PepLine User's guide

Version v2.0.3 - May. 2007

Contacts :

Alain.Viari@inrialpes.fr

1 Overview

PepLine is software package to map MS/MS fragmentation spectra on genomic DNA sequences. The theory is described in (1) and will not be detailled here (we strongly suggest to read the paper before). The purpose of this documentation is to provide information about the package implementation and practical usage.

(1) Ferro M, Tardif M, Reguer E, Cahuzac R, Bruley, C, Vermat T, Nugues E, Vigouroux M, Vandenbrouck Y, Garin J and Viari A. Pepline: a software pipeline for high-throughput direct mapping of tandem mass spectrometry data on genomic sequences.
J. Proteome Res. 2008 May;7(5):1873-83. doi: 10.1021/pr070415k Epub 2008 Mar 19.

1.1 Package distribution

```
+ config   : configuration files (for recompilation)
|
+ data     : runtime configuration files
| |
| + aa_mono.ref     : mass of normal and modified aminoacids
| |
| + trypsine.ref    : trypsic patterns
|
+ doc      : documentation
|
+ include  : C headers files (for recompilation)
|
+ ports    : Binaries
| |
| + <portname>  : port name
|   |
|   + bin       : binaries for port <portname> (see below)
|
+ samples  : sample files
|
+ src      : C source files (for recompilation)
  |
  +lib           : for libraries
  |
  +prog          : for programs
```

Unless you want to recompile the package, the only directories you need are :
ports/<portname>/bin (binaries) and **data** (runtime configuration files).

The binaries are located in **PepLine/ports/<portname>/bin**

where <portname> is one of :

<portname>	compiled for
i386-linux	Linux - i386
ppc-darwin	MacOSX - PPC
i386-darwin	MacOSX - Intel
x86-win32	Win32 - i386

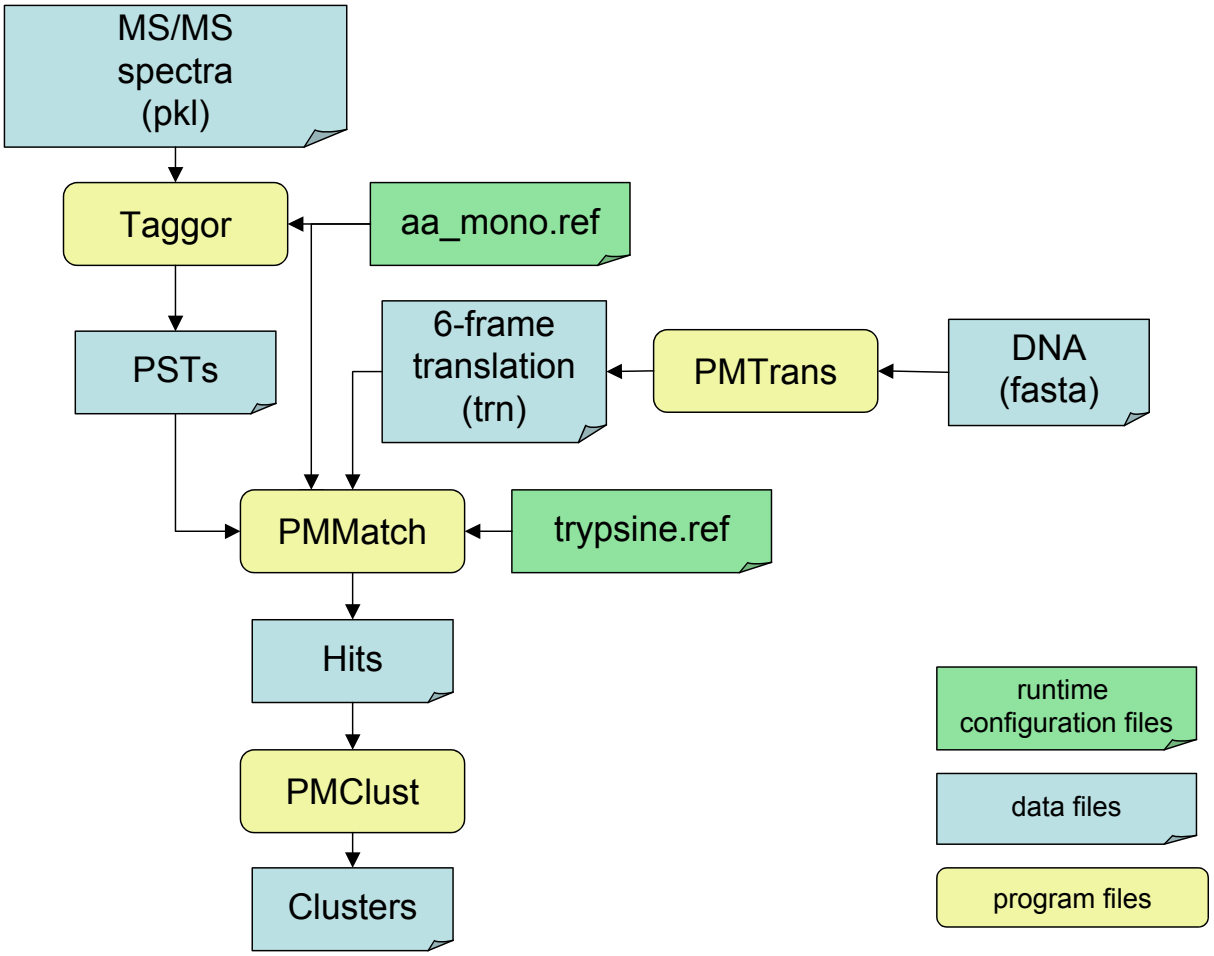
Note: In the rest of this document we shall assume that the proper binaries have been included in your path.

