similar to the
HE                  operon gene AND the operon gene must be the most
HE                  similar to the genomic gene]
HE
```

```
HE  -S           : no classes are computed, only similarity
HE                 between operon genes and genome genes are
HE                 output
HE
HE  -s           : Do not filter [s]ingletons [Default: Off]
HE                 (single couples operon/genomic genes are
HE                 filtered by default, i.e. are not considered
HE                 as classes)
HE
HE  -Q           : classes are computed also for operon genes
HE
HE  -C           : output also R1 class (i.e. sub-operons in the
HE                 operon genes)
HE
HE  -U           : operon gene set is circular [default: Off]
HE
HE  -K integer   : number of similar sequences to keep in first run
HE                 (fastp or bestfit) for one entry
HE                 (default:6)
HE
HE  -R char      : [R]eplace unknown symbol by 'char'
HE                 default: -R 'X'
HE                 Note: alphabet is [A-Z]
HE                 see also: -D
HE
HE  -D           : [D]elete unknown symbol
HE                 default: off
HE                 see also: -R
HE
HE  -Z           : statistics are computed on shuffled genomic
HE                 sequences
HE                 If fastp is to be done, statistics are computed
HE                 on fastp scores of shuffled genomic sequences
HE                 If only bestfit (-N option) statistics are
HE                 computed on bestfit scores of shuffled genomic
HE                 sequences (this option doubles computing time)
HE
HE  -T           : s[T]ores bestfit shuffled scores
HE                 Speeds up, but take huge memory
HE
HE  -a           : [experimental] combinatoric search of operons
HE                 based upon a systematic search of all potential
HE                 operons (even if genomic genes are not the
HE                 bijective "homologs" of the operon genes, but
HE                 only in the list of kept genes of the operon
HE                 genes)
HE
HE  -v           : [v]erbose
HE                 completion verbosity
HE                 default = Off
HE
HE  -V           : [Very] verbose
HE                 completion + huge verbosity
HE                 (alignments are output)
HE                 default = Off
HE
HE  -h           : [H]elp - print <this> help
HE
HE  file formats
HE
HE  operon_query:   "mnemo format" (special fasta format special)
HE
HE  db_file:        "mnemo format" (special fasta format special)
```

```
HE
HE                    (see README.opscan)
HE  --------------------------------
HE  ----------------------------------------
HE   options for the Fastp part of Opscan
HE  ----------------------------------------
HE
HE  ----- Scoring options  -----------------
HE
HE  -k integer   : [K]uple size (integer)
HE                 default: -k 2
HE
HE  -L integer   : [L]ower threshold (integer:0-100)
HE                 similarity lower threshold in fastp for
HE               subsequent exclusion in bestfit search
HE                 [default: -L 5]
HE               Range: 0 (no similarity)-100 (complete
HE               similarity)
HE
HE  -x           : Fastp score is computed as the number of
HE                 kuples which match for the best shift between
HE                 the two sequences (default: off, the Fastp score
HE                 is computed based on the alignment score of the
HE                 best shift alignment)
HE
HE  -i integer   : Only in case of -x option, the integer gives
HE                 the diagonal [I]ntegration in fastp computations
HE                 default:  -i 2       (when -x option, else no
HE                                  integration is done)
HE
HE  ----- Report options          ------------------
HE
HE  -p filename  : [P]rint out alignments into filename(*)
HE                 default = Off
HE
HE  ----------------------------------------
HE   options for bestfit
HE  ----------------------------------------
HE
HE  -l           : [L]ocal alignment (BestFit)
HE                 default = On
HE  -g           : [G]lobal alignment
HE                 default = Off
HE
HE  -t <integer> : bestfit score threshold (0-100) for a couple
HE                 operon gene / genomic gene to be considered as
HE                 homolog      [default 40]
HE
HE  ----- Gap & Weights options  ------------------
HE  -M filename  : Scoring [M]atrix
HE                 default = <internal> Identity
HE  -o ff        : Gap [O]pening weigth
HE                 default = 1.2
HE  -e ff        : Gap [E]xtension weigth
HE                 default = 0.8
HE
HE  ----- Report options          ------------------
HE  -P           : [P]rint out alignment
HE                 default = Off
HE  ----------------------------------------
HE  file formats
HE
HE  matrix_file:    blast format
```

