User-Manual

Integron Identification and Visualization Pipeline (I-VIP)

Copyright: An-Ni Zhang, Prof. Tong Zhang, University of Hong Kong
Contact: caozhichongchong@gmail.com
Citation:

Introduction

The Integron Visualization and Identification Pipeline (I-VIP) is a well-organized pipeline to identify, classify, annotate and visualize class 1 integrons (Fig 1) in complete/draft genomes and assembled metagenomes. To facilitate flexible application by the users, I-VIP was separated into two modules; Module A for integron identification and classification (orange framework in Fig 1), and Module B for integron extraction, annotation and visualization (blue framework in Fig 1). The I-VIP also provides multiple optional parameters and diverse output formats for further analysis by the user end (more details in the following sections).

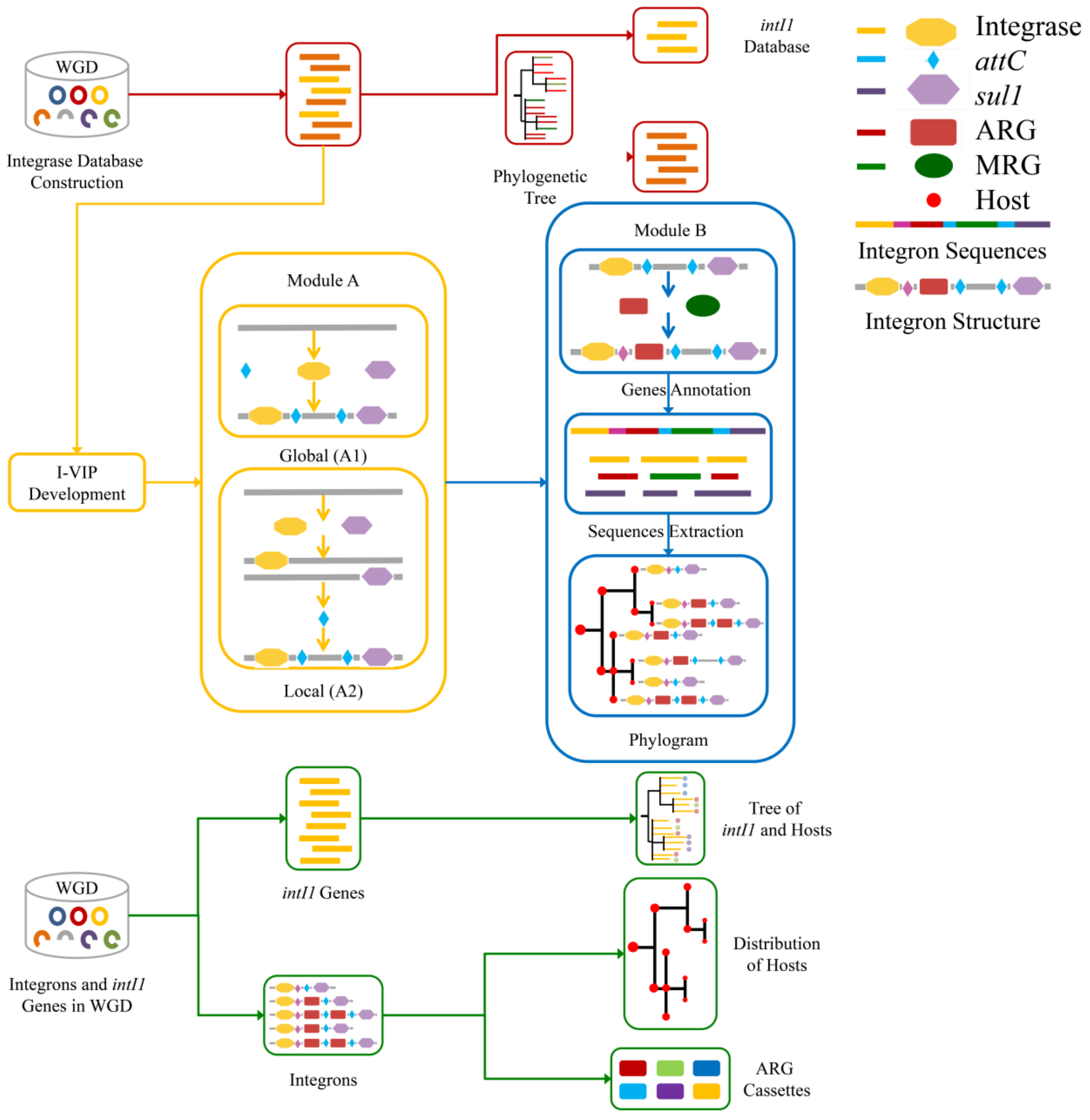


Fig 1. Flow chart of I-VIP.

