# MetaCHIP User Manual

V1.0.1



Centre for Marine Bio-Innovation, University of New South Wales

January 1st, 2019

songwz03@gmail.com

## Introduction

MetaCHIP is a community-level HGT identification pipeline, it is implemented in Python and makes use of a list of 3rd party dependencies and R packages. Details of these dependencies can be found at https://github.com/songweizhi/MetaCHIP. MetaCHIP can be installed with `pip` after its dependencies were installed on your system. Python libraries required by MetaCHIP will be installed automatically during the pip installation. You can either add its dependencies to your system path or specify full path to their executables in MetaCHIP_config.py, which is in folder lib/site-packages/MetaCHIP.

```
$ pip install MetaCHIP
```

The input files for MetaCHIP include a folder that holds the sequence file (in FASTA format) of all query genomes, as well as a text file, which holds taxonomic classification of all input genomes. Please make sure the length of sequence IDs for all input genomes is **NO LONGER THAN 22 letters**.

The MetaCHIP pipeline contains three main modules, which are PI, BM and PG.

```
$ MetaCHIP -h

             ...::: MetaCHIP :::...

    HGT detection modules:
       PI        ->    Prepare Input files
       BM        ->    Best-Match approach
       PG        ->    PhyloGenetic approach

    # for command specific help
    MetaCHIP <command> -h
```

## PI module

```
MetaCHIP PI -h
-i               input genome folder
-taxon           taxonomic classification
-p               output prefix
-r               grouping rank
-g               grouping file
-x               file extension
-grouping_only   run grouping only, deactivate Prodigal and Blastn
-nonmeta         annotate Non-metagenome-assembled genomes (Non-MAGs)
-noblast         not run all-vs-all blastn
-t               number of threads
-qsub            run blastn with job scripts, only for HPC users
-quiet           not report progress
```

PI module will group input genomes at defined taxonomic rank according to their taxonomic classification results. GTDBTk (https://github.com/Ecogenomics/GTDBTk) is recommended for taxonomic classification of input genomes. An example of the taxonomic classification file is provided together with the scripts. Options for "-r" include: d (domain), p (phylum), c (class), o (order), f (family) and g (genus).

**Example command:**

```
# grouping input genomes at provided levels according to taxonomic classifications
$ MetaCHIP PI -i soil_bins -x fa -taxon GTDB_op.tsv -r c -p Soil -t 6
$ MetaCHIP PI -i soil_bins -x fa -taxon GTDB_op.tsv -r o -p Soil -t 6 -grouping_only
$ MetaCHIP PI -i soil_bins -x fa -taxon GTDB_op.tsv -r f -p Soil -t 6 -grouping_only

# run with customized grouping profile
$ MetaCHIP PI -i soil_bins -x fa -g customized_grouping.txt -p Soil -t 6
```

**Output files:**

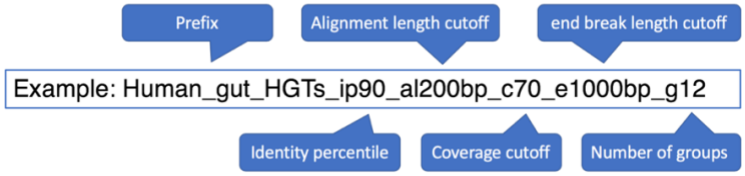
1. Grouping result is exported to **[prefix]_grouping_[taxon_rank][group_num].txt**.
2. Gene calling results in GenBank and FASTA format.
3. A SCG protein tree of input genomes.
4. A bar plot shows the number of input genomes in each group at provided taxonomic rank.
5. Blast results

## BM module

```
MetaCHIP BM -h

-p          output prefix
-r          grouping rank
-g          grouping file
-cov        coverage cutoff, default: 75
-al         alignment length cutoff, default: 200
-flk        the length of flanking sequences to plot (Kbp), default: 10
-ip         identity percentile cutoff, default: 90
-ei         end match identity cutoff, default: 95
-plot_iden  plot identity distribution
-NoEbCheck  disable contig end match and full length match checking for
            fast processing, not recommend for metagenome-assembled genomes.
-t          number of threads, default: 1
-quiet      Do not report progress
-tmp        keep temporary files
```

HGT candidates predicted by the best-match approach, as well as the plots of their flanking regions are exported to a folder named in the following format:




**Example command:**

```
$ MetaCHIP BM -p Soil -r c -t 6

# run with customized grouping profile
$ MetaCHIP BM -p Soil -g customized_grouping.txt -t 6
```

**Output files:**

A list of HGT candidates identified by the BM approach are exported to **HGT_candidates_BM.txt**. Their nucleotide and amino acid sequences are exported to **HGT_candidates_BM_nc.fasta** and **HGT_candidates_BM_aa.fasta**.