1.2 Pipeline overview

The overall pipeline is composed of four binaries :

Taggor, PMTrans, PMMatch and PMClust

and is depicted on the following figure :



2 Programs usage

Note: all programs take an optional **-h** argument, providing help on program usage

2.1 From MS/MS spectra to PSTs: Taggor

Taggor takes a spectra file as input and provides a PSTs file as output

2.1.1 Usage

```
Taggor [options]

 options:

 -a aminoAcidFile : aminoacid reference file
                    (default = "aa_mono.ref")
 -c               : turn ON tag correlation mode in score calculation
                    (default = OFF)
 -d tolerance     : mass tolerance in ppm
                    (default = 50)
 -F pkl | dta     : spectrum format
                    (default = pkl)
 -h               : print this help
 -i SpectrumFile  : spectrum filename
                    (default = <stdin>)
 -l tagLength     : length of tags
                    (default = 3)
 -m threshold     : spectrum low mass threshold
                    (default = 0.000000)
 -n param         : set denovo parameter
                    use <= 0 to turn off denovo calculation
                    (default = 1.0)
 -o outputFile    : output (usually PST) filename
                    (default = <stdout>)
                    note : with the -R option this file contains
                           the recalibrated spectra
 -r nbResult      : max number of PSTs per spectrum
                    (default = 10)
 -R               : recalibrate spectra using tags
                    (default = off)
                    note : with this option no tags are printed,
                           the output file contains the recalibrated
spectra
 -s zscore        : minimum PSTs score
                    (default = 0)
 -S R | E | S | T : score statistics
                    R : rank statistics
                    E : empirical PST distribution
                    S : shuffled spectrum distribution
                    T : theoretical spectrum distribution
                    (default = R)
                    note : R|E|S|T can be immediately (no space)
                           followed by digits interpreted as
                           a statistic optional parameter
                           default for R = 10
                           default for EST = 0
 -t threshold     : spectrum intensity threshold (in %)
                    (default = 0.100000)
 -T threshold     : spectrum rank threshold (<=0 means no threshold)
                    (default = 50)
 -u               : unique PST mode
                    remove duplicate PST per spectrum
                    (default = off)
 -U               : unique lexicon mode
                    remove duplicate lexicon per spectrum
```

```
                    (default = off)
 -v                : verbose mode (add some internal information)
                    (default = off)
```

note: -S and -c options are still experimental (you better have to use default values)

note: the allowed error on mass is M = M0 ± dM
where dM = (M0+100) * dM/M
and dM/M is given by the -d (tolerance) parameter in ppm

for instance, a tolerance of 50 ppm (default) allows for a mass error of ± 0.055 amu at 1000 amu.

2.1.2 Example

```
$ cd samples

$ Taggor -a ../data/aa_mono.ref -i 132134F_ME.pkl -o 132134F_ME.pst
```

note: the AA configuration file defaults to "aa_mono.ref" in the current directory. You may either specify the path to this file (as in the example before) or copy it to your current working directory.

