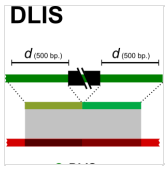

Feb 26, 2021

Finding insertion sequence mobilization events with ISCompare.

Ezequiel G. Mogro[1], Nicolas Ambrosis[1], Mauricio Lozano[1]

[1]Instituto de Biotecnología y Biología Molecular (IBBM)

1 *Works for me* dx.doi.org/10.17504/protocols.io.bst6nere

Mauricio Lozano

SUBMIT TO PLOS ONE

ABSTRACT

Insertion sequences (ISs) are small transposable elements composed only by a transposase and imperfect terminal inverted repeats, which have an important role in genome evolution and contribute to bacterial genome plasticity and adaptability. Bacterial strains from a same species usually present genome rearrangements and variation in the location of insertion sequences and other transposable elements, which might produce phenotypic variations, including antibiotic resistance and adaptation to vaccination strategies. We developed ISCompare to profile IS mobilization events in related bacterial strains. Here we present a comprehensive description on how to use ISCompare, and interpret the obtained results.

Basic protocol: Automatic search of Differentially located Insertion Sequences (DLIS)
Support protocol 1: Search for differentially located ISs using local files
Support protocol 2: Using the shift mode to identify differentially located Group II introns

EXTERNAL LINK

https://github.com/maurijlozano/ISCompare

THIS PROTOCOL ACCOMPANIES THE FOLLOWING PUBLICATION

Easy identification of insertion sequence mobilization events in related bacterial strains with ISCompare E.G. Mogro, N. Ambrosis, M.J. Lozano bioRxiv 2020.10.16.342287; doi: https://doi.org/10.1101/2020.10.16.342287

DOI

dx.doi.org/10.17504/protocols.io.bst6nere

EXTERNAL LINK

https://github.com/maurijlozano/ISCompare

PROTOCOL CITATION

Ezequiel G. Mogro, Nicolas Ambrosis, Mauricio Lozano 2021. Finding insertion sequence mobilization events with ISCompare.. **protocols.io** https://dx.doi.org/10.17504/protocols.io.bst6nere

MANUSCRIPT CITATION please remember to cite the following publication along with this protocol

Easy identification of insertion sequence mobilization events in related bacterial strains with ISCompare E.G. Mogro, N. Ambrosis, M.J. Lozano bioRxiv 2020.10.16.342287; doi: https://doi.org/10.1101/2020.10.16.342287

Citation: Ezequiel G. Mogro, Nicolas Ambrosis, Mauricio Lozano (02/26/2021). Finding insertion sequence mobilization events with ISCompare.. https://dx.doi.org/10.17504/protocols.io.bst6nere

CREATED

Feb 26, 2021

LAST MODIFIED

Feb 26, 2021

PROTOCOL INTEGER ID

47710

Introduction

1 Bacteria accessory genome is composed, in its majority, of mobile genetic elements (MGE). MGEs are grouped into two classes: plasmids and bacteriophages, and transposable elements (TEs) (Siguier *et al.*, 2014). Among the TEs, Insertion sequences (ISs) are the smallest, being only formed by a transposase coding region and imperfect terminal inverted repeats(Mahillon and Chandler, 1998). ISs can spread in a genome by transposition, and participate in genome evolution, being mediators of genomic rearrangements, gene inactivation, over-expression and modulation of the expression of neighbor genes (Siguier *et al.*, 2014). In addition, ISs contribute to bacterial phenotypic variation (Schneider and Lenski, 2004), altering the resistance to antibacterial agents (Mugnier *et al.*, 2009), virulence, pathogenicity, catabolism (Vandecraen *et al.*, 2017), defence against harmful genes (Fan *et al.*, 2019) and adaptation of bacterial strains to vaccination strategies (Pawloski *et al.*, 2014; Carriquiriborde *et al.*, 2019; Zomer *et al.*, 2018). Profiling IS insertion sites and their differential location within bacterial species is thus of great importance.

We developed a new program, **ISCompare** (Mogro EG *et al.*, 2020), in order to automate the detection of IS mobilization events in related bacterial strains. Here we present a comprehensive description on how to use ISCompare, and interpret the obtained results.