## PG module

```
MetaCHIP PG -h

-p          output prefix
-r          grouping rank
-g          grouping file
-cov        coverage cutoff, default: 75
-al         alignment length cutoff, default: 200
-flk        the length of flanking sequences to plot (Kbp), default: 10
-ip         identity percentile, default: 90
-ei         end match identity cutoff, default: 95
-t          number of threads, default: 1
-quiet      Do not report progress
```

**Example command:**

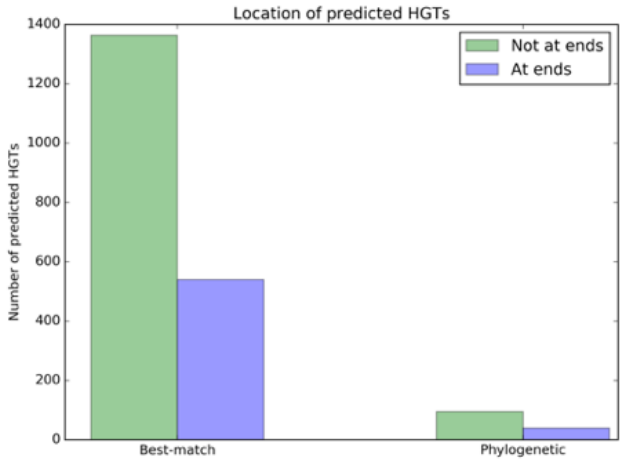
```
$ MetaCHIP PG -p NorthSea -r c -t 6

# run with customized grouping profile
$ MetaCHIP PG -p NorthSea -g customized_grouping.txt -t 6
```

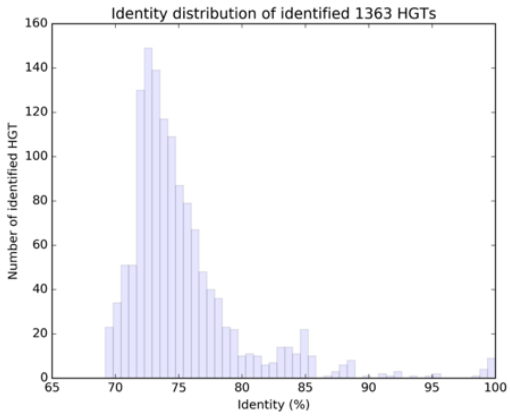
**Output files:**

HGT candidates validated by PG approach are exported to the same folder as BM approach.

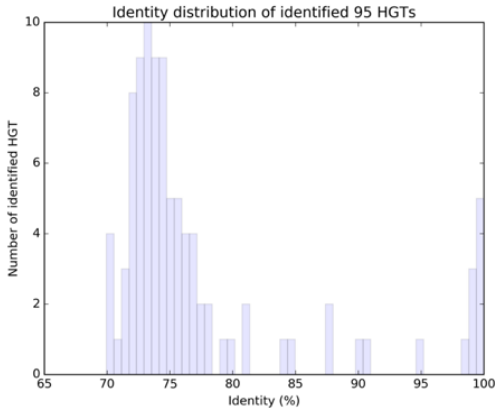
1. **HGT_candidates_PG.txt**: BM approach predicted HGTs, with additional information provided by the PG approach.
2. **HGT_candidates_PG_validated.txt**: HGTs that are only validated by the PG approach.
3. **HGT_candidates_PG_aa.fasta**: Nucleotide sequences of HGTs that are validated by the PG approach.
4. **HGT_candidates_PG_nc.fasta**: Amino acid sequences of HGTs that are validated by the PG approach.
5. **[prefix]_plot_at_ends_stat.png**: Location statistics of predicted HGTs by BM and PG approaches.



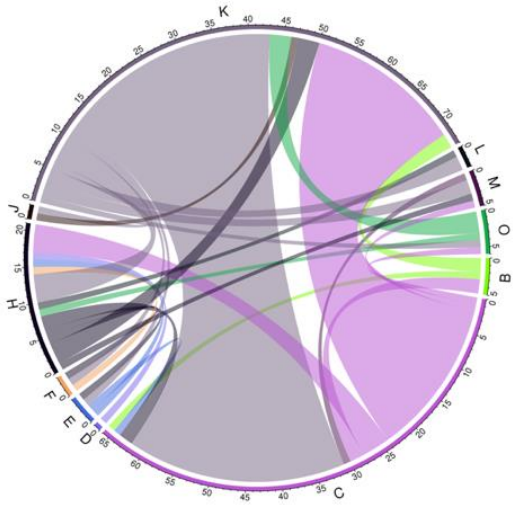

6. **[prefix]_plot_HGT_identities_BM.png**: Identity distribution of BM approach predicted HGTs.




7. **[prefix]_plot_HGT_identities_PG.png**: Identity distribution of predicted HGTs that are validated by the PG approach.




8. **[prefix]_plot_circos_PG.png**: Gene flow between groups. Bands on the plot connect donors and recipients, with the width of the band correlating to the number of HGTs and the colour corresponding to the donors.



## References

1. Parks DH, Imelfort M, Skennerton CT, Hugenholtz P, Tyson GW: CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome research 2015, 25:1043-1055.