2.1.3 Input file format

Accepted spectrum formats (-F option) are :

pkl : standard pkl format (default)
first line (parent ion) is :
[M+zH]/z intensity z

PKL : modified pkl format
first line (parent ion) is :
[M+H] intensity z

dta : mono pkl format
first line (parent ion) is :
[M+H] intensity

PKL and pkl are multi-spectrum formats
dta is single spectrum format

Please check carefuly what your data file format, since using wrong format will yield completely wrong results

2.1.4 Output file format (PSTs)

the PSTs are grouped by spectrum from the input file. Each group reads as :

```
# Tags generated from spectrumId (mass = ParentMass)
PSTId NTermMass CTermMass Tag ParentMass Score  // repeated foreach PST
%EndOfTags
```

with

spectrumId : spectra_filename **'.'** spectrumIndex
where spectrumIndex goes from 1 to number of spectra
ParentMass : mass of the Parent Ion (M+H)

PSTId : spectrumId **'.'** pstIndex
where pstIndex goes from 1 to number of PSTs generated for this spectrum
NTermMass : mass of the NTerm part of the PST in amu
CTermMass : mass of the CTerm part of the PST in amu
Tag : PST Tag

note: Taggor does not distinguish between L/I and Q/K. Therefore the tag part makes use of 'I' and 'K' symbols and no 'L' and 'Q'.

ParentMass : mass of the Parent Ion (M+H)
Score : PST score

example :

```
# Tags generated from 132134F_ME.1 (mass = 1496.677368)
132134F_ME.1.1 432.137329 816.412415 AGT 1496.677368 0.757858
132134F_ME.1.2 375.110352 917.438232 GAG 1496.677368 0.694959
132134F_ME.1.3 503.170349 759.382812 GTG 1496.677368 0.64842
132134F_ME.1.4 260.111450 974.488647 DGA 1496.677368 0.64842
132134F_ME.1.5 560.220764 645.348206 TGN 1496.677368 0.517632
%EndOfTags
# Tags generated from 132134F_ME.2 (mass = 1558.676758)
%EndOfTags
# Tags generated from 132134F_ME.3 (mass = 1428.786987)
132134F_ME.3.1 573.295410 565.314453 VTA 1428.786987 0.784584
132134F_ME.3.2 472.247986 636.345215 TVT 1428.786987 0.694959
132134F_ME.3.3 385.227661 737.376709 STV 1428.786987 0.63728
132134F_ME.3.4 256.192505 836.473206 EST 1428.786987 0.626332
132134F_ME.3.5 0.000000 1024.540894 KKE 1428.786987 0.574349
%EndOfTags
```

Note: there is an output group even if Taggor did not generate any PST at all (see example)

2.1.5 Runtime configuration file : aa_mono.ref

This file contains information about normal and modified amino-acids.

In its current version Taggor only considers normal amino-acids (the rest of the file is ignored by Taggor but will be used latter by othe programs (like PMMatch))

2.1.6 Taggor recalibration mode

Taggor can be used to recalibrate input spectra (-R option) using predicted PSts. In that mode no PSTs are printed. Instead, the output contains the recalibrated spectra. We suggest to use high threshold parameters in order to recalibrate on "sure" PSTs. Typical values are : -r 5 and -s 0.8

```
$ cd samples

$ Taggor -a ../data/aa_mono.ref -r 5 -s 0.8 -R < 132134F_ME.pkl > 
132134F_ME_RECALIBRATED.pkl

or

$ Taggor -a ../data/aa_mono.ref -r 5 -s 0.8 -R -i 132134F_ME.pkl -o 
132134F_ME_RECALIBRATED.pkl
```

Note: the recalibration process assume that all spectra correspond to the same acquisition run. Do not use it on a concatenated PKL file (unless you are sure of what you're doing).

2.2 Six-frame translation of DNA sequence(s) : PMTrans

PMTrans will translate a DNA file (fasta format) into six-frame translations for further use by PMMatch.

