1. Introduction

1.1 What is HGT?

Horizontal gene transfer (HGT), also known as Lateral Gene Transfer (LGT), is frequently defined as the exchange of genetic material between two species that are not genealogically related. This phenomenon is commonly encountered in prokaryotes through mechanisms such as conjugation, transduction, and transformation. In contrast to vertical gene transfer, which occurs from parent to offspring, HGT transcends the boundaries of kinship, thereby complicating the dynamics of gene flow.

Observations of HGT date back as early as 1959. A series of publications documented the ability of high-frequency transducing (Hfr) strains of Escherichia coli to laterally transfer genetic information to specific mutant strains of Salmonella typhimurium. In the same year, Tomochiro Akiba and Kunitaro Ochiai uncovered resistance plasmids in pathogenic bacteria, ultimately leading to the discovery that these plasmids carrying resistance could be transferred between bacterial strains.

However, the concept of HGT was not yet formulated at that time. It was only in the 1990s, with the advent of genetically engineered organisms (GEOs), particularly genetically engineered microorganisms (GEMs), and the emergence of numerous drug-resistant pathogens whose origins could no longer be solely attributed to genetic mutations, that the concept of horizontal gene transfer gained traction and gradually emerged as a research focus.

The occurrence of similar phenotypes across genetically distant organisms is often attributed to the shared genes present in their genomes. The absence of these genes in closely related lineages can be rationalized by multiple independent instances of gene loss. Alternatively, the sharing of genes could stem from horizontal gene transfer (HGT) across vastly different lineages. In principle, HGT can take place between any two organisms possessing DNA genomes. In contrast to vertical gene transmission, which underpins the conservation of biological heritage and the stability of the eukaryotic life tree, HGT acts as a significant driving force for eukaryotic species diversification, enabling adaptation to diverse environments. While HGT is widely accepted as a major evolutionary force in prokaryotes, its role in eukaryote evolution remains highly contested due to the complex evolutionary histories, intricate genomes, and frequent sequence contaminations.

To address this issue, we investigated bacterial-derived HGTs that have recently been integrated into 23 congeneric species of Chromalveolate, a supergroup renowned for its broad ecological distribution and substantial phenotypic diversity (Fan et al., 2020. Science Advances). Our findings provide compelling evidence of HGT across all examined lineages, with a remarkable diversity in the sources and functions of the transferred genes among recipient lineages. Notably, consistent with their exogenous origins, these HGTs exhibit significant differences in gene structure, codon usage, and expression profiles when compared to the core genes of the chromalveolate hosts. These results emphasize the profound impacts of HGT on the genetic and functional

divergence among chromalveolate lineages.
HGT primarily occurs in two distinct modes: sporadic horizontal gene transfer, where one species actively acquires or passively receives a single DNA fragment from another, and endosymbiosis gene transfer (EGT), a gradual exchange of genetic material between a donor and a host that maintain a stable endosymbiotic relationship.

1.2 Methodology of HGT

On the subject of the sources of these accessory genes, two dominant evolutionary hypothesis, the differential loss hypothesis (DL) and horizontal gene transfer (HGT), have gamble mutually. DL hypothesis favors a "pristine" view of eukaryotic genome evolution with the current sporadic distribution of many HGT-derived gene families explained by DL since formation of the last eukaryotic ancestor. The main claim of this theory is that many genes shared by the eukaryotic common ancestor have been lost in some lineages.

The fixation of transferred genes is highly unpredictable due to difference between foreign genes and host genes in GC content, codon usage, intron presence, transcriptional promoters. This leads to different evolutionary trajectories of different genes that are transferred from the same donor genomes. HGT has its unique role in gene, genome and biological evolution. In the early days, common analysis methods include: evolutionary tree analysis, base composition analysis, selection pressure analysis, intron analysis, specific sequence analysis and nucleotide composition bias analysis. The most outcomes of these methods hold propensity to false positive given their simple and crude nature. However, none of these applications should be arbitrarily excluded.

So far, the golden methodologies for HGT analysis, the phylogeny strategy, followed a basic principle: if a gene in the host nuclear genome can be monosystemically evolved with the group to which the endosymbiont belongs, it is considered to be an HGT. That is, if a nesting relation of two taxonomically distant gene in a tree (taxonomic conflict) need to interpreted by more gene loss events, the more likely it is caused by HGT. Phylogeny strategy with a reasonable estimation of evolutionary model is the key step in HGT identification given its wide application of statists and thorough consideration of evolution making it became the main battleground between the "DL" and "HGT" theories and eventually brought them to a compromise. With the development of technology, a variety of databases and pipeline have been developed to largely automate the identification process of HGT, include Pfam, COG, TIGRfam, RAST, HGTree, GOLD, fusionDB.

