Genome Topology Network

The genome topology network (GTN) is a new approach for studying the phylogenetics of bacterial genomes by analysing their gene order. It can be downloaded from

https://github.com/0232/Genome_topology_network

Installation

GTN is wrote in Perl and be performed in LINUX system only.

Firstly you need to download the package from github website and unzip it:

unzip GTN_2.0.zip

Add the GTN path to your perl library path just like this:

echo 'export PERL5LIB=$PERL5LIB:/home/xiaodeng/GTN_2.0' >> ~/.bashrc

Then:

source ~/.bashrc

This action can solve the following error:

"Can`t locate iGraph.pm in @INC (you may need to install the iGraph module)"

Before running, you need to install some tools that required by GTN:

✓ Perl:

https://www.perl.org/

✓ COG database (put this file into your GTN_2.0 folder):

ftp://ftp.ncbi.nih.gov/pub/COG/COG/myva

The following tools must be set to you environment path:

✓ BLAST (if you need to run blast+mcl clustering mode):

ftp://ftp.ncbi.nlm.nih.gov/blast/executables/legacy.NOTSUPPORTED

✓ MCL (if you need to run blast+mcl clustering mode):

https://micans.org/mcl/

✓ mcxdeblast (if you need to run blast+mcl clustering mode):

It is in the mcl package, usually you can found it in the /PATH TO THE MCL/src/alien/oxygen/src/

✓ cdhit (if you need to run cdhit+diamond clustering mode):

http://www.bioinformatics.org/cd-hit/

✓ diamond (if you need to run cdhit+diamond clustering mode):

http://ab.inf.uni-tuebingen.de/software/diamond/

✓ MUMmer (if you need to run gtn_WithDraft.pl):

https://github.com/mummer4/mummer/

✓ R:

https://www.r-project.org/

Note that:

The R package "ape" is required for bootstrap testing, types:

install.packages('ape')

in your R command interface to install it.

The MEGA(http://www.megasoftware.net/) is needed for drawing phylogenetic tree.

Input files:

File format

For each genome that you want to analyze, you need its fna (Nucleic Acid file) format file, faa (FASTA Amino Acid) format file and gff3 (General Feature Format) file.

The format of sequence title in fna and faa files should be like this:

faa file:

```
>AHN29682.1 chromosomal replication initiator protein DnaA [Streptococcus agalactiae 138P]
MTENEQLFWNRVLELSRSQIAPAAYEFFVLEARLLKIEHQTAVITLDNIEMKKLFWEQNLGPVILTAGFEIFNAEITANY
VSNDLHLQETSFSNYQQSSNEVNTLPIRKIDSNLKEKYTFANFVQGDENRWAVSASIAVADSPGTTYNPLFIWGGPGLGK
THLLNAIGNQVLRDNPNARVLYITAENFINEFVSHIRLDSMEELKEKFRNLDLLLIDDIQSLAKKTLGGTQEEFFNTFNA
LHTNDKQIVLTSDRNPNQLNDLEERLVTRFSWGLPVNITPPDFETRVAILTNKIQEYPYDFPQDTIEYLAGEFDSNVREL
EGALKNISLVADFKHAKTITVDIAAEAIRARKNDGPIVTVIPIEEIQIQVGKFYGVTVKEIKATKRTQDIVLARQVAMYL
AREMTDNSLPKIGKEFGGRDHSTVLHAYNKIKNMVAQDDNLRIEIETIKNKIR
>AHN29683.1 DNA polymerase III subunit beta [Streptococcus agalactiae 138P]
MIHFSINKNFFLHALTVTKRAISHKNAIPILSTVKIEVTRDAIILTGANGQISIENTIPASNENAGLLVTNPGSILLEAG
FFINIISSLPDVTLEFTEIEQHQIVLTSGKSEITLKGKDVDQYPRLQEMTTDTPLTLETKLLKSIINETAFAASQQESRP
ILTGVHLVISQNKYFKAVATDSHRMSQRTFQLEKSANNFDLVVPSKSLREFSAVFTDDIETVEVFFSDSQMLFRSENISF
YTRLLEGNYPDTDRLLTNQFETEIIFNTNALRHAMERAYLISNATQNGTVRLEIQNETVSAHVNSPEVGKVNEELDTVSL
KGDSLNISFNPTYLIESLKAVKSETVTIRFISPVRPFTLTPGEDTEDFIQLITPVRTN
>AHN29684.1 hypothetical protein V193_00020 [Streptococcus agalactiae 138P]
MYQVGSLVEMKKPHACVIKETGKKANQWKVLRVGADIKIQCTNCQHVIMMSRYDFERKLKKVLQP
>AHN29685.1 GTP-binding protein [Streptococcus agalactiae 138P]
MALTAGIVGLPNVGKSTLFNAITKAGAEAANYPFATIDPNVGMVEVPDERLQKLTELITPKKTVPTTFEFTDIAGIVKGA
SKGEGLGNKFLANIREVDAIVHVVRAFDDENVMREQGREDAFVDPIADIDTINLELILADLESINKRYARVEKMARTQKD
KESVAEFNVLQKIKPVLEDGKSARTIEFTEEEAKVVKGLFLLTTKPVLYVANVDEDKVADPDDIDYVNQIRAFAETESAE
VVVISARAEEEISELDDEDKLEFLEAIGLTESGVDKLTRAAYHLLGLGTYFTAGEKEVRAWTFKRGIKAPQAAGIIHSDF
ERGFIRAVTMSYDDLIQYGSEKAVKEAGRLREEGKEYIVQDGDIMEFRFNV
```

