HoMer (version 1.0) Manual

Overview

HoMer (short for "Horizontal Multi-gene transfer inference") is a software package for inferring instances of *horizontal multi-gene transfer (HMGT)* during the evolutionary history of a collection of microbial species/strains. An HMGT occurs when multiple genes are horizontally transferred in single horizontal transfer event. HoMer takes as input a rooted species tree, gene ordering information for the species/genomes (leaves) represented in the species tree, and rooted gene trees for all gene families (with at least three genes each) present in the species/genomes under consideration. The software outputs a list of inferred HMGTs for each donor-recipient pair on the species tree, where donors/recipients can be leaves (i.e., given genomes) or internal (i.e., ancestral) edges on the species tree.

HoMer is written in Python and requires version 3.0 or greater. The implementation also assumes that ETE 3 toolkit is already installed. ETE toolkit is available freely from etetoolkit.org. Technical details about the algorithm implemented in HoMer appear in the paper cited below.

HoMer was implemented by Lina Kloub and is available open source under GNU GPL.

This software package can be cited as follows:

> *Systematic Detection of Large-Scale Multi-Gene Horizontal Transfer in Prokaryotes*
> L. Kloub, S. Gosselin, M. Fullmer, J. Graf, J. P. Gogarten, M. S. Bansal
> Under review.

Software description

The HoMer software package consists of two Python programs: *HoMer.py*, which implements the core HMGT inference approach, and *Homer_Aggregate.py,* which computes phylogenetic reconciliations and performs other preprocessing and creates the input files used by *HoMer.py*. The program *Homer_Aggregate.py* also makes use of two precompiled C/C++ executables to compute and summarize phylogenetic reconciliations for high-confidence horizontal gene transfer (HGT) inference. These precompiled C/C++ executables are already included in the Linux and macOS versions of HoMer.

Next, we describe how to use *Homer_Aggregate.py* and *HoMer.py.*

Using *Homer_Aggregate.py*

HoMer_Aggregate.py takes as input a set of gene trees and a rooted species tree, and reconciles the gene trees with the species tree (100 reconciliations per gene tree to account for

reconciliation uncertainty) and produces as its output the "reconciliation results" and "labeled species tree" that are required as input for *HoMer.py*. It has three required parameters and one optional parameter, as given below.

```
python Homer_Aggregate.py -g <path to the directory containing rooted gene trees> -s <rooted species tree> -f <path to an empty directory to save output files> [-t <transfer cost to use for reconciliation>]
```

Available command line options:

-g Path to directory containing the rooted gene trees in Newick format. This is a required parameter.

-s Rooted species tree. This is a required parameter.

-f Path to an empty directory to save output files. This is a required parameter.

-t Transfer cost to use when computing phylogenetic reconciliations. Defaults to 4. Must be a natural number.

Format of input files:

All input trees must be rooted, binary, and in Newick format. Species names in the species tree must be unique and composed of only alphanumeric characters.

E.g., ((speciesA, speciesB), speciesC);

Each leaf in the gene tree must be labeled with the name of the species from which that gene was sampled. The gene name/ID can be appended to the species name separated by an underscore '_' character. The gene tree may contain any number (zero, one, or more) of homologous genes from a species.

E.g., (((speciesA_gene1, speciesC_gene1), speciesB_geneX), speciesC_gene2);

Important notes on usage:

1. Each gene tree must be in a separate file, and the first part of the gene tree file name (before the first ".") must be the "gene family name/ID" that the gene tree corresponds to.
2. The species tree must be not have any internal node labels or bootstrap/support values.
3. The directory provided through the -f option should already exist (i.e., should be created before executing *Homer_Aggregate.py*).
4. We recommend leaving the transfer cost at its default value of 4. A lower cost of 3 may also be appropriate and will result in more HGTs and HMGTs being inferred. We do not recommend using a transfer cost lower than 3.

Test data and test run:

Homer_Aggregate.py is available for macOS and Linux operating systems. In the "HoMer_Aggregate" folder inside the HoMer software package, there is a directory named "testGeneTrees" containing four gene trees and a file named "speciesTree.newick" containing a rooted species tree. You can execute Homer_Aggregate.py on this test data as follows:

```
mkdir testResults
```

```
python Homer_Aggregate.py -g testGeneTrees -s speciesTree.newick -f
testResults
```

Interpreting *Homer_Aggregate.py* output

Homer_Aggregate.py writes all its output to the directory specified using the -f option. This output consists of four components.