However, this analysis strategy has several major limitations. First, the genetic coverage of some groups in the database was prone to false positives. The breadth of the sampling pool is a key factor in phylogeny based HGT analysis. Additionally, recent expansion of sequenced genomic data has enabled the construction of genome wide phylogenies for defining taxonomy. This would be helpful to address an important aspect of the species-sampling effect, the phylogenetic bias in the data sets being analyzed. Secondly, the automatic selection algorithm of evolutionary tree is not mature. The more comprehensive tests are expected for past theories and observations that were seemingly at odds in explaining HGTs. Third, the study of a

single species or gene cannot reveal the general rules of a certain kind of evolutionary event; Finally, there is a lack of cross-sectional comparisons between different species under uniform analytical standards. In order to more accurately deduce the evolutionary history of HGT driven organisms, the shortcomings of HGT analytical methodology need to be improved from the above aspects. A broader background database and standardized, automated, high-throughput analysis platform need to be estblished.

1.3 Common HGT analysis method

Currently, common analysis methods include: evolutionary tree analysis, base composition analysis, selection pressure analysis, intron analysis, specific sequence analysis and nucleotide composition bias analysis.

a. Base composition analysis

The GC content of the genome varies among bacterial species, and the GC content of the genome of each bacterial species is relatively stable and consistent among different genes, and they are not affected by external factors. If the GC content of a particular DNA sequence of a strain is significantly higher or lower than the rest of its genome, it implies that the particular DNA sequence was obtained from an exogenous bacterial or other species' plasmid by horizontal transfer.

Commonly used software: Alien_hunter. It’s suitable for bacteria.

b. Selective pressure analysis

In a sense, the evolution of organisms is the evolution of genes, and genomic DNA is constantly tested by selection pressure during the evolution of organisms. If two distantly related species are highly similar in a particular gene, and the amino acid encoded by the gene has not changed, and the gene is not under selective pressure, then the gene may have been transferred horizontally between the two species. The ratio of the number of nonsynonymous substitutions (dN) to the number of synonymous substitutions (dS) (dN/dS) is used to determine whether a gene is under selection pressure. If the dN/dS value of a gene is significantly greater than the dN/dS value of a conserved gene in the species in which the gene is located, then the gene is not under evolutionary selection pressure.

Commonly used software: yn00. It’s suitable for eukaryotes and also for prokaryotes.

c. Intron analysis

Generally speaking, the degree of variation in intron sequences during evolution is very high because they are not subject to selection pressure or low selection pressure, which adds a new method to determine gene level transfer. If a particular gene from two species with a large genetic evolutionary gap is highly homologous not only in its coding region but also in its non-coding intron region, it is likely that the gene was obtained by horizontal transfer.

It’s suitable for eukaryotes.

d. Evolutionary tree analysis

The most widely used and simplest method to test for HGT is to use Blast similarity search. A high degree of sequence similarity between distantly related species for a specific gene or a segment of a specific gene is generally taken as initial

evidence or suspicion of HGT. The evolutionary relationships of the vast majority of genes between species are consistent with biological classification, and only a few genes that have undergone horizontal transfer have evolutionary relationships that differ significantly from traditional biological classification. Thus, the arrangement of evolutionary branches on the evolutionary tree becomes an important criterion for determining HGT. Some genes are quite conserved in the species, and they can be used to establish the evolutionary relationships of the studied species as a reference standard to determine whether other genes are horizontally transferred. The evolutionary tree constructed with the target gene of horizontal transfer can be compared with the evolutionary tree constructed with conserved genes or traditional taxonomic methods to determine whether, when and where the horizontal transfer of the target gene occurred.

Commonly used software: BLAST and evolutionary tree construction software. It's suitable for both eukaryotic and prokaryotic.

2. HGTstart

We developed HGTstart to facilitate large-scale, high-throughput detection of HGT and explore the breadth of HGT-driven biological evolution across the tree of life (TOL). This platform carries out automated, standardized, high-throughput analysis to predict HGTs *de novo* with protein sequences as input combined with an open-access knowledgebase to browse a pre-computed genome-wide repository of HGT-derived genes, alignments, and complete phylomes. We applied HGTstart to over a thousand species, generating a list of candidate HGT genes and revealing HGT distribution patterns within and between kingdoms. By chronologically ordering HGTs, we inferred their contribution to the complexification and diversification of biological systems, offering research possibilities for validating numerous evolutionary theories.

HGTstart is implemented in Perl and JAVA, with easy installation, and runs in command line on LINUX and Mac OS platforms, and was built on the web server platform https://hgtstart.cn. As a user-friendly and comprehensive web application for browsing, searching, and predicting HGT events, HGTstart is designed to work with genome-wide data and ascertain the extent and impact of HGT on species evolution. With the inclusion of ~1,200 genomes in GNM1157, and the collection of external results from worldwide users, HGTstart fulfills a critical need in evolutionary research.

2.1 Construction of a large background database

To do an effective HGT survey, we included as many high-quality protein sequences as possible from whole genome data collected from a wide variety of prokaryotic and eukaryotic lineages, as well as individual sequences from the main protein databases.