Preparation

tar xvf I-VIPv1.0.tar.gz

System requirement

- Python ≥ 2.7 (modules including **Biopython** http://biopython.org/wiki/Download, **glob, copy, os, argparse**)
- Cmsearch (http://eddylab.org/infernal/, **--cmsearch**)
- Blast or Blast+ (ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/, **--blastp**)
- The whole package of I-VIP, including **"I-VIP.py"** and the folders of **"scripts"**, **"database"**.
- **Please keep the folders of "scripts" and "database" under the current directory/folder (".") when calling I-VIP.py, or call I-VIP.py under the folder of I-VIP_1.0!** (more examples in the **Example** session)

The I-VIP was built and tested using Python 2.7. If you encounter some problems when using Python ≥ 3.0, please contact An-Ni Zhang by email.

Optional requirement

- Prodigal (https://github.com/hyattpd/Prodigal, **--prodiga**l) for ORF prediction of fasta files and genbank files with empty CDS annotation.
- Usearch (https://www.drive5.com/usearch/, **--usearch**), for two-step search method for integrases, *sul1* genes, antibiotic and metal resistant genes (ARGs and MRGs).
- Diamond (http://ab.inf.uni-tuebingen.de/software/diamond/), for two-step search method for integrases, *sul1* genes, antibiotic and metal resistant genes (ARGs and

MRGs).

- Cytoscape (http://www.cytoscape.org/), for visualizing integron structure with host phylogenetic tree as a phylogram.

Complete path is required to directly call these tools if not in PATH!

Input files

1. The I-VIP supports both the **genbank files** and **fasta files** of complete/draft genomes or assembled contigs, and supports files containing **multiple sequences**. Users can put all sequence files into **a single folder** (**-i**) and input their filename extension (**-f**, such as "**.gbff**" or "**.fa**"). For genbank files, users **must** change their file extension into "**.gbff**" or "**.gbff.gz** ("**.gbff.gz**" which will be extracted by I-VIP). Please make sure that all **sequence/contig names are identical** (no replicate sequence id). Please remove the characters of **":"**, **"____"**, and **"#"** from the filename. It's highly recommended to keep **less than 10,000 files** in the input folder, or the users may merge some small genome/contig files. If the users encounter the **MemoryError**, check whether the file "Temp/all.orf.length" in the output directory has exceed 8Gb. If so, please separate the input files into several folders.
2. The **CDS or ORF** files of the target genomes or contigs are **not compulsory**. If the users would like to provide their own ORF files (for all or part of the source files), please make sure that the ORF files **share exactly the same filename** with their source genome/contig files, and put them under the same input folder. For example, "GCA_000738065.1.fa" (source genome file) and "GCA_000738065.1.faa" (ORF files), here "GCA_000738065.1" is the filename, ".fa" and ".faa" are filename extensions. Also, please input the filename extension (**--o**, such as "**.faa**") of the ORF files and the **method** used to parse or predict the ORFs (**--ot**, **"1"** for extracting from genbank files; **"1"** for predicting by prodigal). The I-VIP will check the

corresponding ORF files for all source genome/contig files, and supplement those missing ORF files by our GbffParser.py (for genbank files) or prodigal prediction (for fasta files).

3. If the users would like use genbank files to annotate the target genome/contig files, please input the folder of genbank files (**--g**), and I-VIP will extract the genbank annotation in the whole range of the integrons. The **filename** of genbank files **should be exactly the same** to the source genome file (only difference is the filename extension).

4. If the users would like to visualize the phylogram of integron structure together with the taxonomy information of the hosts, a file containing all taxonomy metadata should be provided in the **input folder** (together with the source genome files). The first column of the taxonomy file should list all the source genome filename (with or without filename extension). All taxonomy information will be automatically curated to construct a phylogenetic tree (by Taxon_normalization.py). **Please DO revise the empty taxonomy level into: 'NA', or '', or 'None'**. **Example: example/taxon.txt.**

Parameters

Compulsory parameters

“-i”, the input directory or folder of all the target genome/contig files, ORF files (optional) and taxonomy file (optional).

“-f”, the filename extension of the target genome/contig files, for genbank files, please input”-f .gbff” or ”-f .gbff.gz” (other extensions will not be recognized by I-VIP).