1) "RangerInput" directory: This directory contains files that *Homer_Aggregate.py* feeds as input to the RANGER-DTL reconciliation program for each gene tree. This directory and its contents can be safely ignored and the directory can be deleted after *Homer_Aggregate.py* finishes executing.
2) "RangerOutput" directory: This directory contains the raw reconciliation output generated by RANGER-DTL for each gene tree. This directory and its contents can be safely ignored and the directory can be deleted after *Homer_Aggregate.py* finishes executing. In cases where computing these reconciliations takes a long time, it may be worth saving this directory so that subsequent HMGT analysis can be repeated without having to recompute all reconciliations.
3) "ReconciliationResults" directory: This directory contains the aggregated reconciliations for each input gene tree. This directory must be provided as input to *HoMer.py* to perform HMGT inference. Note that the files in this directory have the same names as the original input gene tree files, but their content is different.
4) "LabeledSpeciesTree.newick" file: This file contains a labeled version of the input species tree, where each internal node of the species tree is given a specific label. This labeled species tree must be provided as input to *HoMer.py* to perform HMGT inference.

Using *HoMer.py*

HoMer.py should be executed after first executing *Homer_Aggregate.py* to create the necessary input files. *HoMer.py* takes as input the "ReconciliationResults" directory output by *Homer_Aggregate.py*, the "LabeledSpeciesTree.newick" file output by *Homer_Agrregate.py*, and the gene ordering (synteny) files for each species (leaf) present in the species tree. It has three required parameters and several optional parameters, as described below.

```
python HoMer.py -g <path to the directory containing gene tree
reconciliation results> -s <labeled species tree> -n <path to the
directory containing gene ordering (synteny) files> [-other options]
```

A description of available command line options follows:

-g	Path to the directory containing gene tree reconciliation results. This is a required parameter.
-s	Rooted species tree. This is a required parameter.
-n	Path to the directory containing gene ordering (synteny) files. This is a required parameter.
-a 0|1	0 for leaf-to-leaf HMGTs. 1 for other (non-leaf-to-leaf) HMGTs. Defaults to 0.
-t	Confidence threshold for inferring transfer events. Must be between 1 and 100. Defaults to 100.
-m	Mapping confidence threshold for transfers. Must be between 51 and 100. Defaults to 51.
-e	File extension for the genome ordering files. Defaults to ".synteny".
-k	File containing list of rare genes. Default is to assume no rare genes.
-x	HMGT inference parameter. Specifies minimum number of HGTs required in each window. Defaults to 3.
-y	HMGT inference parameter. Specifies window size for HMGT inference. Defaults to 4.
-z	HMGT inference parameter. Specifies allowable gap size for HMGT extension. Defaults to 1.
-f 1	Perform randomization analysis for transfers. Used to estimate false discovery rate for HMGTs.
-h	Output this help message.

Format of gene ordering/synteny files:

The directory containing the gene ordering (synteny) files must contain exactly one file for each species (leaf) represented in the species tree. The first part of each file's name (before the first ".") must exactly match the corresponding species name used in the species tree. By default, each such file is assumed to end with a ".synteny" file name extension. Please note that the

default file name extension for these file is set to ".synteny". If your synteny files have different extension, you can sure pass it along using the optional parameter -e.

Each synteny file can contain multiple contigs, with one contig per line (i.e., different contigs must appear on different lines). Each contig (line) consists of a tab-separated ordering of genes in the following format:

SpeciesName:GeneName:_contig_ContigNumber:GeneFamilyID

For example,
AfluvialisLMG24681T:123:_contig_11:11986 AfluvialisLMG24681T:124:_contig_11:15157 AfluvialisLMG24681T:125:_contig_11:10561

The above example shows an ordering of three tab-separated genes from contig 11 in species AfluvialisLMG24681T. The gene names/IDs of these three genes are 123, 124, and 125, and their gene family IDs are 11986, 15157, and 10561, respectively.

Note that the species name used in these gene orderings should match the species name used in the species tree. Gene names and gene family IDs can be composed of alphanumeric characters. Also note that the gene family IDs used in these synteny files should match the gene family IDs used to name the files in the "ReconciliationFiles" directory. The contig number should always be a positive integer.

Format of optional "rare genes" file:

HoMer allows for the option of skipping over rare genes in the genome orderings (synteny files) when inferring HMGTs. Rare genes are defined to be genes from those gene families that have only a total of one or two genes in all of the species in the analysis. Such rare genes are assumed to have been acquired through HGT (likely from species not represented in the species tree). To invoke this option, a file containing a list of rare gene families can be provided using the -k option. This file should contain a list of gene families along with the species in which that gene family is found occur, one per line, n the following format: Gene family ID>Species name.

