Date: 2010-10-14

JCVI Metagenomics Annotation Pipeline Process Documentation

Step 1 Split Sequences

Split sequences for parallel searching (Steps 2-6)

executable	split_multifasta.pl
input	fasta file
output	multiple split fasta files
command	split_multifasta.pl --input_file=input.fasta --output_dir=/tmp --output_list=/tmp/split.list --output_file_prefix='split_' --seqs_per_file=50000 --compress_output=0

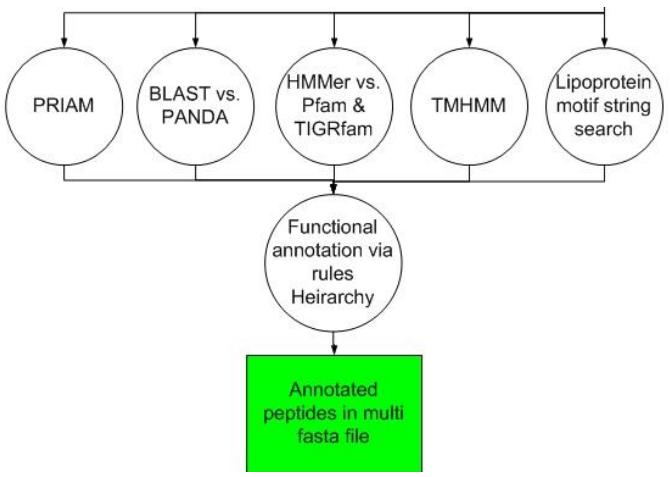
PRIAM
BLAST vs. PANDA
HMMer vs. Pfam & TIGRfam
TMHMM
Lipoprotein motif string search
Functional annotation via rules Heirarchy
Annotated peptides in multi fasta file

Figure 1 Overview (Steps 2-6 are executed in parallel)

Step 2 HMM Component (Pfam & TIGRfam)

Step2.1 Run HMM search

executable	hmmpfam
input	split fasta file
output	HMM raw output (ldhmmpfam_out.raw)
command	hmmpfam --threads 1 ALL_LIB_bin.HMM split1.fasta

Step2.2 Parser HMM results and generate HMM tab delimited file (JCVI HTAB)

Parses output files generated by hmmpfam

executable	camera_htab.pl
input	split HMM raw files (ldhmmpfam_out.raw)
output	split HTAB files (ldhmmpfam_out.htab)
command	camera_htab.pl ldhmmpfam_out.raw > ldhmmpfam_out.htab

step 2.3 Parse JCVI HTAB

Performs HMM defline lookups for common name, gene symbol, GO, EC, and TIGR Roles from HMM defline flat-file index (data/hmm-index). Classifies HMM hits based on the HMM Iso-Type (10 classes, see box below).

executable	camera_parse_annotation_results_to_text_table.pl
input	JCVI HTAB (ldhmmpfam_out.htab)
output	JCVI HTAB parsed(ldhmmpfam_out.htab.parsed)
command	camera_parse_annotation_results_to_text_table.pl --input_file ldhmmpfam_out.htab --input_type HTAB --output_file ldhmmpfam_out.htab.parsed --work_dir /tmp

HMM ISO-TYPES

```
if ($iso_type =~ /^(equivalog)$|^(PFAM_equivalog)$/) {
         $type .= 'Equivalog';
     } elsif ($iso_type =~ /^(hypoth_equivalog)$/) {
         $type .= 'HypotheticalEquivalog';
     } elsif ($iso_type =~ /^(exception)$/) {
         $type .= 'Exception';
     } elsif ($iso_type =~ /^(subfamily)$/) {
         $type .= 'Subfamily';
    } elsif ($iso_type =~ /^(superfamily)$/) {
         $type .= 'Superfamily';
     } elsif ($iso_type =~ /^(equivalog_domain)$|^(PFAM_equivalog_domain)$/) {
         $type .= 'EquivalogDomain';
     } elsif ($iso_type =~ /^(hypoth_equivalog_domain)$/) {
         $type .= 'HypotheticalEquivalogDomain';
     } elsif ($iso_type =~ /^(subfamily_domain)$/) {
         $type .= 'SubfamilyDomain';
     } elsif ($iso_type =~ /^(domain)$/) {
         $type .= 'Domain';
     } elsif ($iso_type =~ /^(PFAM)$/) {
         $type .= 'Uncategorized';
     } else {
         $type = '';
     }
```

Step 3 BLAST Component

Step 3.1 Run BlastP

Run blastp on individual fasta split files and generate JCVI BTAB format from blast XML output (-m 7 option)

executable	blastall
input	split fast file
output	Blast results in XML format
command	blastall -v 10 -b 10 -X 15 -e 1e-5 -M BLOSUM62 -J F -K 10 -f 11 -Z 25.0 -W 3 -U F -I F -E -1 -y 7.0 -G -1 -A 40 -Y 0.0 -F "T" -g T -p blastp -z 1702432768 -m 7'

Step 3.2 Convert XML files to JCVI tab delimited blast result files (BTAB)

executable	blast_xml_to_btab.pl
input	Blast XML results
output	JCVI BTAB
command	blast_xml_to_btab.pl < blast_out.xml > blast_out.btab

Step 3.3 Parse JCVI BTAB

Perform PANDA defline lookups for common name, gene symbol, GO, EC, is characterized from PANDA flat-file index (will be changed to use a UniRef100 lookup table). Classifies blast hits based on coverage and identity (4 classes, see box below).