Optional file and folder parameters

“--o”, optional: to provide the CDS or ORFs files, please input the file type or filename extension of the CDS or ORFs files, such as “.faa”.

“--ot”, optional: to provide your own CDS or ORFs files, please set the method you used to extract orfs, eg: “1” for genbank parsing or “2” for prodigal prediction, only “1” and “2” are allowed.

“--g”, optional: to provide the genbank files for gene cassettes annotation, please set the directory of Genbank files (default: “None” for no gbff input). The genbank annotation in the whole range of integrons will be extracted.

“--r”, set the output directory or folder of all the integron searching results.

Optional search parameters

'--d', optional: set the distance cutoff for two cassettes to be clustered together (default is “4000”).

'--c', optional: set the e-value cutoff for *attC* search using cmsearch (default is “1.0”).

'--m', optional: set the search strategies for Module A, integron identification (1: global search by Module A1, which is slow but comprehensive; 2: local search by Module A2, which is quick and focuses on Types A-C integrons) The results of Types

A-C integrons using two Modules showed no difference.

“--q”, optional: set the *attC* search using cmsearch (from least strict to most strict: “max”, “nohmm”, “mid”, “rfam”, default is “mid”). The results of “mid” and “max” under the same e-value cutoff of 1.0 showed no difference.

“--a”, optional: whether to annotate gene cassettes against antibiotic and metal resistance databases (“Y” for Yes or “N” for No, default is “Y”).

“--t”, optional: set the thread number assigned for running I-VIP (default 1).

“--tx”, optional: a file of taxonomy metadata (under the input folder, default is “None”).

“--tc”, optional: the column range corresponding to the taxonomy, i.e., from phylum to strain, for example “2,8” that the taxonomy information is arranged in column 2 (lower taxonomy level) to column 8 (higher taxonomy level), default is “None”.

Parameters for software calling

“--u”, optional: use two-step method for integrase and *sul1* search, (“ None” for using one step, which is slower but more comprehensive, “usearch” or “diamond” for using two-step, which is quicker (please input the complete path to usearch or diamond if not in PATH, and make sure the search tools can be directly called. For example, input “usearchv8” if you would like to use this version of usearch).

“--cmsearch”, please input the complete path to cmsearch if not in PATH.

“--prodigal”, please input the complete path to prodigal h if not in PATH.

“--blastp”, please input the complete path to blastp if not in PATH.

e.g. /usr/bin/blastp or /usr/local/bin/blastp

Example

1. Input folder (-i example), input files in the format of .gbff.gz (-f .gbff.gz), use GbffParser.py in scripts to extract contigs and ORFs (--ot 1), provide genbank files (--g example), use Module A2 (local search, --m 2), annotate the ORFs of antibiotic and metal resistance (--a Y), use usearch (--u usearch), output folder (example/example_output), input taxonomy metadata (--tx taxon.txt) and columns for phylum to strain level (--tc 4,10)
   python I-VIP.py -i example -f .gbff.gz --ot 1 --g example --a Y --m 2 --t 1 --u usearch --r example/example_output --tc 4,10 --tx taxon.txt

2. Input folder (-i example), input files in the format of .gbff (-f .gbff), use GbffParser.py in scripts to extract contigs and ORFs (--ot 1), provide genbank files (--g example), use Module A1 (global search, --m 1), annotate the ORFs of antibiotic and metal resistance (--a Y), use diamond (--u diamond), output folder (example/example_output), input taxonomy metadata (--tx taxon.txt) and columns for phylum to strain level (--tc 4,10)
   python I-VIP.py -i example -f .gbff --ot 1 --g example --a Y --m 1 --t 1 --u diamond --r example/example_output --tc 4,10 --tx taxon.txt

3. Input folder (-i example), input files in the format of .fa (-f .fa), input ORF files (--o .faa), ORFs were extracted from genbank files (--ot 1), use prodigal to predict ORFs for input sequence files with no ORF input (--prodigal prodigal), use Module A2 (local search, --m 2), annotate the ORFs of antibiotic and metal resistance (--a Y), use blast directly (--u None, **time-consuming, not recommended!**), output folder (example/example_output), no input taxonomy

metadata (--tx None) and columns for phylum to strain level (--tc None)

```
python I-VIP.py -i example -f .fa --o .faa --ot 1 --a Y --m 2 --t 10 --u None --r
example/example_output --tc None --tx None
```