Background Information

Insertion sequences are drivers of genome evolution and adaptability of microorganisms, which makes the analysis of ISs location of great importance. The principal methods used for IS profiling have been restriction fragment length polymorphism (Das et al., 1995), chromosomal DNA hybridization (Fan et al., 2019; Soria et al., 1994) and diverse PCR related methods (Bik et al., 1996; Lozano et al., 2010; Suzuki et al., 2004). With the increasing amount of whole genome sequencing projects for microorganisms, several bioinformatic tools were developed for the automatic identification of ISs (ISFinder, Siguier et al., 2006; ISEScan, Xie and Tang, 2017; Oasis, Robinson et al., 2012; ISQuest, Biswas et al., 2015). In addition, some tools are able to compare the location of ISs between bacterial strains (Breseq, Barrick et al., 2014; Transposon Insertion Finder, Nakagome et al., 2014; ISMapper, Hawkey et al., 2015; and panISa, Treepong et al., 2018). Most of these programs are based on the soft mapping of short reads from whole genome shotgun sequencing experiments to a reference genome. Some disadvantages of these programs are that they require a high enough genome coverage for an accurate detection of ISs, and that they are difficult to apply to the analysis of massive genomic datasets (Adams et al., 2016). Also, obtaining genome sequences with high sequence coverage for numerous bacterial strains is costly and laboratories in developing countries often do not have the required resources. To overcome some of these limitations, Adams et al. (2016) designed ISSeeker for the rapid and high-throughput mapping of ISs using whole genome sequence assemblies, including draft genomes. In draft genomes, contigs are typically broken at IS locations (Sohn and Nam, 2018) and contain partial IS sequences at their ends. ISSeeker uses Blast (Altschul et al., 1990) to identify the locations of a provided insertion sequence, then its flanks are extracted and mapped against a reference genome. Some of the limitations of ISSeeker are that it requires background knowledge on the ISs present in the test organism, that it can analyse one IS at a time, and that the comparison to establish differentially located ISs (DLIS) has to be manually done.

We developed ISCompare (Mogro EG et al., 2020) to improve on these issues, and to facilitate the interpretation and analysis of the results. ISCompare uses Blast to search simultaneously for all the ISs in a database in both the reference and query genomes, and to determine and compare their genomic location by analysing their flanking sequences. Briefly, in a first step ISCompare uses blast to search for ISs in the query genome. Then, the Query IS Flanks (QIF) are extracted and used in a blast search against the reference genome. The correct matches are analysed and

Citation: Ezequiel G. Mogro, Nicolas Ambrosis, Mauricio Lozano (02/26/2021). Finding insertion sequence mobilization events with ISCompare.. https://dx.doi.org/10.17504/protocols.io.bst6nere

their flanks (Reference Anchor Flanks, RAFs) tested for the presence of ISs. If a RAF presents an IS, then the location of that IS in both genomes is the same. Otherwise, a differentially located IS (DLIS) is informed. Further, we implemented a novel shift mode that allows the correct identification of DLISs which flanks present other ISs (i.e. the case of two or more adjacent IS) or repeated sequences (i.e. the case of multicopy genes).
ISCompare outputs several tables (see, Guideline for understanding the results) containing the mapping information, differential location status and the annotation data of the flanking sequences on both the query and reference genomes. In addition, with the -p argument, a graphic PDF report is generated, which can be used to confirm the results.

