Pan-Genomes Analysis Pipeline (PGAP)

Zhao Yongbing

October 10, 2011

Any suggestion or problems for PGAP please mail to pgap@big.ac.cn

1. What's PGAP?

PGAP is a **P**an-**G**enome **A**nalysis **P**ipeline and it has integrated several useful analysis methods in genome research into one Perl script, so it will make genome research easier.

2. What could PGAP do?

PGAP has five program sections, there are as following:

1) Orthologs clusters identification among multiple genomes
2) Pan-genome analysis for given strains
3) Functional genes variation and SNP calling among given strains
4) Evolutionary analysis based on pan-genome and SNP data
5) Orthologs clusters function analysis

3. How to install PGAP?

PGAP is Perl script, so it is unnecessary to complie PGAP. However, to do genome analysis better, several extra programs are invoked in PGAP. So, before running PGAP, user should set the path of those extra programs in the PGAP.pl source.

The following are the programs list:

1) blastall and formatdb in BLAST (2.2.12 or higher).
   BLAST is available from ftp://ftp.ncbi.nih.gov/blast/executables/release/
2) mafft.
   mafft is available from http://mafft.cbrc.jp/alignment/software/
3) dnaml, dnadist, neighbor, seqboot, consense in PHYLIP (version 3.69)
   PHYLIP package is available from http://evolution.gs.washington.edu/phylip.html
4) mcl
   mcl is available from http://micans.org/mcl/

4. Input data

There are three type files required for running the whole pipeline. They are protein sequences, and their corresponding nucleotide sequences and annotation files.

PGAP recognizes these three type files by their special extension names, ***.pep*** for protein sequences, ***.nuc*** for nucleotide sequences and ***.function*** for annotation file. For each strain, these three type file should has the same prefix, which we call **nickname**. All input file are named as ***nickname.pep***, ***nickname.nuc*** or ***nickname .function***. For different strains, the nickname should be different.

Take *Salmonella enterica* subsp. enterica serovar Typhi str. CT18 for example, we could name its three type files "CT18.pep, CT18.nuc, CT18.function" or "NC_003198.pep, NC_003198.nuc, NC_003198 .function" or others like these.

As for gene name, 16758994, STY0117, orf_0112 are ok. But the gene name should be unique in all strains. For each strain, the gene names should keep consistent in all the three files.

In protein sequences files (***.pep*** file), the sequence should be in FASTA format. The data are like the follows:

```
>16758994
MNRISTTTITTITITTGNGAG
>16762656
MLYKGCLMKSDVQLNLRAKESQRALIDAAAEILHKSRTDFILETACRAAENVILDRRVFNFNDEQYEEFIN
LLDAPVADDPVIEKLLARKPQWDV
```

In the nucleotide sequences files (***.nuc*** file), the sequence should be in FASTA format. The data are like the follows:

```
>16758994
ATGAACCGCATCAGCACCACCACCATTACCACCATCACCATTACCACAGGTAACGGTGCGGGCTGA
>16762656
ATGCTATACAAGGGGTGTCTCATGAAATCAGATGTTCAACTTAACCTTAGAGCTAAGGAGTCGCAGCG
GGCGCTCATTGATGCAGCTGCGGAAATCCTTCACAAGTCACGTACAGATTTCATTCTGGAAACGGCC
TGCCGGGCTGCCGAGAATGTGATCCTTGACCGCCGTGTATTTAACTTTAACGATGAGCAATATGAGGA
GTTCATCAATCTGCTTGATGCACCGGTCGCAGATGATCCCGTTATCGAAAAACTGCTGGCAAGGAAA
CCTCAGTGGGACGTGTAA
```

In the annotation files (***.function*** file) are as follows, the format requirements are as follows:

There are three columns, the first column is the gene name, the second is COG classification and the last one is function description. If the gene has no COG classification information, put "-" on the corresponding position. These three columns are **separated by <tab>** and there is **NO header** in this table. The data are like the follows:

```
16758994    -    thr operon leader peptide
16762656    COG4453S    hypothetical protein
```

If some data are download from NCBI ftp, the `Converter_finished.pl` and `Converter_draft.pl` could be used to converted and prepare the input data. As for `Converter_finished.pl and Converter_draft.pl` usage, it will be introduced latter.