Output

Name of integrons and integron elements

- The integrons are named in the form of "Genome/Contig filename" + "filename extension" + "_" + "Record id of a sequence in the genome/contig file" + ":" + "The number of this integron on this sequence".
- The integron elements are named in the form of "Integron_name" + ":" + "The number of this integron element on this integron for integrases, *sul1* genes and ORFs" or "*attC* for *attC*".
- For example, the integron name of **GCA_003031325.1.pla.fa_GCA_003031325.1_CP024876.1:2** represents the Filename (GCA_003031325.1.fa), the Record id (GCA_003031325.1_CP024876.1) and the second integron on this sequence.
- The name of integron elements is like **GCA_001250235.2. fa_GCA_001250235.2_LT907990.1:1:1** (first integron element), **GCA_001250235.2. fa_GCA_001250235.2_LT907990.1:1:2** (second integron element), **GCA_001250235.2. fa_GCA_001250235.2_LT907990.1:1:attC** (an *attC*).

Phylogram

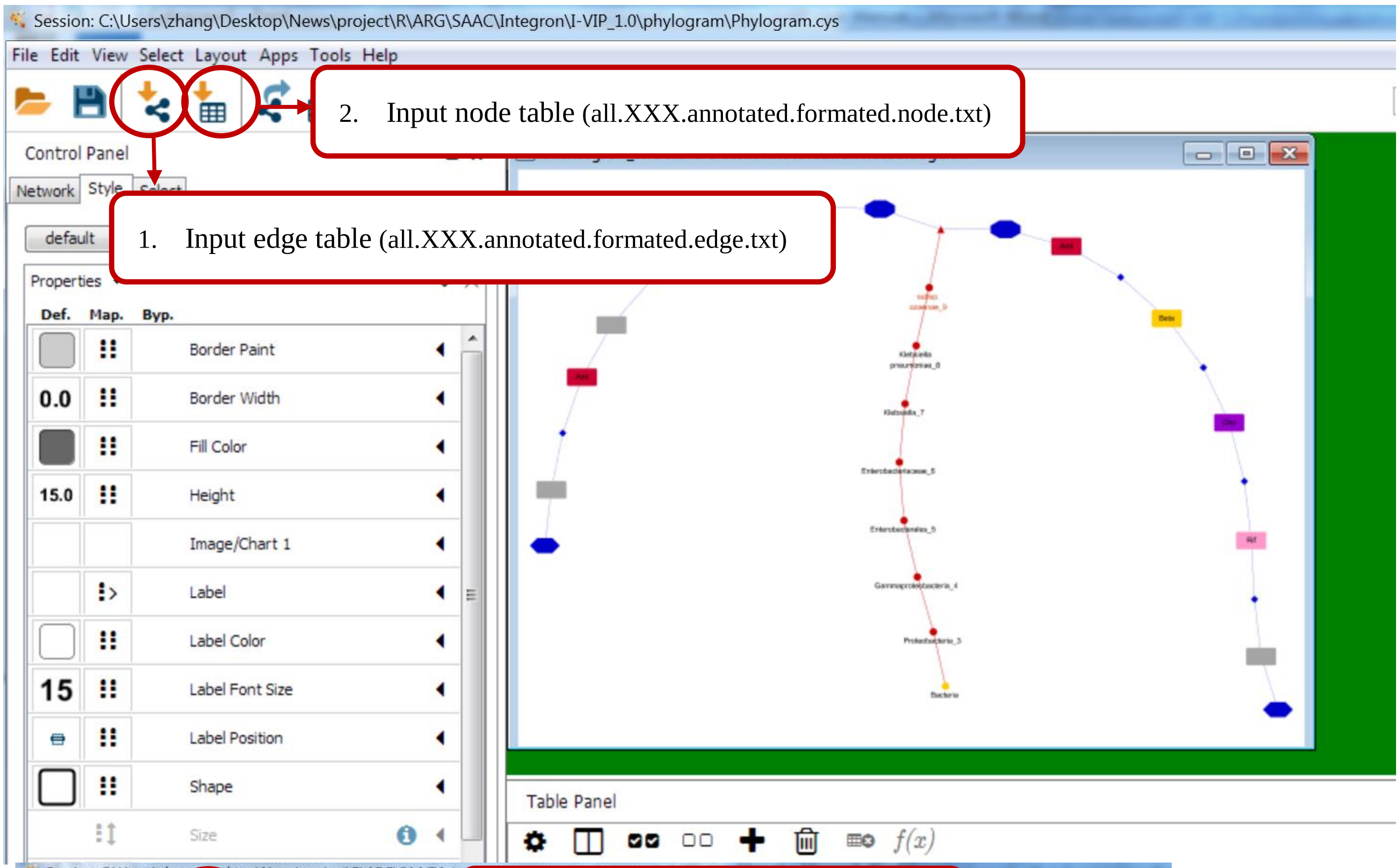
Please try to import the all.Integron_structure.ClassI.annotated.formated.edge.txt and all.Integron_structure.ClassI.annotated.formated.node.txt in I-VIPv1.0/phylogram/.