```
HE  --------------------------------
HE
abikini 115%
abikini 115%
abikini 115% ../bin/opscan -h
HE  ----------------------------------------
HE   Opscan with Operon Query  0.1 (mars 2000)
HE  ----------------------------------------
HE  synopsis :
HE    Fastp then SmithWaterman (bestfit) operon search in genome
HE    database starting with an operon file
HE
HE  usage:
HE    opscan [options] -O operon_query [db_file]
HE
HE  options:
HE  (*): not yet implemented
HE  ----- General options   ------------------
HE  -N           : no fastp, fastp is not run as a prefilter
HE                 i.e. only bestfit will be run between all
HE                 operon sequences and all genome sequences
HE              (default: fastp is run as a prefilter)
HE              Note: this option slows the run
HE
HE  -F           : genomic sequences are in standard fasta format
HE                  (not in "mnemo format" (see README.opscan))
HE                  the genomic genes are supposed to be contiguous
HE                  and in the order they are placed on the chromosome
HE
HE  -E           : operon sequences are in standard fasta format
HE                  (not in "mnemo format" (see README.opscan))
HE                  the operon genes are supposed to be contiguous
HE                  and in the order they are placed on the chromosome
HE
HE  -n           : maximum number of genes separating two genes
HE                 to be considered in the same class
HE                 default: 5
HE
HE  -r <float>   : [r]atio length threshold (float)
HE                 ratio length threshold between operon and genomic
HE                 genes to be candidate to be "homolog",
HE                 i.e. genomic gen is retained for comparison with
HE                 operon ope, if ratio of the longest sequence
HE                 length divided by the shortest one is less than
HE                 "ratio length threshold"
HE                 In other words, if one sequence is greater than r
HE                 times the other, or lesser than 1/r times the other,
HE                 they won't be compared, and they never can be "homolog"
HE                 Example: -r 2 means that genomic gene g and operon
HE                 gene o won't be compared if length of g is greater
HE                 than twice or lesser than 0.5 times length of o
HE                 (no threshold when r is less than 1.0)
HE                 default: -r 1.3
HE
HE  -c           : chromosome is circular [default: Off]
HE
HE  -b           : not [b]ijective:  two or more operon genes can
HE                 be "homolog" with the same genomic gene
HE                  [default: Off, i.e. one to one      relation
HE                  between operon gene and genomic gene:
HE                  the genomic gene must be the most similar to the
HE                  operon gene AND the operon gene must be the most
HE                  similar to the genomic gene]
```

```
HE
HE  -S           : no classes are computed, only similarity
HE                 between operon genes and genome genes are
HE                 output
HE
HE  -s           : Do not filter [s]ingletons [Default: Off]
HE                 (single couples operon/genomic genes are
HE                 filtered by default, i.e. are not considered
HE                 as classes)
HE
HE  -Q           : classes are computed also for operon genes
HE
HE  -C           : output also R1 class (i.e. sub-operons in the
HE                 operon genes)
HE
HE  -U           : operon gene set is circular [default: Off]
HE
HE  -K integer   : number of similar sequences to keep in first run
HE                 (fastp or bestfit) for one entry
HE                 (default:6)
HE
HE  -R char      : [R]eplace unknown symbol by 'char'
HE                 default: -R 'X'
HE                 Note: alphabet is [A-Z]
HE                 see also: -D
HE
HE  -D           : [D]elete unknown symbol
HE                 default: off
HE                 see also: -R
HE
HE  -Z           : statistics are computed on shuffled genomic
HE                 sequences
HE                 If fastp is to be done, statistics are computed
HE                 on fastp scores of shuffled genomic sequences
HE                 If only bestfit (-N option) statistics are
HE                 computed on bestfit scores of shuffled genomic
HE                 sequences (this option doubles computing time)
HE
HE  -T           : s[T]ores bestfit shuffled scores
HE                 Speeds up, but take huge memory
HE
HE  -a           : [experimental] combinatoric search of operons
HE                 based upon a systematic search of all potential
HE                 operons (even if genomic genes are not the
HE                 bijective "homologs" of the operon genes, but
HE                 only in the list of kept genes of the operon
HE                 genes)
HE
HE  -v           : [v]erbose
HE                 completion verbosity
HE                 default = Off
HE
HE  -V           : [Very] verbose
HE                 completion + huge verbosity
HE                 (alignments are output)
HE                 default = Off
HE
HE  -h           : [H]elp - print <this> help
HE
HE  file formats
HE
HE  operon_query:   "mnemo format" (special fasta format special)
HE
```

```
HE  db_file:        "mnemo format" (special fasta format special)
HE
HE                   (see README.opscan)
HE  -------------------------------
HE  ----------------------------------------
HE   options for the Fastp part of Opscan
HE  ----------------------------------------
HE
HE  ----- Scoring options  -----------------
HE
HE  -k integer   : [K]uple size (integer)
HE                 default: -k 2
HE
HE  -L integer   : [L]ower threshold (integer:0-100)
HE                 similarity lower threshold in fastp for
HE               subsequent exclusion in bestfit search
HE                 [default: -L 5]
HE               Range: 0 (no similarity)-100 (complete
HE               similarity)
HE
HE  -x           : Fastp score is computed as the number of
HE                 kuples which match for the best shift between
HE                 the two sequences (default: off, the Fastp score
HE                 is computed based on the alignment score of the
HE                 best shift alignment)
HE
HE  -i integer   : Only in case of -x option, the integer gives
HE                 the diagonal [I]ntegration in fastp computations
HE                 default:  -i 2       (when -x option, else no
HE                                  integration is done)
HE
HE  ----- Report options          ------------------
HE
HE  -p filename  : [P]rint out alignments into filename(*)
HE                 default = Off
HE
HE  ----------------------------------------
HE   options for bestfit
HE  ----------------------------------------
HE
HE  -l           : [L]ocal alignment (BestFit)
HE                 default = On
HE  -g           : [G]lobal alignment
HE                 default = Off
HE
HE  -t <integer> : bestfit score threshold (0-100) for a couple
HE                 operon gene / genomic gene to be considered as
HE                 homolog      [default 40]
HE
HE  ----- Gap & Weights options  ------------------
HE  -M filename  : Scoring [M]atrix
HE                 default = <internal> Identity
HE  -o ff        : Gap [O]pening weigth
HE                 default = 1.2
HE  -e ff        : Gap [E]xtension weigth
HE                 default = 0.8
HE
HE  ----- Report options          ------------------
HE  -P           : [P]rint out alignment
HE                 default = Off
HE  ----------------------------------------
HE  file formats
HE
```

```
HE  matrix_file:    blast format
HE  --------------------------------
HE
```