DART version 1.0 Manual

Overview

DART (short for “Detection of Additive and Replacing Transfers for single HGT events”) is a software package for classifying horizontal gene transfer (HGT) events as either *additive* or *replacing*. An *additive HGT* occurs when the transferred gene adds itself as a new gene in the recipient genome. When a transferred gene replaces an existing homologous gene in the recipient genome, it is called a *replacing HGT*. The current version of DART only classifies those HGTs where the donor and recipient are both terminal (leaf) edges on the species tree. Note that this version of DART is designed to classify only putative “single-gene” HGTs, where only a small number of genes, ideally only one but perhaps, say, no more than two or three, were simultaneously transferred in the horizontal transfer event.

DART takes as input a rooted species tree, a rooted gene tree for each gene family consisting of at least four genes from the species/genomes under consideration, and gene ordering information for the species/genomes (leaves) represented in the species tree. The software uses phylogenetic reconciliation to infer high-confidence leaf-to-leaf single-gene HGTs and classifies them as “additive”, “replacing”, or “ambiguous”. DART’s classification is based on comparing the gene neighborhood conservation around the transferred gene in the recipient genome as well as in the recipient’s closest phylogenetic neighbors. Further algorithmic and technical details appear in the paper cited below.

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

DART can be cited as follows:

> *Investigating Additive and Replacing Horizontal Gene Transfers using Whole Genomes*
> L. Kloub, S. Gosselin, J. Graf, J. P. Gogarten, M. S. Bansal
> Under review.

Software description

The DART software package consists of two main Python programs: DART.*py*, which implements the core classification approach, and *DART_Aggregate.py,* which computes phylogenetic reconciliations and performs other preprocessing and creates the input file used by DART.*py*. The program *DART_Aggregate.py* also makes use of two precompiled C/C++ executables and another Python program to compute and summarize phylogenetic reconciliations and prepare the final list of high-confidence leaf-to-leaf single-gene HGTs. The precompiled C/C++ executables are already included in the Linux and macOS versions of DART. Next, we describe how to use *DART_Aggregate.py* and DART*.py.*

Using DART_Aggregate.py

DART_Aggregate.py takes as input a set of rooted gene trees, a rooted species tree, and gene ordering information for the species/genomes (leaves) represented in the species tree. It reconciles the gene trees with the species tree (100 reconciliations per gene tree to account for reconciliation uncertainty) using RANGER-DTL v2.0 (included executable) with both "restrictive" and "permissive" transfer costs, infers high-confidence leaf-to-leaf HGTs using the restrictive transfer cost, and then uses the permissive set of inferred HGTs along with gene ordering information along genomes to identify and filter out potential horizontal multi-gene transfers (HMGTs) where more than one gene was likely transferred in a single horizontal transfer event. The final output is the file "Single_HGT.txt", consisting of a list of high-confidence single-gene leaf-to-leaf HGTs. This file can then be used as input for *DART.py*.

DART_Aggregate.py has four required parameters and nine optional parameters, as given below.

```
python DART_Aggregate.py -g <path to the directory containing
rooted gene trees> -s <rooted species tree> -n <path to the
directory containing gene ordering (synteny) files> -f <path to
an empty directory to save output files> [-other options]
```

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.

-n Path to the directory containing gene ordering (synteny) files. This is a required parameter.

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

-r Restrictive transfer cost to use when computing phylogenetic reconciliations. This cost is used to compute the set of high-confidence HGTs to be classified. Defaults to 4. Must be a natural number, and greater than the permissive transfer cost.

-c Permissive transfer cost to use when computing phylogenetic reconciliations. This cost is used to compute a larger set of HGTs used to identify potential multi-gene transfers (HMGTs). Defaults to 3. Must be a natural number.

-e Number of neighboring genes to consider on each side of the transferred gene when calculating conservation of genomic context. Defaults to 4.

-d Number of closest neighboring genes on each side of the transferred gene to consider for HMGT filtering. Any inferred HGTs for which any of these immediate genes are HGTs from the same donor (or rare genes, if a rare gene list is provided) are filtered out as potential horizontal multi-gene transfers. Defaults to 2.