2.2.1 Usage

```
PMTrans [options]
 options:
   -c code         : genetic code to use
             code  : species
                0  : Universal
                1  : Mitochondrial Yeast
                2  : Mitochondrial Vertebrates
                3  : Mitochondrial Filamentous Fungi
                4  : Mitochondrial Insects and Platyhelminthes
                5  : Candida cylindracea
                6  : Ciliata
                7  : Euplotes
                8  : Mitochondrial Echinoderms
                     (default = 0)
 -h                : print this help
 -i fastaFile      : input chromosome(s) filename (fasta format)
                     (default = <stdin>)
                     note: may contain more than one sequence
 -o outputFile     : output translated chromosome(s) filename
                     (default = <stdout>)
 -v                : verbose mode (add some internal information)
                     (default = off)
```

2.2.2 Example

```
$ cd samples

$ PMTrans -i arabido.fst -o arabido.trn
```

2.2.3 Input file format

The chromosome(s) file should be in fasta format :

```
>name  [optional comments]
sequence on any number of lines (any number of chars per line)
```

Note: the sequence may be lower of upper case chars

Note: the input file may contain more than one chromosome.

examples

```
>chc
ATGGGCGAACGACGGGAATTGAACCCGCGATGGTGAATTCACAATCCACT
GCCTTAATCCACTTGGCTACATCCGCCCCTACGCTACTATCTATTCTTTT
```

```
TTGTATTGTCTAAAAAAAAAAAAAAATACAAATTTCAATAAAAAATAAAA
AAAGGTAGCAAATTCCACCTTATTTTTTTTCTAATAAAAAATATATAGTA
....
>chm
ggatccgttcgaaacaggttagcctactataatataaggattggattcta
ataagttcgaaacaggttagccttagcctactataggattagatctttct
tatcaacctactaacttcttccttgttgggatgagaaaccctttgcaac
caagcgtgctttgagtttgtcaagggacccatctgcattcagtttcactc
```

2.2.4 Output file format

The output file is also in fasta format. However the sequence name encode some information regarding the frame of the original DNA sequence. Please don't modify this file by hand.

2.3 From PSTs to hits : PMMatch

The purpose of PMMatch is to match PSTs found by Taggor on the six-frame translation of DNA sequence(s). A hit correspond to the location of a PST on a translated sequence.

2.3.1 Usage

```
PMMatch [options]

 options:
 -a AminoAcidFile : aminoacid reference file
                    (default = "aa_mono.ref")
 -d tolerance     : mass tolerance in ppm
                    (default = 50)
 -e EnzymeFile    : digestion enzyme reference file
                    (default = "trypsine.ref")
 -f FastaFile     : translated chromosome filename
                    (default = NONE) *you should provide this filename*
 -h               : print this help
 -i PSTFile       : PSTs filename
                    (default = <stdin>)
 -k               : differentiate between K/Q
                    (default = off)
 -m miscleavage   : maximum number of miscleavage(s)
                    (default = 1)
 -M overcleavage  : maximum number of overcleavage(s) (0, 1 or 2)
                    (default = 0)
 -n maxModif      : maximum number of modified residues
                    (default = 0)
 -o outputFile    : output (hits) filename
                    (default = <stdout>)
 -p               : report partial hits too
                    (default = off)
 -v               : verbose mode (add some internal information)
                    (default = off)
 -y metBefore     : number of positions to scan for Nterm Methionine
                    0 means no Nterm scan
                    (default = 1)

note 1: -S (and -s) options is still *highly* experimental
```

2.3.2 Example

```
$ cd samples

$ PMMatch -a ../data/aa_mono.ref -e ../data/trypsine.ref -f arabido.trn -i
132134F_ME.pst -o 132134F_ME.hit
```

note: the configuration files ("aa_mono.ref" adn trypsine.ref" are search for in the current directory. You may either specify the path to theses file (as in the example before) or copy them to your current working directory.