fna file:

```
>CP007482.1 Streptococcus agalactiae 138P, complete genome
GGGTGTTGATTATTTTTTTATTTAATCAAACTTATCCACAAGGTATTTTGCTATTTTTCAGTTGATTCTCTAAGCTTTTC
TAATTTTCACAGTCTGTGGAAAACTTTAATTAACATTGTTGATTTTATTCTTCAACATCTGTGGAAAACTTTATTTTTTT
ATGGTACAATATAACAATAATTATCCACAAGACAATAAGGAAGAAGCTATGACGGAAAACGAACAACTATTTTGGAATAG
AGTACTAGAGCTATCTCGTTCTCAAATAGCACCAGCAGCTTATGAATTTTTTGTTCTAGAGGCTAGACTCCTCAAAATTG
AACATCAAACTGCAGTTATTACTTTAGATAACATTGAAATGAAAAAGCTATTCTGGGAACAAAATTTGGGGCCTGTTATC
CTAACAGCTGGTTTTGAAATTTTCAATGCTGAAATTACAGCTAACTATGTCTCAAACGATTTACATTTACAAGAAACTAG
```

The id of sequence should be stand between “>” and the first blank.

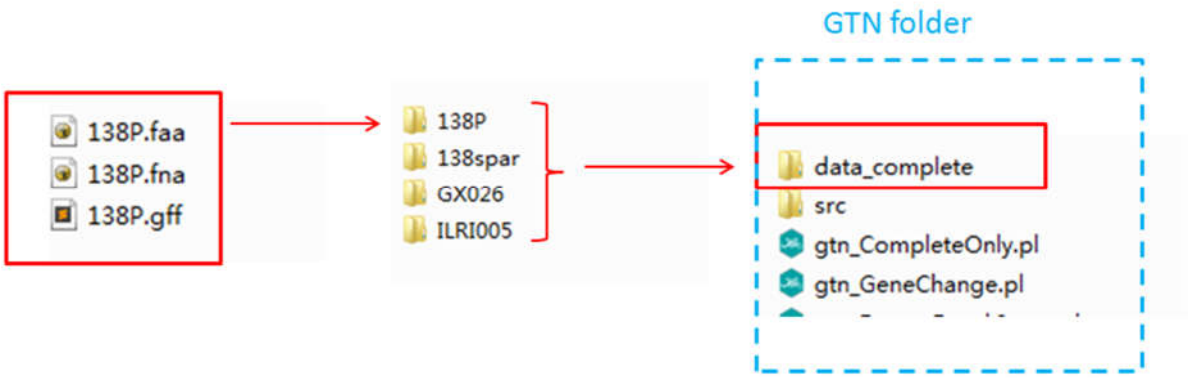
Copy data to GTN_2.0

1) Create a folder for each genome to store its fna, faa and gff3 files, the name of folder will be used as the name of this genome in GTN analysis.

2) Then put all complete genome folders into folder “data_complete” and all draft genome folders into folder “data_draft”.

3) Put “data_complete” and “data_draft” (you should at least prepare one of them) to the GTN_2.0 folder.

An example of complete gene data preparation is shown below:



If there is draft data you need to process, put the files into ‘data_draft’:

GTN usage

GTN now supports two way to perform gene family cluster assignment: BLAST + MCL or CDHIT + DIAMOND.

Our practise suggested that BLAST + MCL method will give higher resolution phylogenetics tree than CDHIT + DIAMOND. However, BLAST + MCL method will cost much more time than CDHIT + DIAMOND.