For Types A-B integrons

- all.Integron_structure.ClassI.annotated.formated.edge.txt (the edge table)
- all.Integron_structure.ClassI.annotated.formated.node.txt (the node table)

For Types C-E integrons

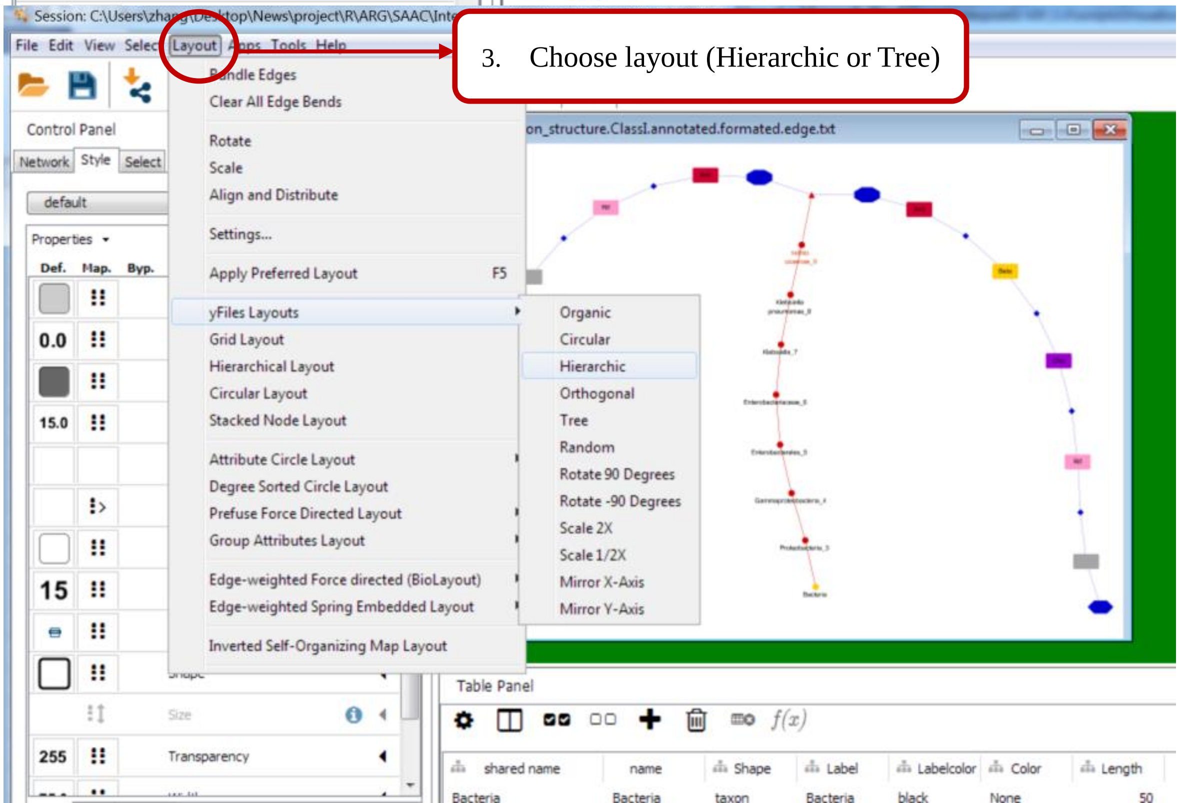
- all.Integron_structure.annotated.formated.edge.txt (the edge table)
- all.Integron_structure.annotated.formated.node.txt (the node table)
- Download the **Phylogram.cys** > import **edge.txt** by “Import **Network** From File” > import **node.txt** by “Import **Table** From File” > change layout by choosing “Layout” > “yFiles Layouts” > **“Tree” or “Hierarchic”**.