-i Maximum number of HGTs from the same donor (or rare genes, if a rare gene is provided) allowed in the set of genes neighboring the transferred gene. Default to 1; i.e., if the genes neighboring the transferred gene (as specified using the -e option) have 2 or more HGTs from the same donor then that transferred gene is filtered out as belonging to a possible HMGT.

-u Confidence threshold for inferring high-confidence leaf-to-leaf HGTs (using the restrictive transfer cost). Must be between 1 and 100. Defaults to 100.

-m Mapping confidence threshold for HGTs. Must be between 51 and 100. Defaults to 51.

-k File containing list of rare genes to be used for additional filtering of HGTs to be classified. Default is to assume no rare genes.

-h Output this help message.

Format of input files:

Species tree and gene trees:

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);

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.
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_11:11986AfluvialisLMG24681T: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 tree names. The contig number should always be a positive integer.

Format of optional "rare genes" file:

DART_Aggregate.py allows for the option of using a supplied list of rare genes to further filter out HGTs whose classifications may be affected by the close proximity of rare genes. Rare genes are defined to be genes from those gene families that have only a total of one, two, or three 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 and specific gene names can be provided using the -k option.
This file should contain a list of gene families and gene names along with the species in which that gene family is found, one per line, in the following format:
Gene family ID>Gene Name/ID>Species name.

For example:
1183>344>SpeciesA
1183>122>SpeciesD
35232>896>SpeciesA
54364>12>SpeciesB

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 Newick format species tree should 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 *DART_Aggregate.py*).

4. In general, we recommend that users do not change the default parameter values for the HGT inference pipeline (-c, -r, -u, and -m options). These defaults should only be changed if the user has a good understanding of how their changes will impact the inference of the final set of HGTs to be classified.

Test data and test run:

DART_Aggregate.py is available for macOS and Linux operating systems. The *DART* software package, contains a directory named "Test_GeneTrees" containing five gene trees, a directory named "Test_Genomes" containing the gene ordering (synteny) files for each species (leaf) present in the species tree, a file named "Test_speciesTree.newick" containing a rooted species tree, and a file named "Test_rareGenesList.txt" containing a list of rare genes. You can execute DART_Aggregate.py on this test data as follows:

```
mkdir saveResults

python DART_Aggregate.py -g Test_GeneTrees -s
Test_speciesTree.newick -n Test_Genomes -k
Test_rareGenesList.txt -f saveResults
```

Executing *DART_Aggregate.py* on this test data will require approximately 5 to 10 minutes of running time.

Interpreting DART_Aggregate.py output

Homer_Aggregate.py writes all its output to the directory specified using the -f option. This output consists of six sub-directories.

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 *DART_Aggregate.py* finishes executing.

2) "RangerOutput_TransferCost*(transfer cost specified using -c)*" directory: This directory contains the raw reconciliation output generated by RANGER-DTL for each gene tree when using the transfer cost specified by the "-c" option. For example, if a "-c" transfer cost of 3 is used then the directory will be "RangerOutput_TransferCost3".This directory and its contents can be safely ignored, and the directory can be deleted after *DART_Aggregate.py* finishes executing.

3) "RangerOutput_TransferCost*(transfer cost specified using -r)*" directory: This directory contains the raw reconciliation output generated by RANGER-DTL for each gene tree when using the transfer cost specified by the "-r" option. For example, if a "-r" transfer

cost of 4 is used then the directory will be "RangerOutput_TransferCost4". This directory and its contents can be safely ignored, and the directory can be deleted after *DART_Aggregate.py* finishes executing.

4) "ReconciliationResults_TransferCost*(transfer cost specified using -c)*" directory: This directory contains the aggregated reconciliations for each input gene tree when using the transfer cost specified by the "-c" option. For example, if a "-c" transfer cost of 3 is used then the directory will be "ReconciliationResults_Transfer3". Note that the files in this directory have the same names as the original input gene tree files, but their content is different. This directory and its contents can be safely ignored, and the directory can be deleted after *DART_Aggregate.py* finishes executing.