For example:
1183>SpeciesA
1183>SpeciesD
35232>SpeciesA
54364>SpeciesB

Important notes on usage:

1) By default, HoMer.py only computes HMGTs between leaf-species on the species tree (i.e, both the donor species and recipient species are extant taxa). HMGTs where either the donor or recipient, or both, are ancestral species (internal edges) can be computed by invoking the "-a 1" option.

2) Output is written directly to the screen and must be redirected to an output file in order to save it.
3) The -t and -m option allow users to be more or less permissive in defining which HGTs can be used for HMGT inference. The -t option specifies the confidence threshold for HGT events and its default value is set to the maximum possible value of 100 percent support. We recommend using this default value for HMGT analysis. The -m option specifies the confidence in donor and recipient species (edge) assignment for inferred HGTs. The default value is set to 51 percent support. In some cases, a higher threshold, say 75, can be used to be more stringent in HMGT analysis.
4) HoMer allows for the customization of the size of inferred HMGTs and the stringency of their inference. HoMer can also be used to estimate the false discovery rate (FDR) for inferred HMGTs. We discuss these features in detail later in this manual.

Test data and test run:

In the "HoMer" folder inside the HoMer software package, there is a directory named "ReconciliationResults", generated using *Homer_Aggregate.py* and containing aggregated reconciliation files for each of the 8277 gene trees from the *Aeromonas* dataset used in our analysis, a directory named "Genomes" containing the synteny files for 103 *Aeromonas* genomes, and a file named "LabeledSpeciesTree.newick" containing the labeled species tree on the 103 Aeromonads (generated using *Homer_Aggregate.py*). The folder also contains a rare genes file named "rareGenesList.txt". *HoMer.py* can be executed on this dataset as follows:

```
python HoMer.py -g ReconciliationResults -s LabeledSpeciesTree.newick
-n Genomes -k rareGenesList.txt >> HMGTresults.txt
```

The above command will write the output to the file "HMGTresults.txt". Note that analyzing this dataset requires about 10 minutes of runtime.

To infer HMGTs between ancestral species (non-leaf-to-leaf HMGTs), one would instead execute:

```
python HoMer.py -g ReconciliationResults -s LabeledSpeciesTree.newick
-n Genomes -a 1 -k rareGenesList.txt >> HMGTresults_internal.txt
```

Interpreting *HoMer.py* output

The output of *HoMer.py* consists of a list of all donor-recipient pairs (more precisely, ordered pairs) that have at least one detected HMGT. For each such donor-recipient pair, a list of the identified HMGTs is output, along with some auxiliary and summary information including (i) the number of nodes between the donor and recipient in the species tree, (ii) number of gene transfers detected within each HMGT, (iii) the total number of HMGTs found for that donor-recipient pair, (iv) total number of gene transfers detected within all HMGTs found for that

donor-recipient pair, and (v) the total number of gene transfers found for that donor-recipient pair. After all donor-recipient pairs have been listed, a summary block is output with information on the total number of donor-recipient pairs listed, total number of inferred HMGTs, total number of detected HGTs within all HMGTs, and total number of detected HGTs for all listed donor-recipient pairs.

For example, the output may look like the following:

Donor: AspAMC34, Recipient: AveroniiBAQ135
Number of nodes between AspAMC34 and AveroniiBAQ135: 11.0

Contig: 1, Gene Number: 3350, Gene Name: 2758, Gene Family: 11131 -----> Contig: 7, Gene Number: 143, Gene Name: 3792, Gene Family: 11131
Contig: 1, Gene Number: 3351, Gene Name: 2759, Gene Family: 21113 -----> Contig: 7, Gene Number: 142, Gene Name: 3791, Gene Family: 21113
Contig: 1, Gene Number: 3353, Gene Name: 2761, Gene Family: 9814 -----> Contig: 7, Gene Number: 141, Gene Name: 3790, Gene Family: 9814

Number of transfers in HMGT 1 = 3

Total number of HMGTs = 1
Total number of genes in HMGTs for this pair = 3
Total number of HGTs for this pair = 28

Donor: AspAMC34, Recipient: AveroniiAK227
Number of nodes between AspAMC34 and AveroniiAK227: 10.0

Contig: 1, Gene Number: 249, Gene Name: 3393, Gene Family: 13583 -----> Contig: 15, Gene Number: 40, Gene Name: 726, Gene Family: 13583
Contig: 1, Gene Number: 250, Gene Name: 3394, Gene Family: 10429 -----> Contig: 15, Gene Number: 41, Gene Name: 727, Gene Family: 10429
Contig: 1, Gene Number: 251, Gene Name: 3395, Gene Family: 1064 -----> Contig: 15, Gene Number: 42, Gene Name: 728, Gene Family: 1064