Session: C:\Users\zhang\Desktop\News\project\R\ARG\SAAC\Integron\I-VIP_1.0\phylogram\Phylogram.cys

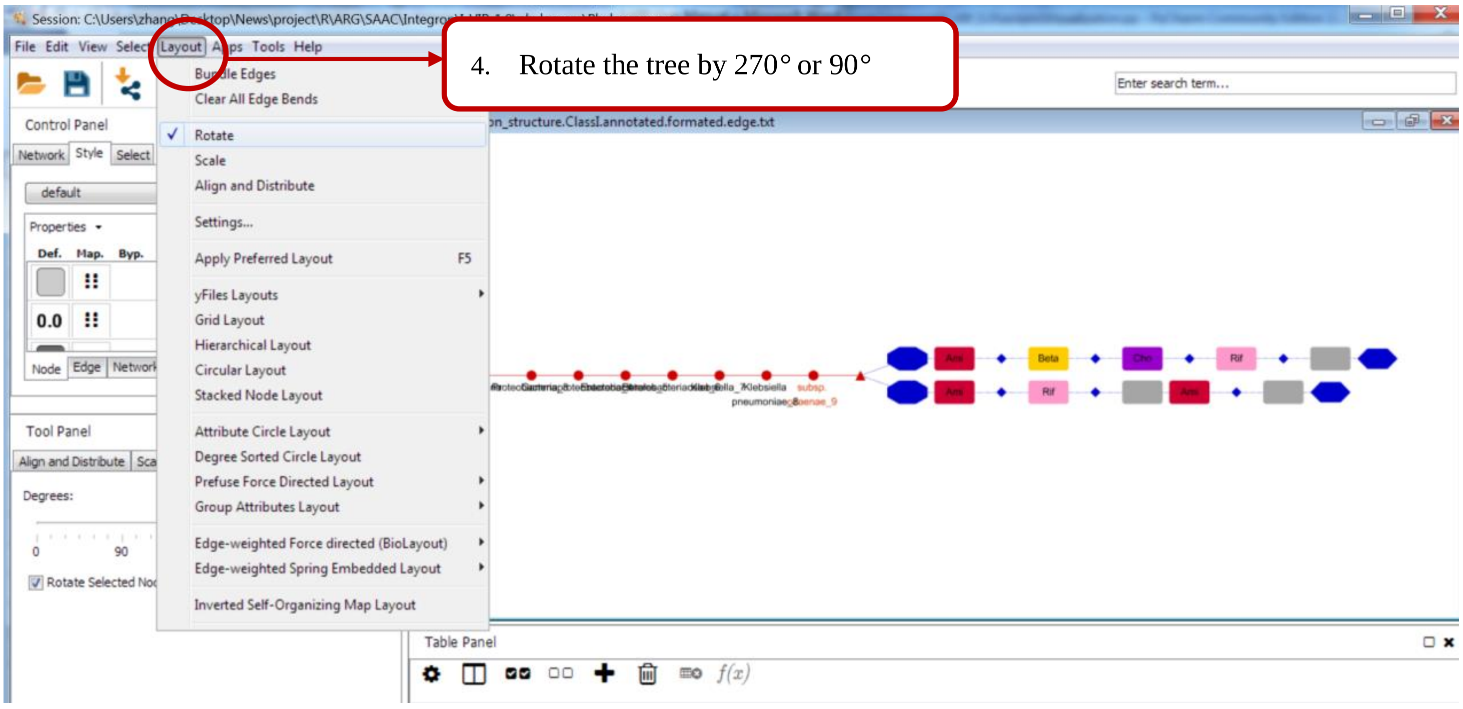
2. Input node table (all.XXX.annotated.formated.node.txt)