5) "ReconciliationResults_TransferCost*(transfer cost specified using -r)*" directory: This directory contains the aggregated reconciliations for each input gene tree when using the transfer cost specified by the "-r" option. For example, if a "-r" transfer cost of 4 is used then the directory will be "ReconciliationResults_Transfer4". Note that the files in this directory have the same names as the original input gene tree files, but their content is different. This directory and its contents can be safely ignored, and the directory can be deleted after *DART_Aggregate.py* finishes executing.

6) "Single_HGT_List" directory: This directory contains a file named "Single_HGT.txt" that represents the primary output of *DART_Aggregate.py*. This file contains a list of all classifiable high-confidence HGTs that can be classified using *DART.py*. This file must be provided as input to *DART.py* for classification.

Format of "Single_HGT.txt" file:

The Single_HGT.txt file includes a list of donor-recipient pairs, one per line, along with their corresponding HGTs. Only those donor-recipient pair for which at least one classifiable high-confidence HGT was detected are included in this list. For each donor-recipient pair, the specific HGTs detected for that pair are appended one after the other, separated by commas, on the same line. Each HGT is written in the following format "Gene family ID>Gene ID in donor species>Gene ID in recipient species".

For example, consider the following line in the Single_HGT.txt file:

```
SpeciesA>SpeciesB:100>3933>3971,343>3804>1060,688>235>968
```

This line implies that three HGTs were identified between donor species "SpeciesA" and recipient species "SpeciesB", with the first of these HGTs (written as "`100>3933>3971`") being an HGT in the gene family with ID "100", and involving the gene with gene ID "3933" in the donor species and gene ID "3971" in the recipient species.

Using *DART.py*

DART.py should be executed after first executing *DART_Aggregate.py* to create the necessary input file. *DART.py* takes as input the "Single_HGT.txt" file output by *DART_Aggregate.py*, the species tree, 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 DART.py -g <HGTs file> -s <species tree> -n <path to the
directory containing gene ordering (synteny) files> [-other
options]
```

A description of available command line options follows:

-g File containing list of HGTs to be classified. In almost all cases, this will just be the file "Single_HGT.txt" created by executing *DART_Aggregate.py.* 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.

-d Number of closest phylogenetic neighbors of the recipient genome to consider during classification. Defaults to 3.

-e Number of neighborhood genes to consider on each side of the transferred gene. Defaults to 4.

-a Genic neighborhood conservation threshold for additive transfers. Must be between 0 and 99. Defaults to 20; i.e., HGTs for which the genic neighborhood conservation is no more than 20% are classified as additive.

-r Genic neighborhood conservation threshold for replacing transfers. Must be between 0 and 99. Defaults to 80; i.e., HGTs for which the genic neighborhood conservation is at least 80% are classified as replacing.

-b Phylogenetic neighborhood conservation threshold for additive transfers. Must be between 1 and specified number of closest phylogenetic neighbors. Defaults to 3.

-u Phylogenetic neighborhood conservation threshold for replacing transfers. Must be between 1 and specified number of closest phylogenetic neighbors. Defaults to 1.

-f 1|2 Perform randomization analysis to estimate false positive classification rate. 1: for estimating false positive rate for additive classification, 2: for estimating false positive rate for replacing classification.

-h Output this help message.

Format of gene ordering/synteny files:
Same format as mentioned before for *DART_Aggregate*

Format of species tree:
Same format as mentioned before for *DART Aggregate*

Important notes on usage:

1) Output is written directly to the screen and must be redirected to an output file in order to save it.
2) The -b and -u options, together with the -d option, allow users to be less or more permissive in classifying HGTs as additive or replacing. We recommend that users do not change the default values for these classification parameters. These defaults should only be changed if the user has a good understanding of how their changes will impact DART's classification of HGTs.
3) The -a and -r options, together with the -e option, allow users to fine tune classification thresholds to the specific characteristics of their dataset; specifically, to the degree of evolutionary divergence (reflected in conservation of gene orderings along genomes) between the species under consideration. The default values should work well in many scenarios, but these defaults can be changed, as needed, as long as the user understands how these thresholds are used for classification (see associated publication).
4) *DART.py* can also be used to estimate the false positive rate (FPR) for inferred classifications based on the specific characteristics of the dataset. We discuss this functionality in detail towards the end of this manual.

Test data and test run:

The DART software package includes a file named "Test_Single_HGT.txt" (previously generated using *DART_Aggregate.py*) containing the HGTs to be classified, a directory named "Test_Genomes" containing the gene ordering (synteny) files for each species (leaf) present in the species tree, a file named "Test_speciesTree.newick" containing a rooted species tree. *DART.py* can be executed on this dataset as follows:

```
python DART.py -g Test_Single_HGT.txt -s Test_speciesTree.newick -n
Test_Genomes > Classification_results.txt
```

The above command will write the output to the file "Classification_results.txt". Executing this command will require only a few seconds of running time.

Interpreting *DART.py* output

The output of *DART.py* consists of a list of all donor-recipient pairs that have at least one classified HGT. For each such donor-recipient pair, a list of the classified HGTs is output, along with some auxiliary information including (i) the contig number, gene number (i.e., position along the contig), gene name/ID, and gene family ID for the transferred gene in the donor genome, (ii) the contig number, , gene number (i.e., position along the contig), gene name/ID, and gene family ID for the transferred gene in the recipient genome, (iii) the list of gene family IDs (within square brackets) for the genes neighboring the transferred gene along the donor genome, recipient genome, and along the genomes of the closest phylogenetic neighbors (3, by default) of the recipient species (if the corresponding gene family is found in that phylogenetic neighbor), (iv) the neighborhood conservation percentages between the donor and recipient and between the recipient and each of the closest phylogenetic neighbors, and (v) the final classification for that HGT as either additive, replacing, or ambiguous.

After all donor-recipient pairs and their HGTs have been listed, a summary block is output with information on the total number of donor-recipient pairs listed, total number of HGTs classified, and total numbers of HGTs classified as additive, replacing, and ambiguous.

A sample output follows:

```
Donor: AspAMC34, Recipient: AaustrailiensisCECT8023T
Contig: 1, Gene Number: 565, Gene Name: 3711, Gene Family: 1181 -----> Contig: 57, Gene Number: 14, Gene
Name: 3046, Gene Family: 1181

('Donor: ', ['14057', '19613', '18192', '15914'], '1181>3711', ['4274', '2967', '15157', '17780'])
('Recipient: ', ['15694', '14057', '18192', '15914'], '1181>3046', ['4274', '23085', '13276', '17756'])
('AveroniiAK241', ['12686', '13276', '23085', '4274'], '1181>827', ['15914', '18192', '295', '14057'])
('AveroniiAER39', ['12686', '13276', '23085', '4274'], '1181>3589', ['15914', '18192', '19613', '14057'])
('AveroniiTCO21', ['12686', '13276', '23085', '4274'], '1181>3639', ['15914', '18192', '14057', '15694'])

The neighborhood conservation percent between the donor AspAMC34 and recipient AaustrailiensisCECT8023T:
50.0
The neighborhood conservation percent between the neighbor species AveroniiAK241 and recipient: 75.0
The neighborhood conservation percent between the neighbor species AveroniiAER39 and recipient: 75.0
The neighborhood conservation percent between the neighbor species AveroniiTCO21 and recipient: 87.5

Final classification, where < 20% for additive, and > 80% for replacing:
Replacing

Contig: 1, Gene Number: 1891, Gene Name: 1037, Gene Family: 10704 -----> Contig: 21, Gene Number: 78, Gene
Name: 944, Gene Family: 10704

('Donor: ', ['11956', '20166', '21542', '16546'], '10704>1037', ['6210', '6290', '1536', '15124'])
('Recipient: ', ['20166', '21542', '6083', '16546'], '10704>944', ['19267', '14498', '7813', '10761'])
('AveroniiAK241', ['20166', '21542', '6083', '16546'], '10704>2280', ['19267', '5178', '14818', '20800'])
('AveroniiAER39', ['14818', '21148', '540', '19267'], '10704>2635', ['16546', '6083', '21542', '20166'])
('AveroniiTCO21', ['20166', '21542', '6083', '16546'], '10704>3214', ['19267', '18301', '19374', '14818'])