Build a phylogenetics tree

✧ Run_gtn_CompleteOnly.pl

First you need to change your working directory to GTN:

cd /PATH/TO/GTN/

If you only have complete genomes, you can run first step of GTN like this:

perl gtn_CompleteOnly.pl <thread number> <cluster method> <outgroup>

Where *<thread number>* is the thread number GTN will use to perform alignment; *<cluster method>* is the method you want to use, it should be either '*blast*' or '*cdhit*'; *<outgroup>* is the outgroup name, if you don`t have it, blank it.

An example command is shown below:

perl gtn_CompleteOnly.pl 4 blast ILRI005

It means that I want to use 4 thread number in BLAST + MCL method to run GTN, and the outgroup genome is ILRI005.

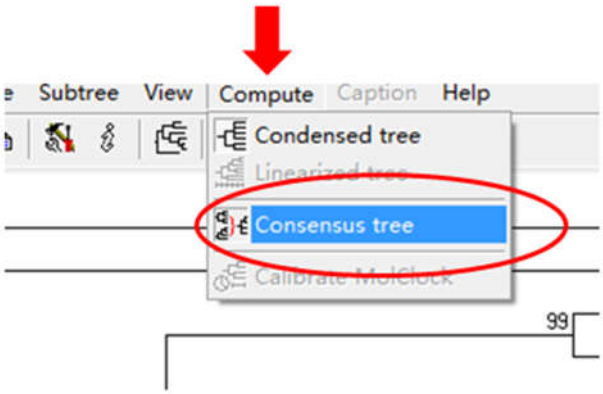
✧ Run gtn_WithDraft.pl

If you have draft genomes to analyse, put folder 'data_draft' to GTN folder(folder 'data_complete' also needed to put into GTN folder if you hanve complete genome data), then run:

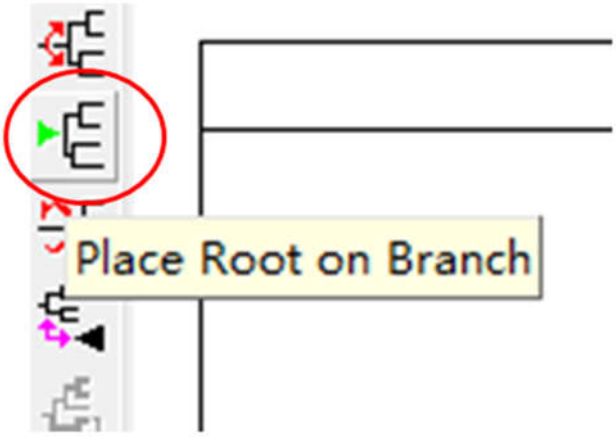
perl gtn_WithDraft.pl <thread number> <cluster method> <outgroup>

You will receive two files named "bootstrap.nwk" and "distance.meg" in the folder GTN. File "bootstrap.nwk" is the bootstrap test result and file "distance.meg" is the original distancs matrix of your genomes.

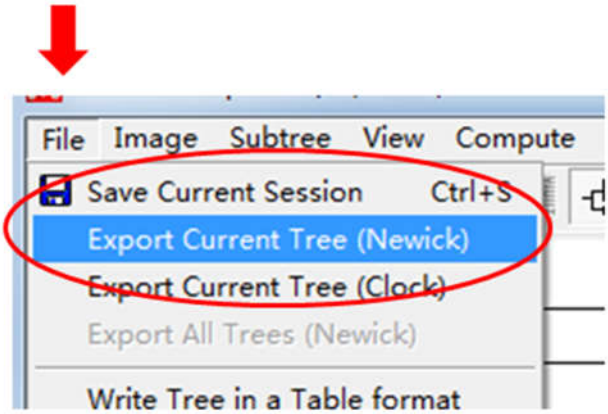
Open the "bootstrap.nwk" with MEGA, then choose 'compute' → 'consensus' the phylogenetics tree:



Somewhile, you need to indicate the outgroup clade by using the ‘Place Root on Branch’ button:



Then choose ‘File’ → ‘Export Current Tree (Newick.)’ to save the result as a newick format file (Let`s assume the name of your saved file is "your.nwk").



Choose "Image"→"Save as PNG (or PDF) file" to save your phylogenetic tree.