1. Input edge table (all.XXX.annotated.formated.edge.txt)

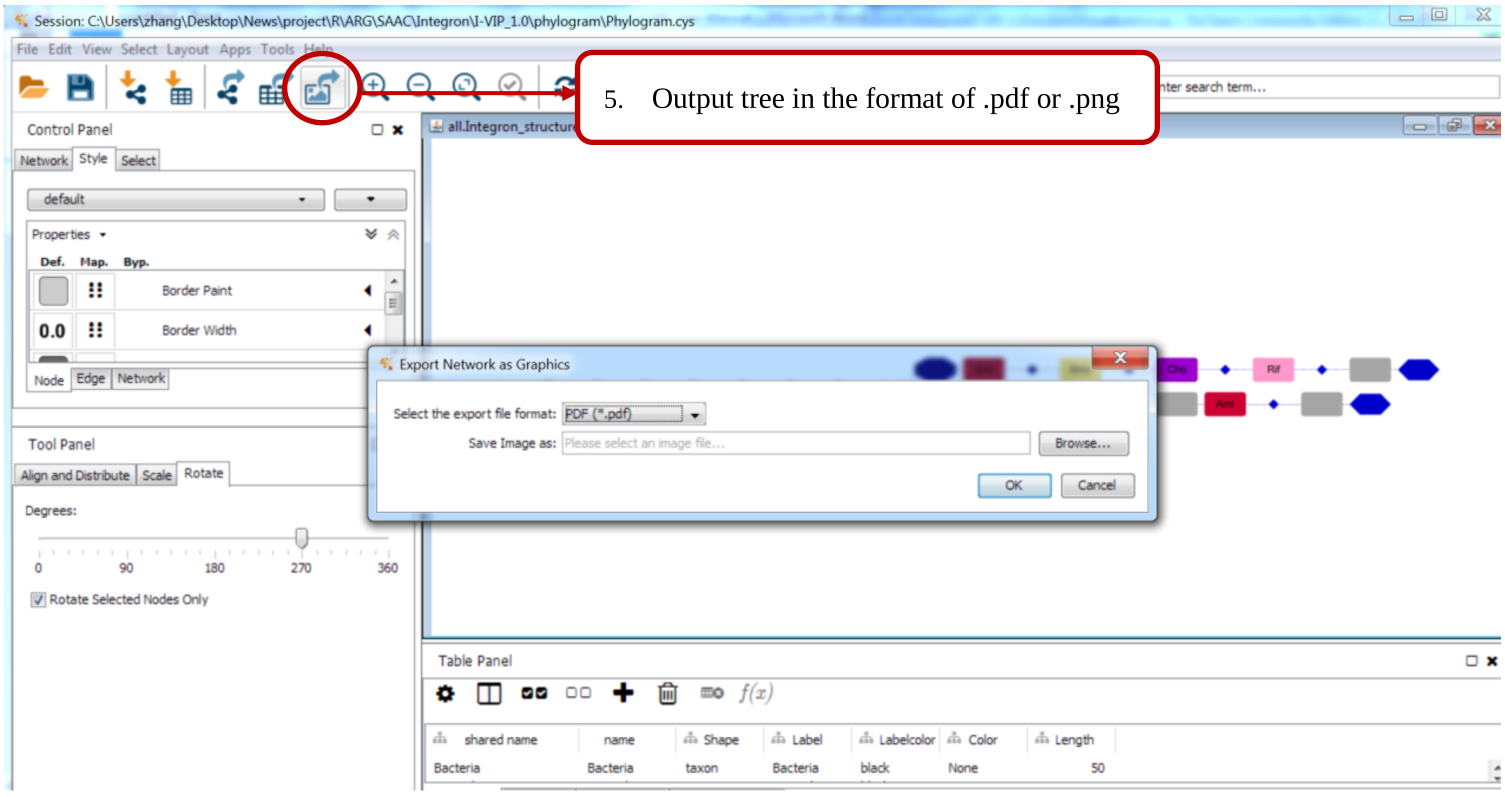


3. Choose layout (Hierarchic or Tree)

shared name	name	Shape	Label	Labelcolor	Color	Length
Bacteria	Bacteria	taxon	Bacteria	black	None	50


4. Rotate the tree by 270° or 90°

Session: C:\Users\zhang\Desktop\News\project\R\ARG\SAAC\Integron\I-VIP_1.0\phylogram\Phylogram.cys
File Edit View Select Layout Apps Tools Help
Bundle Edges
Clear All Edge Bends
Rotate
Scale
Align and Distribute
Settings...
Apply Preferred Layout F5
yFiles Layouts
Grid Layout
Hierarchical Layout
Circular Layout
Stacked Node Layout
Attribute Circle Layout
Degree Sorted Circle Layout
Prefuse Force Directed Layout
Group Attributes Layout
Edge-weighted Force directed (BioLayout)
Edge-weighted Spring Embedded Layout
Inverted Self-Organizing Map Layout
Control Panel
Network Style Select
default
Properties
Def. Map. Byp.
Node Edge Network
Tool Panel
Align and Distribute
Degrees:
Rotate Selected Nodes Only
Enter search term...
Table Panel


5. Output tree in the format of .pdf or .png

Session: C:\Users\zhang\Desktop\News\project\R\ARG\SAAC\Integron\I-VIP_1.0\phylogram\Phylogram.cys
File Edit View Select Layout Apps Tools Help
Control Panel
Network Style Select
default
Properties
Def. Map. Byp.
Border Paint
Border Width
Node Edge Network
Tool Panel
Align and Distribute Scale Rotate
Degrees:
0 90 180 270 360
Rotate Selected Nodes Only
Export Network as Graphics
Select the export file format: PDF (*.pdf)
Save Image as: Please select an image file...
Browse...
OK Cancel
Table Panel

shared name	name	Shape	Label	Labelcolor	Color	Length
Bacteria	Bacteria	taxon	Bacteria	black	None	50

The color, shape, size and label of nodes and edges have been specifically designed for the integron elements, ARG phenotypes and taxonomy. Users are free to change these parameters in “Stype” > “Node” or “Edge”.