The neighborhood conservation percent between the donor AspAMC34 and recipient AaustrailiensisCECT8023T:
37.5
The neighborhood conservation percent between the neighbor species AveroniiAK241 and recipient: 62.5
The neighborhood conservation percent between the neighbor species AveroniiAER39 and recipient: 62.5
The neighborhood conservation percent between the neighbor species AveroniiTCO21 and recipient: 62.5

Final classification, where < 20% for additive, and > 80% for replacing:
Ambiguous

Donor: AspAMC34, Recipient: AsobriaARS14514
Contig: 1, Gene Number: 2905, Gene Name: 2190, Gene Family: 20114 -----> Contig: 29, Gene Number: 40, Gene
Name: 1700, Gene Family: 20114

('Donor: ', ['10473', '19225', '2744', '19867'], '20114>2190', ['7526', '13295', '6050', '10640'])
('Recipient: ', ['10473', '19225', '2744', '19867'], '20114>1700', ['13295', '6050', '10640', '14369'])

The neighborhood conservation percent between the donor AspAMC34 and recipient AsobriaARS14514: 87.5
Transfer gene not in neighbor species AsobriaJG2080
Transfer gene not in neighbor species AsobriaPAQ0910145
Transfer gene not in neighbor species AsobriaCECT4245T

Final classification, where < 20% for additive, and > 80% for replacing:
Additive
*********************************************************
Summary results:
Total number of donor-recipient pairs: 2
Total number of HGTs: 3

Additive transfers : 1
Replacing transfers: 1
Ambiguous transfers : 1
*********************************************************
```

The above sample output, using the default parameters, shows three classified HGTs (blocks separated by one or two horizontal lines) from two distinct donor-recipient pairs (separated by two horizontal lines), followed by an overall summary.

Note that each classified HGT is specified 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 DART to each gene 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.

In the first donor-recipient pair, the first HGT is classified as replacing since at least one of the closest phylogenetic neighbors of the recipient species (the neighbor "AveroniiTCO21" in this case) has a gene neighborhood conservation percentage of more than 80. The second HGT, from the same donor-recipient pair, is classified as ambiguous since the neighborhood conservation percentages for all 3 of the closest phylogenetic neighbors are less than 80% and more than 20%. The third HGT, from the second donor-recipient pair, is classified as additive since homologs of the transferred gene are not found in any of the 3 closest phylogenetic neighbors of the recipient.

Estimating false positive rate for inferred HGTs

As previously mentioned, DART can also be used to estimate false positive rates (FPR) for inferred classifications based on the specific characteristics of the dataset. When *DART.py* is executed with the command line option “-f 1” or “-f 2” it performs a randomization analysis to estimate how many HGTs would be classified as additive or replacing, respectively, just by chance.

The randomization analysis is performed differently for additive and replacing HGTs as described below.

To determine how often a replacing HGT may get classified as an additive HGT, we consider the null hypothesis that all HGTs are replacing and then estimate the fraction of those HGTs that would be classified as additive using DART with specific parameter settings. To perform this estimation, we repeatedly generate a randomized set of HGTs (all assumed to be replacing) that preserves the overall phylogenetic distribution, counts, and characteristics of the inferred HGTs, and apply DART to these randomized collections of HGTs to determine the fraction of the randomized HGTs that DART infers as being additive. Additional details appear in the associated publication. This randomization test can be invoked using the “-f 1” option.

A similar, but distinct, randomization procedure is used to estimate how often an additive HGT may get classified as a replacing HGT.

The resulting output can be used to estimate the overall FPR for all classified HGTs as well as FPRs for specific donor-recipient pairs. We suggest repeating this randomization test at least 10 times and averaging over the results.

Contact

Please contact either Lina Kloub (lina.kloub@uconn.edu) or Mukul Bansal (mukul.bansal@uconn.edu) in case of any questions, suggestions, or concerns.