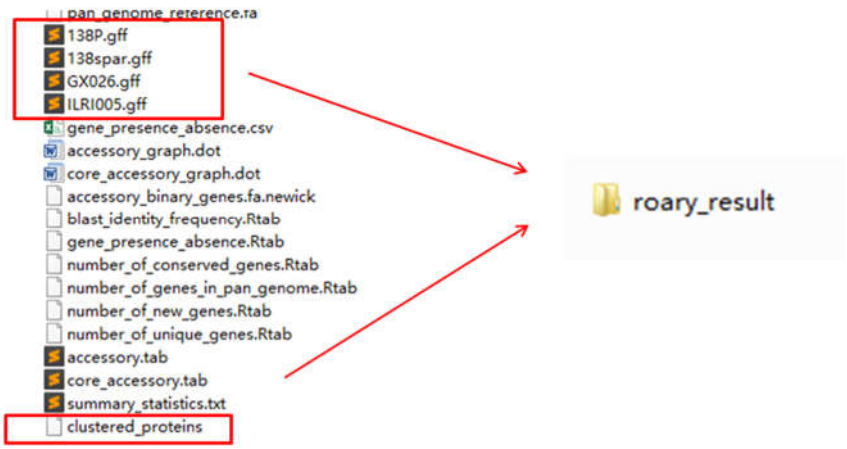
✧ Using ROARY result

GTN can take ROARY result file as cluster result. You can download the test data 'test_data_for_ROARY' in our website for running ROARY by following their manual (http://sanger-pathogens.github.io/Roary/):

*roary -e --mafft -p 4 *.gff*

Make sure the file 'clustered_proteins' is yielded by ROARY.

Copy the GFF files and file 'clustered_proteins' to a new folder (such as 'roary_result')



Change your working directory to GTN and run:

perl GTN_Roary_ResultInput.pl <roary_result_folder_path> <1:complete data | 2: draft data> <outgroup>

Where *<roary_result_folder_path>* is the path to your folders which stores the gff files and file 'clustered_proteins' (such as 'roary_result'); *<1 complete|2 draft>* if you only have complete genome,types '1', else types '2'; *<outgroup>* is the outgroup name, if you don`t have it, blank it.

For example:

perl GTN_Roary_ResultInput.pl /home/dx/test/roary_result 1 ILRI005

After this, you also can get "bootstrap.nwk" and "distance.meg". However, COG function cannot be annotated by using this script.

Get gene order information

perl gtn_GeneChange.pl <nwk file> <1 complete|2 draft> <outgroup>

where *<nwk file>* is the newick format file you saved from MEGA (your.nwk); *<1 complete|2 draft>* if you only have complete genome,types '1', else types '2'; *<outgroup>* is the outgroup name, if you don`t have it, blank it.

An example command is shown below:

perl gtn_GeneChange.pl your.nwk 1 S.pyogenes

It means that my input file is your.nwk; complete genomes is the only data set; outgroup genome is S.pyogenes.

Result files

In GTN, different gene order differentiates the clades in phylogenetics tree, the basic unit of different gene order is 'gene to its adjacent gene' which called 'gene connection' in GTN, such as 'COG0001-COG0010'. So, the results of gene order information are mainly associate with unique gene connections.

These files will be yielded when the calculation is finished:

genes_in_unique_connection.txt: detail result of unique connections in all clades

query_clade	The clade that you plan to query
reference_clade	The clade that you plan to compare to
connection	Gene connection detail
status	'unique' is unique in query_clade, 'miss' is unique in reference_clade

genome	The genome which this row exhibits
COG	First COG in connection
COG_function	Function annotation of First COG
Gene	Gene name of First COG
COG	Second COG in connection
COG_function	Function annotation of second COG
Gene	Gene name of second COG

relative_dd.txt: relative DD value of COGs.

COG	COG id
COG_function	COG function annotation
DD	DD value
str	Genome number
Para/str	Average genes in each genome
Relative_dd	Relative DD value

connection_cog_gene.stat: summary of results.

query_clade	The clade that you plan to query
reference_clade	The clade that you plan to compare to
unique_connection_number	Unique connections number in query_clade
unique_gene_number	Genes number in unique connections in query_clade
miss_cog_number	Unique connections number in reference_clade
miss_gene_number	Gene number in unique connections in reference_clade
delete_cog_number	COG families number which are unique in reference_clade
delete_gene_number	Genes number in COG families which are unique in reference_clade
insert_cog_number	COG families number which are unique in query_clade
insert_gene_number	Genes number in COG families which are unique in query_clade

unique_connection.cog.list: COG names in unique connection.

unique_connection.gene.list: gene names in unique connection.

miss_connection.cog.list: COG names in unique connection in the sister clades of this clade.

miss_connection.gene.list: gene names in unique connection in the sister clades of this clade.

dele.cog.list: COGs which are unique in the sister clade of this clade.

dele.gene.list: genes which are unique in the sister clade of this clade.

insert.cog.list: COGs which are unique in this clade compare to its sister clade.

inser.gene.list: genes which are unique in this clade compare to its sister clade.

fragment_connection.info: It shows the average legnth (KB) and average number of the fragments which are connected based on the genes in gene connection that exit in each genome (common ancestor) of this clade.

clade	The clade that this line describes
average_number	Average number of fragments in each genome
average_length	Average length of fragments in each genome (KB)