2.3.3 Input files format

PST file : see Taggor output file format

Six-Frame translation : see PMTrans output file format

2.3.4 Output file format

The hits are grouped, in the output file, output by sequence and by PST.
Each group of hits reads as :

```
SequenceName 'D|R' frame SeqLength [Sequence Comment]
PST_output_line_from_Taggor
repeat foreach hit
F|N|C posMn posMc posTn posTc pepSeq pepMatch nbMis nbOver mbMod score
%EndOfHits
```

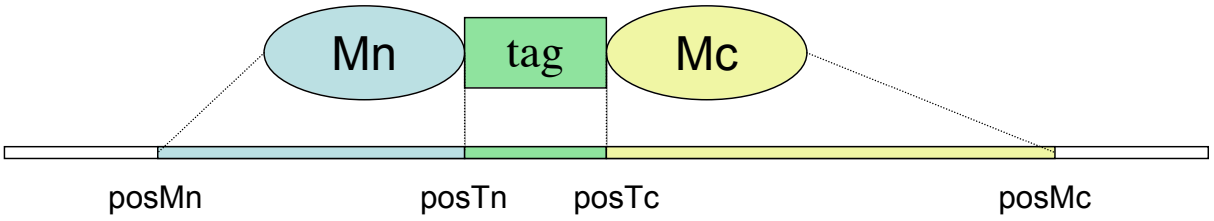
with

SequenceName : name of DNA sequence (from fasta file)
D|R : D for direct strand R for reverse strand
frame : translation frame (1, 2 or 3)
SeqLength : length of original DNA sequence

PST_output_line_from_Taggor : PST line from Taggor
PSTId NTermMass CTermMass Tag ParentMass Score
(see Taggor output format)

F|N|C : Type of hit
F : Full hit : both N and C term masses match
N : N_Partial hit : C term mass is missing
C : C_Partial hit : N term mass is missing

posMn : position of the N term part of PST on the proteic sequence (see figure)
posMc : position of the C term part of PST on the proteic sequence (see figure)
posTn : start position of the tag part of PST on the proteic sequence (see figure)
posTc : end position of the tag part of PST on the proteic sequence (see figure)



Note: all positions are given on the translated sequence

Note: for partial hits (N or C types), the missing posMx position is indicated by 0.

pepSeq : the matched peptide as it appears on the translated sequence.

note: pepSeq is flanked by the preceeding and following symbol on the sequence :
<preceding>.peptide.<following>

special flanking symbols are '*' for sequence extremities and 'X' for a stop codon

pepMatch : the matched peptide as matched by PMMatch, i.e. with possible modifications. if present modified amino-acids are indicated by lowercase characters. Moreover the tag part of the PST is indicated between brackets.

nbMis, nbOver : number of mis- and over-cleavages. Normally, a hit should be flanked by two enzyme cut sites or a stop codon on the Cterm or a Met on a N-Term and no cut sites should appear within the peptide.
An extra cut site within the peptide is called a **miscleavage**.
A missing cut site at any extremity is called an **overcleavage**

note: the maximum number of mis- and over-cleavage can be set by the -m and -M options

note: a stop codon at the C-term part is not counted as an overcleavage

note: a Methionine at the N-term part is not counted as an overcleavage unless the -y option is set to 0.

nbMod : number of modified aa in the peptide.

note: allowed modifications are indicated in the 'aa_mono.ref' runtime configuration file (see below).

note: the maximum number of modified aa can be set with the -n option

note: in the current version, no modifications are allowed in the tag part of the PST, only modifications in the N or C terminal masses are taken into account.

score: PST score. In the current version this is simply the PST score. This will be changed in the future.

Example 1

```
chc D 1 154478 <no comment>
132134F_ME.13.1 530.258423 625.336609 PHG 1465.760742 0.50874
F 18466 18478 18471 18473 K.TFQGPPHGIQVER.D TFQGP[PHG]IQVER 0 0 0 0.508740
%EndOfHits
chc D 1 154478 <no comment>
132134F_ME.20.1 228.111389 682.369324 AVE 1228.646973 0.917004
F 18755 18765 18757 18759 R.DLAVEGNEIIR.E DL[AVE]GNEIIR 0 0 0 0.917004
%EndOfHits
```