executable	camera_parse_annotation_results_to_text_table.pl
input	JCVI BTAB (blast_out.btab)
output	JCVI BTAB parsed
command	camera_parse_annotation_results_to_text_table.pl --input_file blastp_out.btab --input_type BTAB --output_file blastp_out.btab.parsed --work_dir /tmp

```
if ($characterized && $pct_id >= 35 && $pct_cov >= 80){
 "PandaBLASTP::Characterized"; }
elsif ($pct_id >= 35 && $pct_cov >= 80){
 "PandaBLASTP::HighConfidence";
}
elsif ($pct_id < 35 && $pct_cov >= 80){
"PandaBLASTP::Putative";
}elsif ($pct_id >= 35 && $pct_cov < 80){
"PandaBLASTP::ConservedDomain";
}
```

Step 4 Lipoprotein Motif Search

Step 4.1 Run lipoprotein motif search

Scans for membrane lipoprotein lipid attachment sites on amino acid sequence. Uses PROSITE motif (^.{0,6}[KR]).{0,18}[^DERK][^DERK][^DERK][^DERK][^DERK][^DERK][LIVMFWSTAG][LIVMFWSTAG][LIVMFWSTAGCQY][AGS]C).

executable	lipoprotein_motif.pl
input	split fast file
output	BSML formatted file
command	lipoprotein_motif.pl --input split1.fasta --output lipoprotein_out.bsml --gzip_output 0 --id_repository workflow/project_id_repository --is_mycoplasm 0

Step 4.2 Parse lipoprotein motif results

executable	camera_parse_annotation_results_to_text_table.pl
input	BSML formatted file (lipoprotein_out.bsml)
output	BSML parsed file (lipoprotein_out.bsml.parsed)
command	camera_parse_annotation_results_to_text_table.pl --input_file lipoprotein_out.bsml --input_type LipoproteinMotifBSML --output_file lipoprotein_out.bsml.parsed /peptide.fasta.q1_q10_1532122841942589727.bsml.parsed --work_dir /tmp

Step 5 TMHMM Search

Scans proteins for trans-membrane domains.

Step 5.1 Run TMHMM

executable	tmhmm
version	2.0
input	split fast file
output	tmhmm_out.raw
command	tmhmm split1.fasta > tmhmm_out.raw

Step 5.2 Parse TMHMM results

executable	tmhmm2bsml.pl
version	2.0
input	TMHMM raw file (tmhmm_out.raw)
output	BSML formatted file (tmhmm_out.bsml)
command	tmhmm2bsml.pl --input tmhmm_out.raw --output tmhmm_out.bsml --fasta_input split1.fasta --compress_bsml_output 0 --id_repository workflow/project_id_repository

Step 5.3 Parse TMHMM BSML

executable	camera_parse_annotation_results_to_text_table.pl
version	2.0
input	TMHMM BSML file (tmhmm.bsml)
output	parsed BSML formatted file (tmhmm_out.bsml.parsed)
command	camera_parse_annotation_results_to_text_table.pl --input_file tmhmm_out.bsml --input_type TMHMMBSML --output_file tmhmm_out.bsml.parsed --work_dir /tmp

Step 6 PRIAM Search

Step 6.1 Run Reverse PSI Blast (produce tab delimited blast output)

executable	rpsblast
version	2.2.15
input	split fast file
output	RPS Blast Hits (priam_out.raw)
command	rpsblast -i split1.fasta -d /data/priam-index/priam_jun09_gene -m 8 -e 1e-10 > priam_out.raw

Step 6.2 Replace PRIAM IDs with EC IDs

executable	expandPriToEcHitLines.pl
input	RPS Blast Hits (priam.raw)
output	Blast Hits (priam_out.ectab)
command	expandPriToEcHitLines.pl -m data/priam-index/defline_map.txt -i priam_out.raw -o priam_out

Step 6.3 Generate ECTAB (tab delimited format)

executable	create_ec_list.pl
input	priam_out
output	tab delimited EC results (priam_out.ectab)
command	create_ec_list.pl --rps --hits priam_out --output priam_out.ectab

Step 6.4 Parse ECTAB (tab delimited format)

executable	camera_parse_annotation_results_to_text_table.pl
input	tab delimited EC results (priam_out.ectab)
output	parsed EC results (priam_out.parsed)
command	camera_parse_annotation_results_to_text_table.pl --input_file priam_out.ectab --input_type ECTable --output_file priam_out.parsed --work_dir \tmp

Step 7 Annotation Rules

The final annotation for each peptide is being derived based on all previously collected evidences. How evidences are being used to assign the various annotation data types (common name, gene symbol, EC, GO, Tigr Role) is based on a evidence rules hierarchy in lib/CAMERA/AnnotationRules/PredictedProtein.pm.

Step 7.1 Concat parsed results obtained in previous steps into one file

executable	cat
input	all parsed files
output	out.cat.sorted
command	cat *.sorted > out.cat

Step 7.2 Sort the joined file

executable	sort
input	concatenated results (out.cat)
output	sorted concatenated results (out.cat.sorted)
command	sort --key=1,1 -T /tmp -S 1G -d -o out.cat.sorted out.cat

Step 7.3 Generate tab delimited annotation file (final output)

executable	camera_annotate_from_sorted_table.pl
input	Sorted concatenated files (out.cat.sorted)
output	tab delimited annotation results (annotation.tab)
command	camera_annotate_from_sorted_table.pl --input out.cat.sorted --output out.cat.tmp > annotation.tab

File Structure

bin	perl binaries
lib	perl libraries
data	lookup files
example	example files (contains test set)