5. **What does PGAP output?**

All output result files are named with a digital prefix (according to the order listed in ***what could PGAP do?***). For example, the result from orthologs clusters identification will begin with "1", so on and so forth.

1) Orthologs clusters identification among multiple genomes

1.Orthologs_Cluster.txt: cluster list detail, if some strain has no genes in the cluster, mark with "-".

1.Gene_Distribution_By_Conservation.txt: gene number in each strains by clusters conservation.

2) Pan-genome analysis for given strains

2.PanGenome.Profile.txt: *pan-genome and core genome function*

2.PanGenome.Data.txt and ***2.PanGenome.Profile.Median.txt*** *are the temporary data used for fitting pan-genome function.*

3) Genome variation and SNP calling among given strains

3.CDS.variation.txt is variation detail in CDS region among all given strains. There are seven columns (see the demonstrated figure).

The 1st column is the Cluster ID, which is consistent with the ID in *1.Orthologs_Cluster.txt*.

The 2nd column is the cluster conservation of current cluster.

The 3rd column is the variation position, which counts according to the alignment result of protein sequences in this cluster. For indel events, the position is an integer. For synonymous mutation and nonsynonymous mutation, the position is a floating number, in which the integer part marks the position of amino acid in the alignment result of protein sequences, while the decimal part mark the position of codon. *For example: 213.1 mean there are variation on the 21 th amino acid in the alignment result of protein sequences and the variation location on the 1st codon. 225.3 means that the variation location on the 3rd codon of the 225th amino acid.*

The 4th column shows the amino acid types on current position.

The 5th column shows the nucleotide types on current position, For Indel, only "-" will be given.

The 6th column shows all gene nucleotide profile in current position (for indel, amino acid will be listed). The order of nucleotide/amino acid is consistent with the gene order in current cluster in *1.Orthologs_Cluster.txt*.

The 7th column shows the variation type(Indel, synonymous and synonymous).

```
1»      7»      213.1»  I,V»    A,G»    GGGGAGG»nonsynonymous
1»      7»      225.3»  V»      A,T,G»  GTTGATT»synonymous
1»      7»      232.1»  S,A»    T,G»    GGGGTGG»nonsynonymous
1»      7»      234.3»  G»      T,C»    TCCCCCC»synonymous
2»      7»      1»      -,M»    -»      ---M-M-»InDel
2»      7»      2»      -,T»    -»      ---T-T-»InDel
2»      7»      3»      -,E»    -»      ---E-E-»InDel
```

3.Core.CDS.variation.txt: is a temporary DNA sequence file phylip format. This file records the variation in the core CDS region. Another difference between *3.Core.CDS.variation.txt* and *3.CDS.variation.txt* is that, if there is a variation in some amino acid, the corresponding three nucleotides are output in this file.

3.CDS.variation.analysis.txt: is the summery result for *3.CDS.variation.txt*.

4) Evolutionary analysis based on pan-genome and SNP

In the result of this part, all phylogenetic trees calculated by phylip are output in the ***.tree*** file, user could use visualization software to view the phylogenetic trees (Tips:

TreeView could be freely used to output the figure you prefer. It could be downloaded from http://darwin.zoology.gla.ac.uk/~rpage/treeviewx/).

5) Orthologs clusters function analysis

Orthologs_Cluster_Function.txt has listed the COG classification and function description for each cluster.

The remaining four file have recorded the COG distribution infomation for different clusters type.