We checked the genome assembly summary information (updated by May 2021) of RefSeq (ftp://ftp.ncbi.nlm.nih.gov/genomes/README_assembly_summary.txt) to download all available proteins from completed genomes. We grouped the genomes of species within the same genus, among which we downloaded only the genome that

represented the least fragmented assembly in this group (and the latest version, if multiple versions exist for this genome). We also searched for genomes from JGI (https://genome.jgi.doe.gov/portal/) and other databases. One of the most important issues that influence HGT inference is uneven sampling caused by unbalanced data collection among taxa. We included proteins from MMETSP in the database to compensate for the lower amount of Rhodophyta genome data. These extensive genome data resulted in a protein database comprising 17,250,679 protein sequences from 1,157 genomes with reasonable coverage in most lineages in the tree of life (Table S1, 540 bacteria, 45 archaea, 431 Opisthokonta, 15 Rhodophyta, 83 Viridiplantae and 43 genomes from CRASH lineages). These 1,157 complete genomes each represent one representative species for their respective genus. The database was named "GNM1157".
In addition to GNM1157, a larger database comprising as many known protein sequences as possible was constructed. We built this background database by downloading the NCBI RefSeq database (version 82, ftp://ftp.ncbi.nlm.nih.gov/refseq/). Sequences associated with unknown species or derived from environmental studies were discarded. Given the underrepresentation of algal lineages in RefSeq, we included the rich algal protein data derived from the MMETSP project and other public sources to extend the taxonomy span of the background database.

The collected protein sequences (GNM1157+Refseq+ MMETSP) were combined into a master database, REFAL, followed by the removal of highly similar sequences (sequence identity ≥90%) from each order (e.g., Brassicales or Primates) using CD-HIT version 4.5.4. This resulted in a protein database comprising 39.9M sequences from > 7786 taxa with reasonable coverage in most lineages in the TOL.

To facilitate the functional prediction for query sequences, protein sequences of GNM1157 were pre-predicted by accessing several large databases such as Interproscan, EGGNOG, PANTHER, Pfam, and SUPERFAMILY. Query sequences were linked to their gene hit in GNM1157 to retrieve the corresponding predicted function.

2.2 Inference of orthologous groups and benchmarking

Homology relationships (orthology and paralogy) between genes are fundamental to comparative biological research, such as HGT analysis. There has been some database development via tree-based PhylomeDB, Ensembl-Compara, EggNOG, and TreeFam; however, we found that the existing databases could not cover the complete repertoire of genes encoded in GNM1157, indicating a need to customize our orthologous database. We combined the clustering of 17,250,679 proteins encoded in GNM1157 with protein-based topology analysis. All the protein sequences were analyzed by an all-against-all BLAST workflow for all pairs of genomes (version 2.2.28; e-value cutoff = 1e-10; local identity cutoff = 20%). The orthogroup graph based on reciprocal best length-normalized hit (RBNH) information from each pairwise

genome combination was calculated using OrthoFinder V2.3.7, which is good at correcting for the dependence of gene similarity on gene length and phylogenetic distance, thus improving the overall accuracy of orthogroup delimitation. We then independently ran an unsupervised Markov clustering algorithm (MCL3.0) to this orthogroup graph rather than using the MCL integrated in OrthoFinder for two reasons. 1) Given the large size and taxon span of GNM1157, the derived orthogroup graph was as large as 500 GB and interrupted the computation process several times due to exceeding the limit of a single node in a computing cluster. 2) To balance the trade-off between recall (a high recall rate means that many sequences could be assigned to an orthogroup) and precision (high precision means the sequences are assigned to the correct orthogroup), we ran MCL multiple times with a gradient set of inflation parameters (1.2, 1.4, 1.6, 1.8, 2.0, 2.2) and selected the best run. To meet the process demand for RAM and running threads, we customized an iMac Pro with 18 cores (Intel Xeon W) at 2.9 GHz each running macOS 10.14.6. Even when using this machine, it took more than 2 weeks for every clustering task with an inflation value. Achieving high accuracy in orthology inference is essential for many comparative, evolutionary and functional genomic analyses. More than ten of the most popular online orthology databases, e.g., EGGNOG, PANTHERN, and SUPERFAMILY were downloaded locally as benchmarks to calibrate our orthology inference parameters. The precision, recall and F0 of the gradient set were calculated using the NumPy library in Python 3.0 to determine the best trade-off for inflation.

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

where TP is the number of true positive orthogroup assignments (that is, correct assignments), FP is the number of false positive orthogroup assignments (that is, incorrect assignments) and FN is the number of false negative orthologue assignments (that is, missing assignments). The F-score is the harmonic mean of these two measures, where the harmonic mean weights towards the worst-performing measure.

To identify biologically meaningful orthologous clusters, we tested the performance of a group-specific conserved domain for each sequence in an orthologous group (OG) and filtered noisy and weak associations. Multiple sequence alignment within an OG was generated using MAFFT v7.455. HMMER 3.0 was implemented for each OG alignment to build the hmm profile of conserved domains with the hmmbuild program. The Hmmsearch program was subsequently used to search all the protein sequences in each OG against its HMM profiles with an E-value of 0.00001 as the cut-off. Sequences that fell outside the threshold were excluded from this group and collected together for the second round of construction of the sequence orthology. Finally, every sequence was assigned to an orthogroup. This clustering resulted in 27,631 orthogroups with >=2 members. Phylogenetic relationships for orthogroups with ≥ 5 members were checked by reconstructing a maximal likelihood tree. Sequences were aligned using MUSCLE version 3.8.31 under default settings. The resulting alignments were

trimmed using TrimAI version 1.2[32] in automated mode (*-automated1*). Trimmed alignments (≥50 amino acids) were used to construct a phylogenetic tree using FastTree version 2.1.7 under the 'WAG + CAT' model with four rounds of minimum-evolution SPR moves (*-spr 4*) and exhaustive ML nearest-neighbor interchanges (*-mlacc 2 -slownni*). The branch supports were estimated using the Shimodaira-Hasegawa (SH) test. The successfully reconstructing a tree using all members in an OG indicate the orthologous group clustering based on sequence similarity strategy is also supported by the phylogeny strategy. These trees are able to distinguish the nested position formed by taxonomically distant sequences with variable evolution rates and hence clarify the gene relationships of HGT.