If there’s error when loading “Tree” layout, please check your **taxonomy metadata**. This could be caused by one taxonomy that is assigned to two or more different taxonomy level in the metadata (i.e., both as genus and species level in one line or in different lines). Please firstly use “**Taxon_normalization.py**” in the folder of “scripts” to curate the taxonomy metadata. If there’s still problem, please don’t hesitate to contact me by email.

Tip: run I-VIP.py again, change the “--tc” range from phylum, strain into phylum, species (i.e., “--tc 4,10” to “--tc 4,9”), the I-VIP.py will merge the duplicate integrons of one species.

integron

- Folder “ClassI”: all the results for Type A-B integrons
- Folder “Other”: all the results for Type C-E integrons

Each folder stores the integron sequences (“Integron_seqs”), the structure of integrons (“Integron_structure”), integrase sequences (“Integrase_seqs”), all ORFs on the integrons (“ORFs”) and genbank annotation of integron elements and ORFs (“Genbank_annotation”) of individual genome/contig files.

All results of a folder are summarized in the “Integron” folder, with the name of “all.XXX.ClassI.XX” (Types A-B integrons) and “all.XXX.XX” (Type C-E integrons).

About the “Integron_structure”:

- The “all.Integron_structure.ClassI.txt” and “all.Integron_structure.txt” sequentially write out the information of each integron element: “Type of integron”, “Name of integron element", "Integron element type", "Locus on the sequence".

- After annotation against ARG and MRG databases (**--a Y**), the "all.Integron_structure.ClassI.annotated.txt" and "all.Integron_structure.annotated.txt" write out the antibiotic or metal resistance of each ORF: "Type of integron", "Name of integron element", "Integron element type", "Locus on the sequence", "ARG reference sequence", "ARG genotype", "ARG phenotype", "MRG reference sequence", "MRG phenotype". For no ARG and MRG, "None" is filled in.
- After combining with the taxonomy information (**--tx your_taxonomy_file, --tc taxa_column1, taxa_column2**), the "all.Integron_structure.ClassI.annotated.formated.txt" and "all.Integron_structure.annotated.formated.txt" write out the taxonomy information for each integron: "Type of integron", "Filename without extension", "Integron name", "List of all integron elements with annotation and locus", "Total number of ARGs and MRGs", "Taxonomy information". If no taxonomy metadata is provided, I-VIP will write all the above information excluding the "Taxonomy information".

result

The integron classification information is stored under the "result" folder, including the "Type of integron", "Integron name", "Number of *attC*", "Loci of integron elements", and "Annotation and types of integron elements".

output

The "output" folder stores all the sequence-based search results of integrases, *attC*, *sul1* genes by Blastp and cmsearch.

Other information

1. The users are free to go through all python scripts under the “scripts” folder and do any changes they would like. If you would like to make changes to the **result folders**, such as ‘Temp’, ‘output’, ‘extract’, ‘result’, ‘Integron’, please make sure you **change them in all the python scripts**.
2. In ModuleA1 and Module A2 (search *attC* using cmsearch), large genome/contig files (≥ 2Mb size) will be split into 2Mb-subfiles for more comparable results.

Citation

1. This study.

2. (attC database) Cury J, Jove T, Touchon M, Neron B, Rocha EP: Identification and analysis of integrons and cassette arrays in bacterial genomes. Nucleic acids research 2016, 44:4539-4550.

3. (optional: antibiotic resistance database) Yang Y, Jiang X, Chai B, Ma L, Li B, Zhang A, Cole JR, Tiedje JM, Zhang T: ARGs-OAP: online analysis pipeline for antibiotic resistance genes detection from metagenomic data using an integrated structured ARG-database. Bioinformatics 2016.

4. (optional: metal resistance database) Li L-G, Xia Y, Zhang T: Co-occurrence of antibiotic and metal resistance genes revealed in complete genome collection. The ISME Journal 2016.