Contig: 1, Gene Number: 1223, Gene Name: 370, Gene Family: 13155 -----> Contig: 7, Gene Number: 41, Gene Name: 3452, Gene Family: 13155
Contig: 1, Gene Number: 1224, Gene Name: 371, Gene Family: 12131 -----> Contig: 7, Gene Number: 40, Gene Name: 3451, Gene Family: 12131
Contig: 1, Gene Number: 1226, Gene Name: 373, Gene Family: 5694 -----> Contig: 7, Gene Number: 39, Gene Name: 3450, Gene Family: 5694
Contig: 1, Gene Number: 1227, Gene Name: 374, Gene Family: 6001 -----> Contig: 7, Gene Number: 40, Gene Name: 3449, Gene Family: 6001

Number of transfers in HMGT 1 = 3
Number of transfers in HMGT 2 = 4

Total number of HMGTs = 2
Total number of genes in HMGTs for this pair = 7
Total number of HGTs for this pair = 50

****

Summary results:
Total number of donor-recipient pairs: 2

Total number of HMGTs: 3
Total number of HMGT-genes: 10
Total number of HGTs for these donor-recipient pairs: 78

In the above sample output, two donor-recipient pairs are listed, followed by an overall summary. In the first donor-recipient pair, one HMGT comprising of three detected HGTs is found. In the second donor-recipient pair, two HMGTs are found, with the first HMGT comprising of three HGTs and the second HMGT comprising of 4 detected HMGTs.

The detected HGTs within each HMGT are listed in the following format:

```
Donor Contig Number, Donor Gene Number, Donor Gene Name/ID, Gene Family ID ----> Recipient Contig Number, Recipient Gene Number, Recipient Gene Name/ID, Gene Family ID
```

For example:
Contig: 1, Gene Number: 3350, Gene Name: 2758, Gene Family: 11131 -----> Contig: 7, Gene Number: 143, Gene Name: 3792, Gene Family: 11131

Note that the "Gene Number" is assigned internally by HoMer to the genes on each contig in the synteny files. These numbers are assigned sequentially starting with 1, for each contig, and indicate the position of that gene along that contig.

When executed using the "-a 1" option (i.e., for computing non-leaf-to-leaf HMGTs), if the donor in a specific donor-recipient pair is an internal node of the species tree, then the specific leaf-descendant whose synteny file was used to infer the HMGTs for that donor-recipient pair is also output.

Estimating false discovery rate for inferred HMGTs

HoMer can also be used to estimate the false discovery rate (FDR) for inferred HMGTs. When *HoMer.py* is executed with the command line option "-f 1" it performs a randomization analysis to estimate how many HMGTs would be inferred just by chance even if all detected HGTs were in fact single-gene HGTs. This randomization analysis is done by first randomizing the inferred HGTs while preserving total HGT counts as well as their donors and recipients, and then executing the same HMGT inference pipeline using these randomized HGTs instead of inferred HGTs. The resulting output can be used to estimate the overall FDR for all inferred HMGTs as well as FDRs for specific donor-recipient pairs. We suggest repeating this randomization test at least 10 times and averaging over the results.

Customizing HMGT size and stringency of HMGT inference

The most important parameters for customizing HMGT inference are the optional parameters -x, -y, and -z. To infer HMGTs, HoMer, first identifies contiguous regions of *y* genes in which at

least *x* genes were detected as transferred from the donor to the recipient, and then merges the identified regions with neighboring regions or HGTs if the distance between them is no more than *z*. To avoid ambiguity in the merged regions, the value of *z* must never be smaller than *y-x.* By default, the values of *x, y,* and *z,* are set to 3, 4, and 1, respectively. To decrease or increase the size of inferred HMGTs, the *x* and *y* parameters can be made smaller or larger, respectively. To make HMGT inference more stringent (i.e., to tolerate fewer "gaps") *x* can be made equal to *y* and z can be decreased to 0. To make HMGT inference less stringent (i.e., to tolerate more "gaps") *y* can be increased without increasing *x*, and *z* can be correspondingly increased as well.

For example:

```
python HoMer.py -g ReconciliationResults -s LabeledSpeciesTree.newick
-n Genomes -k rareGenesList.txt -x 2 -y 3 -z 1 >> HMGTresults.txt
```

Contact

If you have any questions, suggestions, or concerns, please contact either Lina Kloub (lina.kloub@uconn.edu) or Mukul Bansal (mukul.bansal@uconn.edu).