2.3 Construction of gene phylogenies

HGTstart is embedded with the powerful yet user-friendly 'RoutineTree' model which is independently designed by us for unsupervised homologous search, sequence alignment, tree building, and tree screening for predictions of the HGT to every given protein sequence. Briefly, RoutineTree searches homologous against REFAL and builds a single-gene phylogenetic tree for each customized query protein sequence. To speed up the search process, profile hidden Markov models (profile HMMs) and probabilistic inference methods integrated in HMMER 3.0 were employed to construct database segmentation based on functionally conserved domain. REFAL database was divided into families via scanning 159,424 profiles derived from the PFAM database using hmmscan and fetching sequences using esl-fetch. Every family was named by its corresponding PFAM number to facilitate the subsequent retrieval. The excluded REFAL sequences were collected into a FASTA file.

When HGTstart begins, query sequences will be split into single sequence files at the first step and scan the PFAM HMM profiles separately using hmmscan. A temporary BLAST/diamond database for each split query sequence will be built according to its hit PFAM number, to which the corresponding FASTA files in REFAL will be fetched and combined. After making retrieval index of BLAST/diamond format, the temporary database will be searched I against with default *e*-value cut-off =1e-05. For each query, the top 10,000 significant hits, sorted by bit-score in descending order (by default) were recorded. Sequences corresponding to the hits were then retrieved from the temporary database with no more than three sequences for each genus and no more than 10 sequences for each phylum. The significant hits (with query-hit alignment length ≥120 amino acids) were then re-sorted according to query-hit identity in descending order. The homologous sequences plus the query were then combined and aligned using MUSCLE version 3.8.31 under the default setting. The resulting alignments were trimmed using TrimAI version 1.2 in an automated mode (*-automated1*). We discarded queries that showed significantly different amino acid composition ($P < 0.05$) than the remaining sequences in the alignment. The trimmed alignments (≥50 amino acids) were used for the construction of a phylogenetic tree using with IQtree multicore version 1.6.12 under the optimal model calculated by the ModelFinder implemented in the IQtree. Branch support was estimated using ultrafast bootstrap (UFboot, *-bb 1500*) test, Shimodaira-Hasegawa-like approximate likelihood

ratio test (SH-aLRT, *-alrt 1200*), approximate Bayes test (-abyes), Fast local bootstrap probabilities test (-bb 1500). The novel software and the parameters used, as well as the cut-off value are all parameterized in HGTstart and can be customized according to the needs of users.

2.4 Tree-based HGT inference

Phylogenetic trees were searched for topologies with query sequences being nested among sequences. NestedIn, a Java implementation of the phylogenetic tree scanning tool was embedded in Routinetree to screen the nested position in gene trees. A nested position is defined as two or more monophyletic clades comprising a query and its genetically distant sequences supported by different nodes in a tree. In essence, the nested position reflects the conflict between the gene tree and the species tree. To extend the applicability and flexibility, NestedIn provides a user-friendly parameter '*--donor*' to enter any taxonomic node that the user wants to test an HGT relationship with a query, instead of using a built-in species tree as calibration. To the query (receptor) hands, to include all descendent genes after the HGT happened in the monophyletic clade, NestedIn provides a '--optional' parameter for the user to enter an ancestor level of query. Logically, NestedIn reads the tree file in Newick format at first and parses the topology. Starting with the query sequence and going to its up-level nodes one by one, recording every nested node matching the user' set until it reaches the first non-donor, non-optional leaf. It is reasonable that there might be more than one node match the filter criteria. The final node was determined by: 1) To exclude as much contamination and interference with recent HGT as possible, NestedIn refused the singletons for both the donor and receptor genes and provided the customized minimal number to the user. 2) Only nested positions that were multiply supported by a customized cut-off value in a supporting node were retained (default set: ≥0.70 SH-test and ≥0.70 aByes-test). 3) The highest taxonomic node level of all remaining nodes was selected as the final candidate HGT node.

For a given nested position, the monophyletic most similar hit (MMSHs) sequence to query in both the receptor group (MMSH_IN) and the donor group (MMSH_OUT) was given by NestedIn to obtain the predicted function information (MMSH_IN) and list a representative donor gene (MMSH_OUT) to analyze the evolutionary diverse between receptor and donor after HGT.

2.5 Verification of tree-inferring HGT candidates

To construct a systematic methodology for the rapid, exhaustive and credible detection of HGT, we integrate Alien Index (AI), HGT Score Support Index (hU), HGT Branch Length Support Index (hBL) and their consensus hit support indexes into HGTstart to test the confidence of an HGT candidate.