Example 2

```
chc D 1 154478 <no comment>
132134F_ME.3.1 573.295410 565.314453 VTA 1428.786987 0.784584
N 32386 0 32391 32393 M.PDHELVTA.* PDHeL[VTA] 0 1 1 0.784584
%EndOfHits
chc D 1 154478 <no comment>
```

```
132134F_ME.3.5 0.000000 1024.540894 KKE 1428.786987 0.574349
N 331 0 331 333 L.QQE.* [KKE] 0 1 0 0.574349
%EndOfHits
chc D 1 154478 <no comment>
132134F_ME.4.5 0.000000 764.344604 IVE 1124.579956 0.545254
N 1712 0 1712 1714 L.LVE.* [IVE] 0 1 0 0.545254
N 4755 0 4755 4757 G.LVE.* [IVE] 0 1 0 0.545254
%EndOfHits
chc D 1 154478 <no comment>
132134F_ME.6.3 130.119995 799.402710 KPA 1244.656372 0.564482
C 0 21416 21407 21409 *.QPAMNREGIV.M [KPA]MNREGIV 1 1 0 0.564482
%EndOfHits
chc D 1 154478 <no comment>
132134F_ME.11.1 356.167725 369.238190 IDE 1101.574829 0.659754
N 36753 0 36756 36758 D.LDKIDE.* LDK[IDE] 1 1 0 0.659754
%EndOfHits
chc D 1 154478 <no comment>
132134F_ME.11.3 228.102722 498.277618 KID 1101.574829 0.564482
N 18791 0 18793 18795 T.IDKLD.* ID[KID] 1 1 0 0.564482
N 36753 0 36755 36757 D.LDKID.* LD[KID] 1 1 0 0.564482
%EndOfHits
chc D 1 154478 <no comment>
132134F_ME.13.1 530.258423 625.336609 PHG 1465.760742 0.50874
F 18466 18478 18471 18473 K.TFQGPPHGIQVER.D TFQGP[PHG]IQVER 0 0 0 0.508740
%EndOfHits
chc D 1 154478 <no comment>
132134F_ME.16.3 489.233948 325.155701 NSK 1162.583740 0.707107
N 40722 0 40726 40728 I.SYEKNSQ.* SYeK[NSK] 1 1 1 0.707107
%EndOfHits
```

2.3.5 Runtime configuration file : aa_mono.ref

This file contains information about normal and modified amino-acids.

You may want to add you own AA modifications as :

```
M one_letter_code AA_Name AA_Mass Original_AA Location Probability
```

M : mandatory key
one_letter_code : use a lowercase symbol
AA_name : any single word (no space)
AA_mass : modified AA mass in amu
Original AA : one letter code of unmodified AA
Location: one of X, N, C to indicate the location modification :
anywhere : X
N-terminal : N
C-terminal : C
probability : frequency of modification <currently unused>

examples:

```
M       k       Carbomethyl_Cystein         161.01466 C X 0.01
M       a       Acrylamido_Cysteine         174.04629 C X 0.01
M       e       Pyroglutamic_Acid           111.03203 E X 0.01
```

2.3.6 Runtime configuration file : trypsine.ref

This file contains information about enzyme cut sites that should be used to define the boundaries of trypsic peptides. Default values are for standard trypsin.

You may want to add your own enzyme, using the following format :

```
for each enzyme repeat :

(line 1) enzyme_name number_of_sites over_cleavage_proba
(line 2 to 2+n) cut_proba cut_offset left_cut_regex right_cut_regex
```

enzyme_name : single word (no space)
number_of_sites : number of cut sites
over_cleavage_proba : <unused> -> put 0.0

cut_proba : <unused> -> put 0.0
cut_offset : location of cut relative to the beginning of left regular expression
left_cut_regex : regular expression to the left of the cut
right_cut_regex : regular expression to the right of the cut

example : for trypsine

```
Trypsine 1 0.0
0.0 1 [K|R]$ ^[^P]
```

reads as :
1 cut site for trypsine
cut site is : K or R not followed by P
cut location is right after K or R

note: please use "$" at the end of the left_cut regex and "^" at the beginning of the right cut regex to make sure of the position of your regular expression.

2.4 From hits to clusters : PMClust

PMClust clusterizes hits into clusters and projects hits position back to the original (untranslated) DNA sequence.

2.4.1 Usage

```
PMClust [options]

 options:

 -d distance      : clustering distance parameter (in nucleotides)
                    (default = 3000)
 -f               : force to cluster hits in the same frame
                    (this is used for procaryota only)
                    (default = off)
 -h               : print this help
 -i HitFile       : Hits filename
                    (default = <stdin>)
 -o outputFile    : output (cluster) filename
                    (default = <stdout>)
 -p               : add partial hits in clusters
                    (default = off)
 -P               : cluster in Protein mode
                    (default = Nucleic mode)
 -t threshold     : report only clusters with at least threshold full hits
                    (default = 3)
 -T threshold     : report only cluster with at least threshold pephits
                    (default = 1)
 -v               : verbose mode (add some internal information)
                    (default = off)

note: in Protein mode (-P) , parameters -d and -f are ignored
```