6. Usage for *Converter_finished.pl* and *Converter_draft.pl*

`Converter_finished.pl` **and** `Converter_draft.pl` are Perl script for user to convert the data from NCBI ftp. It could help to convert three types' files from NCBI ftp(.faa, .ffn and .ptt) to the PGAP required formats (***.pep, .nuc*** and ***.function***). `Converter_finished.pl` could be used for convert finished genome data, while `Converter_draft.pl` could be used for draft genome data.

The following the usage for `Converter_finished.pl:`

perl `Converter_finished.pl`

-S input the accession number of the strains. If two or more strains are converted at one time, join the accession numbers with "+", and this requires that all files are in the same folder.

-I the directory of input files.

-O the directory for the converted data.

The following the usage for `Converter_ draft.pl:`

perl `Converter_ draft.pl`

-N Input the strain nickname. Give a new nickname for the strain, just as a identifier.

-I the directory of input files

-O the directory for the converted data.

Only one strain's data from draft genome could be converted at one time.

7. Tutorials for PGAP

Step 1 Preparing input data

Users could prepare input data from NCBI FTP with ***Converter_finished.pl*** and ***Converter_draft.pl*****,** which was included in the PGAP package.

1) For finished bacterial genome data from NCBI

First, put all strains' .ptt, .ffn and .faa files in the same folder. In the finished folder of the testdata directory, there are some test data from five *E. coli* strains: AC_000091.faa, AC_000091.ffn, AC_000091.ptt and so on. For these data, we could convert them one by one, or convert them together.

Convert One by one:

Take AC_000091 for example, we could convert AC_000091 data with the command like this:

```
perl Converter_finished.pl –S AC_000091 –I input_drectory –O output_directory
```

Convert all at once:

```
perl Converter_finsihed.pl –S AC_000091+NC_000913+NC_004431 –I input_drectory –O output_directory
```

2) For draft bacterial genome data from NCBI

Uncompressing each strain's .ffa.tgz, .ffn.tgz, and .ptt.tgz file in an individual folder. Then use the ***Converter_draft.pl*** script to convert each strain's data separately.

```
perl Converter_draft.pl –N nickname –I input_drectory –O output_directory
```

In the above command, different strain should be given a different nickname.

3) For genome data from users

All the three types input files are required in the formats mentioned in the PGAP manual.

Step 2 Preparing the third part programs and program path setting

To run PGAP well, four extra programs (BLAST, MAFFT, PHYLIP and MCL) are required. These programs could be easily installed according to their README files and manuals.

After these programs are installed, open the PGAP.pl file and replace each program's path in the file (line 6 to line 22).

Step 3 Running PGAP

For PAGP, there are five functional modules, cluster analysis of functional genes, pan-genome profile analysis, genetic variation analysis of functional genes, species evolution analysis and function enrichment analysis of gene clusters. The following commands could trigger all these analysis modules with the default parameters.

```
perl PGAP.pl –strains nickname1+nickname2+...+nicknameN –input input_directory –output output_directory --cluster --pangenome --variation --evolution --function –method MP
```

Or

```
perl PGAP.pl –strains nickname1+nickname2+...+nicknameN –input input_directory –output output_directory --cluster --pangenome --variation --evolution --function –method GF
```

--cluster --pangenome --variation --evolution and --function

denote whether to run cluster analysis of functional genes, pan-genome profile analysis, genetic variation analysis of functional genes, species evolution analysis and function enrichment analysis of gene clusters respectively. Theoretically, the entire five modules could be run alone, when the required input data are available. For the first time to analysis a batch of new strains, *--cluster* is required.

--thread could be used to accelerate the process in cluster analysis of functional genes, when *--cluster* is added.
--evalue, *--score*, *--identity*, *--coverage* (or *--local* and *--global*) could be used for adjust the cutoff for homologs identification. *--coverage* is valid when *--method GF* is used, while *--local* and *--global* is valid when *--method MP* is used.

--bootstrap could be used for adjust the boorstrap times when *--evolution* is used