First, we defined some conceptions for a query sequence in its monophyletic clade in the gene tree:

INGROUP: All sequences within a customized rank level (specified by users) that includes the query.

OUTGROUP: All sequences outside a customized rank level (the INGROUP).

SKIPGROUP: The query itself and sequences at the customized lower rank level of the INGROUP boundary, the sequences in SKIPGROUP were supposed to be orthologues that were generated after the HGT.

Alien Index (AI)

We calculated the Alien Index (AI) score for each query gene using e-values of BLAST indexes:

$$AI=\left(\ln(bbhG + 1 * 10^{-200}) - \ln(bbhO + 1 * 10^{-200})\right)$$

BbhG is the e-value of the best hit in the INGROUP lineage, whereas bbhO is the e-value of the best BLAST hit in the OUTGROUP lineage. E-values in SKIPGROUP were skipped given they are supposed to the orthologous after HGT. When no significant BLAST hits were detected, the corresponding bbhG or bbhO was set to 1. The higher the AI score is, the more similar queries are to their homologs in OUTGROUP than to homologs in INGROUP. Since all searches were queried against the same database, applying a single cut-off to all query taxa is reasonable. Because the AI score is combined with other criteria to infer HGTs, we arbitrarily used a less stringent cut-off (AI > 10) as default.

HGT Score Support Index (hU)

We calculated the HGT Score Support Index (hU) score for each query gene based on best bit scores to INGROUP vs OUTGROUP:

HGT Score Support Index (hU) = (Best-hit bitscore of OUTGROUP) - (Best-hit bitscore of INGROUP)

Bitscores in SKIPGROUP were skipped given they are supposed to the orthologous after HGT. When no significant BLAST hits were detected, the corresponding Best-hit bitscore of OUTGROUP or Best-hit bitscore of INGROUP was set to 0. The higher the hU score is, the more similar queries are to their homologs in OUTGROUP than to homologs in INGROUP. We arbitrarily used a less stringent cut-off (hU > 0) as default.

HGT Branch Length Support Index (hBL)

We develop a new index, HGT Branch Length Support Index (hBL), for each query gene based on minimum branch length to query to INGROUP vs OUTGROUP:

HGT Branch Length Support Index (hBL) = (minimum branch length to query of INGROUP) - (minimum branch length to query of OUTGROUP)

Branch length values for SKIPGROUP were skipped given they are supposed to the orthologous after HGT. When no genes were detected, the corresponding minimum branch length to query of INGROUP or minimum branch length to query of OUTGROUP was set to 100. The higher the hBL score is, the more similar queries are to their homologs in OUTGROUP than to homologs in INGROUP. For each leaf, the branch length to query is the sum of branch length of all branches connecting the leaf to the query. Since all branch lengths are normalized in the same tree, we can simply

add them up to compare them between INGROUP and OUTGROUP. We set hBL cut-off (hBL > 0) as default.

Consensus Hit Support

Considering the possibility of sequence contamination introduced accidentally into the INGROUP or OUTGROUP, we calculate the consensus hit support to AI, hU, and hBL respectively. Consensus hit support stands for the support degree by all genes in OUTGROUP, other than the best hit genes. For example, consensus hit support-Evalue (CHE) means the ratio of genes with smaller E values than bbhG to the total gene counts in OUTGROUP. CHE will serve as an indicator of the confidence of AI >0. Similarly, consensus hit support-Score (CHS), means the ratio of genes with a bigger score than bbhG to the total gene counts in OUTGROUP, serving as an indicator of the confidence of hU >0; consensus hit support-Branchlenth (CHBL), means the ratio of genes with smaller branch length than bbhG to the total gene counts in OUTGROUP, serving as an indicator of the confidence of hBL > 0.

In addition to the strategy described above, HGTstart adopted the following points to test the taxonomy of physical flanking genes of an HGT candidate to rule out the possibility of contamination.

1) HGT candidates would be dropped if they are located in a contig in which 50% genes were best hit to other kingdoms;

2) HGT candidates would be dropped if they are located in a contig in which 50% genes were primarily identified as HGT genes;

3) HGT candidates would be dropped if one of their 3 closed flanking genes within both up and downstream is best hit to other kingdoms;

4) Two or more HGT genes that are physically closely linked and fall in the same gene family would be considered as one HGT event, thus all HGTs will be retained.

2.6 Assignment of HGTs to a timeline

To obtain a more straightforward impression of the impact of HGT events on the evolutionary history and geological changes of the earth, we hope to locate the HGT events in a timeline according to when they happened. However, the difficulty of tracing the exact time from gene differentiation based on assessing the accumulation of nucleotide mutations increases with age, given that the more time has passed, the more likely it is that a single site accumulated more than one nucleotide substitution and those unexpected events such as gene loss and gene replication occurred. Limited by the purpose of global browsing and giving an initial impression on HGTs over the existence of an organism, we will not discuss the evolutionary algorithm extensively here. However, we can easily infer the general upper and lower limits of HGT timing based on the following consensus: vertical inheritance is unidirectional. If a successful prokaryote-to-eukaryote HGT event occurred at a certain evolutionary node, for instance, a unicellular individual that is exactly the ancestor of the current Phaeophyta, the homolog of the HGT gene could spread to the descendant node of Phaeophyta by vertical inheritance but not to the ancestor node, Stramenopiles. Thus, for a given query protein sequence identified as the putative HGT, if all its relatives descended from the

initial HGT could be found, we could trace back to a common ancestral node whose occurrence time could be inferred via molecular clock based on archaeological and fossil evidence. Technically, the taxonomic boundary of HGT descendants can be described as the smallest common taxonomic boundary of all gene members in the INGROUP, whereas the smallest common taxonomic boundary of donor descendants can be described as the smallest taxonomic boundary of all gene members in the OUTGROUP. We termed this rule, “the smallest boundary”.