2.4.2 Example

```
$ cd samples

$ PMClust -i 132134F_ME.hit -o 132134F_ME.cln
```

2.4.3 Input file

Hit file : see PMMatch output file

2.4.4 Output file

The output file contains cluster in the following format :

```
CLUSTER idCluster nbHits nbFullHits nbPepHits score
SequenceName 'D|R' frame SeqLength [Sequence Comment]
// the two following lines are rpeated for each hit in the cluster
```

```
PST_output_line_from_Taggor
HIT_output_line_from_PMMatch
%EndOfCluster
```

with

idCluster : an integer starting at 1
nbHits : total number of hits (full + partials)
nbFullHits : number of full hits
nbPepHits : number of different peptides
score : score of cluster (in the current version the sum of full hits scores, this will be modified in the future)

SequenceName : name of DNA sequence (from fasta file)
D|R : D for direct strand R for reverse strand
frame : translation frame (1, 2 or 3)
SeqLength : length of original DNA sequence

note: this line is identcal to the sequence line in the output of PMMatch

PST_output_line_from_Taggor : PST line from Taggor
PSTId NTermMass CTermMass Tag ParentMass Score
(see Taggor output format)

HIT_output_line_from_PMMatch : HIT line from PMMatch
F|N|C posMn posMc posTn posTc pepSeq pepMatch nbMis nbOver mbMod score
(see PMMatch output format)

important note: the posMn, posMc, posTn, posTc positions are now given as **nucleotide positions** on the original DNA sequence.

example :

```
CLUSTER 2 10 10 3 7.87444
chm R 2 366924  <no comment>
132134F_ME.256.4 487.248474 382.266693 FVD 1249.714355 0.771105
F 77610 77642 77619 77627 R.ESTLGFVDLLR.D ESTLG[FVD]LLR 0 0 0 0.771105
chm R 2 366924  <no comment>
132134F_ME.256.1 317.137634 596.378540 IGF 1249.714355 0.933033
F 77610 77642 77625 77633 R.ESTLGFVDLLR.D EST[IGF]VDLLR 0 0 0 0.933033
chm R 2 366924  <no comment>
132134F_ME.256.2 430.226685 497.295624 GFV 1249.714355 0.870551
F 77610 77642 77622 77630 R.ESTLGFVDLLR.D ESTL[GFV]DLLR 0 0 0 0.870551
chm R 2 366924  <no comment>
132134F_ME.256.3 216.092896 743.447510 TIG 1249.714355 0.784584
F 77610 77642 77628 77636 R.ESTLGFVDLLR.D ES[TIG]FVDLLR 0 0 0 0.784584
chm R 2 366924  <no comment>
132134F_ME.95.2 410.195129 156.098907 VID 912.479614 0.757858
F 77751 77774 77754 77762 R.AMHAVIDR.Q AMHA[VID]R 0 0 0 0.757858
chm R 2 366924  <no comment>
132134F_ME.95.4 202.107422 384.215210 HAV 912.479614 0.671286
F 77751 77774 77760 77768 R.AMHAVIDR.Q AM[HAV]IDR 0 0 0 0.671286
chm R 2 366924  <no comment>
```

```
132134F_ME.136.3 286.099121 543.299744 III 1187.681030 0.671286
F 77775 77804 77787 77795 R.DNGLLLHIHR.A DNG[III]HIHR 0 0 0 0.671286
chm R 2 366924  <no comment>
132134F_ME.136.4 625.362915 156.100113 HIH 1187.681030 0.671286
F 77775 77804 77778 77786 R.DNGLLLHIHR.A DNGLLL[HIH]R 0 0 0 0.671286
chm R 2 366924  <no comment>
132134F_ME.136.2 512.285217 293.155792 IHI 1187.681030 0.82645
F 77775 77804 77781 77789 R.DNGLLLHIHR.A DNGLL[IHI]HR 0 0 0 0.826450
chm R 2 366924  <no comment>
132134F_ME.136.1 399.200623 406.243408 IIH 1187.681030 0.917004
F 77775 77804 77784 77792 R.DNGLLLHIHR.A DNGL[IIH]IHR 0 0 0 0.917004
%EndOfCluster
```