An accurate taxonomy is an essential prerequisite to represent the hierarchical structure and organismal relationships of evolutionary nodes. The NCBI taxonomy system was employed given its widespread acceptance and reasonable taxonomic disciplines of structuring communication concerning all forms of life on Earth. We manually added a kingdom level node, CRASH, to the NCBI taxonomy system, and pushed cryptophytes, rhizarians, alveolates, stramenopiles, and haptophytes to this kingdom according to the proposals in some recent researches[10]. The timeline of the interval nodes was widely collected from the previous study[34].

2.7 Output and the visualization

Both HGTstart’s stand-alone program and web server give a detailed report and implement data visualization to users. For each single query protein, HGTstart provides its homologous genes as well as the corresponding alignments and phylogenetic trees, which can be easily accessed, queried and downloaded by users. For the final identified HGT genes, HGTstart provides the detailed information in a tab-delimited text file, which includes the donor node and receptor node with their occurrence time, MMSH genes in INGROUP and OUTGROUP, scores of AI, hU, hBL as well as their support indexes, and the predicted functional information from a diverse database. Accession numbers listed in the table were hyperlinked to the corresponding external databases such GO and KEGG. HGTstart integrate a genome browse to help users with a more intuitive interface on chromosomes with more descriptive links to HGT information.
Having pre-calculating for more than 1000 species so far, HGTstart provides a resource browse interface for users to scan HGT patterns of species over the TOL. In addition, users can verify that a sequence or species already exists in HGTstart's database through the data search module.

3. Installation

The calculation of HGT is a resource-consuming task. Although HGTstart has built-in computation resources for calculating a small number of sequences, it is recommended to localize HGTstart for computing at the scale of multiple genomes. You can download the latest version of the Routinetree pipeline (for macOS or Linux) and the background database REFAL (may be over 60 GB in size) in the HGTstart platform from https://hgtstart.cn/download/.

3.1. Test if your computer has Java and Perl installed

Please ensure that Perl and Java are available locally.

3.1.1 Java

Type following command in terminal:

Java -version

If java is installed, the following information would be printed:

java version "17.0.1" 2021-10-19 LTS

Java(TM) SE Runtime Environment (build 17.0.1+12-LTS-39)

Java HotSpot(TM) 64-Bit Server VM (build 17.0.1+12-LTS-39, mixed mode, sharing)

If no such version information, maybe you don't have java installed yet. Please go to the java download page https://www.oracle.com/technetwork/java/javase/downloads/index.html, download and install the latest version of Java development kit (JDK).

3.1.2 Perl

Type following command in terminal:

Perl -version

If java is installed, the following information would be printed:

This is perl 5, version 26, subversion 2 (v5.26.2) built for darwin-thread-multi-2level Copyright 1987-2018, Larry Wall

Perl may be copied only under the terms of either the Artistic License or the GNU General Public License, which may be found in the Perl 5 source kit.

Complete documentation for Perl, including FAQ lists, should be found on this system using "man perl" or "perldoc perl". If you have access to the Internet, point your browser at http://www.perl.org/, the Perl Home Page.

If no such version information, maybe you don't have perl installed yet. Please go to the perl download page https://www.perl.org/get.html, download and install the latest version of perl.

3.1.3 decompression

To decompress and set up the routinetree software package, which includes FastTree, blastp, diamond, hmmscan, iqtree, trimal, and muscle, you can follow these steps.

 # Decompress the routinetree_Linux.tar.gz file

tar -zxvf routinetree_Linux.tar.gz

 # Move the database.tar.gz file into the newly created routinetree_Linux directory

mv database.tar.gz routinetree_Linux/

 # Change directory to routinetree_Linux

cd routinetree_Linux

 # Decompress the database.tar.gz file (This may take some time due to its size)

tar -zxvf database.tar.gz

cd database

*tar –zxvf *tar.gz*

cd ..

 # Optionally, remove the database.tar.gz file to save space if you no longer need it

rm database.tar.gz

4. Usage

Overall, the workflow of this pipeline involves several key steps: segmenting the FASTA file into individual sequence files, identifying homologous sequences, performing multiple sequence alignment, constructing phylogenetic trees, locating nested positions, validating horizontal gene transfers (HGTs) through AI, hU, and hBL metrics, and annotating the results.

To expedite tree construction and enhance clarity, we selectively utilize a curated set of sequences, specifically those with the highest BLAST scores within each taxonomic level (kingdom, phylum, class), for tree building.

Each gene, regardless of its copy status, serves as a query for constructing a unique phylogenetic tree.

To bolster accuracy, we employ up to four distinct tree-building methods (neighbor-joining, maximum parsimony, maximum likelihood, and Bayesian inference), ensuring that each node in the tree can be supported by a maximum of four distinct metrics.

We identify the common ancestors of both the donor and receptor organisms independently, relying on the topology of the constructed trees.

To minimize false positives, we introduce parameters "-mdn" and "-mrn" to set thresholds for the minimum number of biological donors and receptors required in a nested position.

Furthermore, we offer "-ssn" and "-asn" parameters to establish minimum thresholds for all supporting nodes and strongly supported nodes that validate the monophyly of the query-donor relationship, thereby enhancing accuracy.

To pinpoint the timing of HGT events, we integrate time information from timetree databases into each node of the trees.

We leverage AI, hU values, and introduce the innovative hBL metric to provide a robust validation framework for identifying HGTs.

Our pipeline offers unparalleled flexibility in verifying HGTs across all hierarchical levels, catering to diverse research needs.

Additionally, we identify the most similar genes to the query within both ingroup and outgroup species, facilitating subsequent comparative gene analysis.

Lastly, we annotate the results and make functional predictions for identified HGTs, providing valuable insights for downstream analysis.

4.1. Quick start

Type this in terminal:

perl /pathtoroutinetree/routinetree_Linux/bin/routinetree2.pl -db /pathtoroutinetree/routinetree_Linux/database -step 0123456 -id 2850 -fl

/pathtoroutinetree/routinetree_Linux/example/sample.faa -gff
/pathtoroutinetree/routinetree_Linux/example/sample.fna -gnm
/pathtoroutinetree/routinetree_Linux/example/sample.gff3

To get a summary of all usage options, type the following command:

perl routinetree2.pl

Or:

perl routinetree2.pl -help

In order to better deal with the complicated calculation process, the routine tree is divided into multiple steps, namely 0123456. The functions of each part will be introduced in detail below. We recommend setting the steps as two parts, 01234 and 56, because the time for the fourth step will be very long.

4.2. More usage information

4.2.1 Input

HGTstart takes a fasta format file containing all protein sequences used to predict HGTs as input.

4.2.2. Mandatory parameter

4.2.2.1 --file (-fl)

To specify fasta file contains all protein sequences used to predict HGTs.

perl routinetree2.pl --file example

4.2.2.2 --taxid (-id)

To specify the NCBI taxnomy id of query species.

perl routinetree2.pl --file example --taxid <NCBITaxId>

4.2.2.3 --database (-db）

The path of database that used in prediction.

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database

4.2.2.4 --step (-step)

To specify the process to run (default = 0123456).

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 0123456

step	software	function
0	split2signle.pl	split the fasta file to files contain one sequence and unify the numbering format of sequence
1	make_blast_db.pl hmmscan esl-fetch blastp/diamond blastout_m7info.sort8iden.pl blastout_m8.phylaClass4.species.pl	search homologous
2	Muscle fst.filterbylen.pl	multiple sequence alignment

	u.move_column.pl trimal.longnames.pl fst.filterbylen.absolute.pl fasta2relaxedPhylip.pl phy.rmorphan.keep_order.pl	
3	iqtree/FastTree	build tree
4	nestedIn13.jar addAnnotation.jar	screen nested position verifies HGTs by AI, hU and hBL, and add predict annotation
5	HGT_filter.pl extract_HGT_seqs.pl VisualizeFlanking.py ResultStructureCollation.pl	Filter HGTs according to the homology of their flanking genes and make visualization Collate the structure of Result files
6	CompareHGTvsCore.pl codon_usage.pl	Compare the gene structure and codon usage bias between HGT and CORE genes

4.2.3. Optional parameters of step 0 to 3, 5

4.2.3.1 --threads (-th)

To specify the number of threads to use (default = 1).

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 01234 --threads 3

This command executes using three threads.

4.2.3.2 --total (-tt)

To specify the total number of sequences included in tree (default=60).

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 01234 --total 50

This command allows up to 50 sequences included in tree.

4.2.3.3 --phylum (-py)

To specify the maximal number of sequences in each phylum (default = 10).

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 01234 --phylum 8

This command allow no more than 8 sequences in each phylum included in tree.

4.2.3.4 --class (-cs)

To specify the maximal number of sequences in each class (default = 10).

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 01234 --class 8

This command stipulates no more than 8 sequences in each class included in tree.

4.2.3.5 --self (-sf)

To specify the maximal number of sequences in each selfspecies (default = 6).

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 01234 --self 4

This command stipulates no more than 4 sequences in each selfspecies included in tree.

4.2.3.6 --seqsearchtool (-sst)

We provided 2 methods to search homologous: *diamond* (default) and *blastp*, and you can choose one according to your preference. The default tool is diamond.

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 01234 --seqsearchtool blastp

This command use *blastp* to search homologous.

4.2.3.7 --treebuildtool (-tbt)

We provided 2 methods to build tree: FastTree and iqtree, and you can choose one according to your preference. The default tool is FastTree.

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 01234 -- treebuildtool iqtree

This command use *iqtree* to build tree.

4.2.3.8 --minimumAI (-minAI)

To specify the minimum value of AI. The default value for this parameter is 0.0.

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 01234 --minimumAI 100.5

This command stipulates HGTs with AI more than 100.5 would be selected.

4.2.3.9 --minimumHU (-minHU)

To specify the minimum value of hU. The default value for this parameter is 0.0.

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 01234 --minimumHU 200.5

This command stipulates HGTs with hU more than 200.5 would be selected.

4.2.3.8 --minimumHBL (-minHBL)

To specify the minimum value of hBL. The default value for this parameter is 0.0.

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 01234 --minimumHBL 50.5

This command stipulates HGTs with hBL more than 50.5 would be selected.

4.2.4. Optional parameters of step 4

4.2.4.1 --donor (-don)

To clearly identify the potential donor of horizontal gene transfer (HGT), which refers to the taxonomic group (not necessarily the direct biological donor, as this is determined based on time and other factors) that does not include the species of interest being queried. When you do not specify this parameter, the software pipeline automatically searches for potential donors within the following taxonomic groups: Amoebozoa, Apusozoa, Bacteria, Excavata, Opisthokonta, Plantae, Chromalveolata, and Viruses.

However, you have the option to manually set this parameter according to your specific needs. It's important to note that this parameter is case-sensitive, meaning that the exact spelling and capitalization of the taxonomic names are important for the software to correctly interpret your input. By setting this parameter, you can tailor the search to focus on specific taxonomic groups of interest.

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 4 --donor Bacteria

This command search for nested position where the donor is Bacteria.

If there are more than one donor taxa, separate multiple donors with comma, e.g.,

"Bacteria, Archaea", type:

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 4 --donor Bacteria,Archaea

4.2.4.2 --cutoff (-cut)

To specify cutoff to define strongly supported nodes (default = 0).

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 4 --cutoff 70

If it is set to 70, all interior nodes supporting query-donor monophyly with support values no less than 70 is considered strong supporting nodes.

4.2.4.3 --optional (-opt)

To specify the optional taxa allowed to present in the query-donor monophyletic ingroup. When you don't set this parameter, it is the kingdom of query species.

Also, you can set this parameter by yourself. This parameter is case sensitive.

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 4 --optional Cyanidioschyzon

If there are more than one optional taxa, separate multiple optionals with comma, e.g., "Cyanidioschyzon, Galderia", type:

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 4 --optional Cyanidioschyzon,Galderia

This parameter allows to search for more ancient HGTs that were shared between query taxon and its closely related taxa. The sequences of optional taxa will be recorded and exported.

4.2.4.4 --ignore (-ign)

To specify the taxa to be ignored while screening trees.

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 4 --ignore Xenopus

This parameter allows to ignore sequences from some taxa which they think might be problematic. The sequences of ignored taxa will be skipped while tree processing and will not be recorded by the program.

4.2.4.5 --ssnode (-ssn)

To specify minimal number of strongly supported nodes (supporting value > cutoff) that supports query-donor monophyly (default = 1).

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 4 --ssnode 2

This command scans for trees with two or more nodes supporting query-donor monophyly (enforced nested position requirement)

4.2.4.6 --asnode (-asn)

To specify minimal number of all supporting nodes (regarless of supporting value) that supports query-donor monophyly (default = 2).

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 4 --asnode 1

This command scans for trees with one or more interior nodes supporting query-donor monophyly (turing off nested position requirement).

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database

--step 4 --asnode 3

This command scans for trees with three or more interior nodes supporting query-donor monophyly (turing off nested position requirement).

4.2.4.7 --minimalReceptorNumber (-mrn)

To specify the minimal number of biological receptors judged based on time in a nested position (default = 2).

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 4 --mrn 3

This command scans for trees with three or more biological receptors judged based on time.

4.2.4.8 --minimalDonorNumber (-mdn)

To specify the minimal number of biological donors judged based on time in a nested position (default = 2).

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 4 --mdn 3

This command scans for trees with three or more biological donors judged based on time.

4.2.4.9 --outgroupsize (-ogs)

To specify the minimal number of sequences in outgroup for a tree to be considered valid (default = 0).

To consider only valide tree with 4 or more sequences in the outgroup, type:

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 4 --ogs 4

4.2.5. Be quiet

If you want to let the process message printed to log file, add "-quiet":

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 01234 –quiet

4.3. Sincere recommendation

4.3.1 Use loose conditions for the first round routinetree

Choose diamond and FastTree to search homologous and build tree, set "--total" to 90, and set "--cutoff" to 0 for the first screening.

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 01234 --total 90

4.3.2 Collect the result of the first screening as the candidates, and perform a second round routinetree.

4.3.3 Use high conditions for the secoud round routinetree

Choose blastp and iqtree to search homologous and build tree, set "--cutoff" to 70, and set minAI, minHU, and minHBL to a high value (for example, 30,70,2) for the second screening.

perl routinetree2.pl --file example --taxid <NCBITaxId> --database database --step 01234 --seqsearchtool blastp -- treebuildtool iqtree --cutoff 70 --minimumAI 30 --minimumHU 70 --minimumHBL 2