LITERATURE CITED

- Adams,M.D. et al. (2016) Quantitative assessment of insertion sequence impact on bacterial genome architecture. Microb. Genomics, 2, e000062.
- Afgan,E. et al. (2016) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. Nucleic Acids Res., 44, gkw343.
- Altschul,S.F. et al. (1990) Basic local alignment search tool. J Mol Biol, 215, 403–410.
- Auch,A.F. et al. (2010) Standard operating procedure for calculating genome-to-genome distances based on high-scoring segment pairs. Stand. Genomic Sci., 2, 142–148.
- Bankevich,A. et al. (2012) SPAdes: A new genome assembly algorithm and its applications to single-cell sequencing. J. Comput. Biol., 19, 455–477.
- Barrick,J.E. et al. (2014) Identifying structural variation in haploid microbial genomes from short-read resequencing data using breseq. BMC Genomics, 15, 1039.
- Bart,M.J. et al. (2014) Global population structure and evolution of Bordetella pertussis and their relationship with vaccination. MBio, 5, 1–13.
- Belcher,T. and Preston,A. (2015) Bordetella pertussis evolution in the (functional) genomics era. Pathog. Dis., 73, ftv064.
- Biswas,A. et al. (2015) ISQuest: Finding insertion sequences in prokaryotic sequence fragment data. Bioinformatics, 31, 3406–3412.
- Carriquiriborde,F. et al. (2019) Rare Detection of Bordetella pertussis Pertactin-Deficient Strains in Argentina. Emerg. Infect. Dis., 25, 2048–2054.
- Cock,P.J.A. et al. (2009) Biopython: Freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics, 25, 1422–1423.
- Dewan,K.K. et al. (2020) Acellular pertussis vaccine components: Today and tomorrow. Vaccines, 8.
- Fan,C. et al. (2019) Defensive Function of Transposable Elements in Bacteria.
- Goris,J. et al. (2007) DNA-DNA hybridization values and their relationship to whole-genome sequence similarities. Int. J. Syst. Evol. Microbiol., 57, 81–91.
- Hawkey,J. et al. (2015) ISMapper: Identifying transposase insertion sites in bacterial genomes from short read sequence data. BMC Genomics, 16, 1–11.
- Hiramatsu,Y. et al. (2017) Significant decrease in pertactin-deficient Bordetella pertussis isolates, Japan. Emerg. Infect. Dis., 23, 699–701.
- Inatsuka,C.S. et al. (2010) Pertactin is required for Bordetella species to resist neutrophil-mediated clearance. Infect. Immun., 78, 2901–2909.
- Kitts,P.A. et al. (2016) Assembly: A resource for assembled genomes at NCBI. Nucleic Acids Res., 44, D73–D80.
- Letunic,I. and Bork,P. (2019) Interactive Tree of Life (iTOL) v4: Recent updates and new developments. Nucleic Acids Res., 47, W256–W259.
- López,J.L. et al. (2019) Codon Usage Heterogeneity in the Multipartite Prokaryote Genome: Selection-Based Coding Bias Associated with Gene Location, Expression Level, and Ancestry. MBio, 10, 1–20.
- Mahillon,J. and Chandler,M. (1998) Insertion Sequences. Microbiol. Mol. Biol. Rev., 62, 725–774.
- Maruya,J. and Saeki,K. (2010) The bacA Gene Homolog, mlr7400, in Mesorhizobium loti MAFF303099 is Dispensable for Symbiosis with Lotus japonicus but Partially Capable of Supporting the Symbiotic Function of bacA in Sinorhizobium meliloti. Plant Cell Physiol., 51, 1443–1452.
- Mechanize - Automate interaction with HTTP web servers. v0.4.5 2020.
- Melvin,J.A. et al. (2014) Bordetella pertussis pathogenesis: Current and future challenges. Nat. Rev. Microbiol., 12, 274–288.
- Mogro EG et al. (2020) Easy identification of insertion sequence mobilization events in related bacterial strains with ISCompare. bioRxiv, 2020.10.16.342287.
- Mugnier,P.D. et al. (2009) Functional analysis of insertion sequence ISAba1, responsible for genomic plasticity of acinetobacter baumannii. J. Bacteriol., 191, 2414–2418.
- Nakagome,M. et al. (2014) Transposon Insertion Finder (TIF): A novel program for detection of de novo

Citation: Ezequiel G. Mogro, Nicolas Ambrosis, Mauricio Lozano (02/26/2021). Finding insertion sequence mobilization events with ISCompare.. https://dx.doi.org/10.17504/protocols.io.bst6nere

transpositions of transposable elements. BMC Bioinformatics, 15, 71.
- Pawloski,L.C. et al. (2014) Prevalence and molecular characterization of pertactin-deficient Bordetella pertussis in the United States. Clin. Vaccine Immunol., 21, 119–125.
- Reback,J. et al. (2020) pandas-dev/pandas: Pandas 1.1.2.
- Robinson,D.G. et al. (2012) OASIS: An automated program for global investigation of bacterial and archaeal insertion sequences. Nucleic Acids Res., 40, e174–e174.
- Sallet,E. et al. (2013) Next-generation annotation of prokaryotic genomes with EuGene-P: application to Sinorhizobium meliloti 2011. DNA Res., 20, 339–54.
- Schneider,D. and Lenski,R.E. (2004) Dynamics of insertion sequence elements during experimental evolution of bacteria. Res. Microbiol., 155, 319–327.
- Seemann,T. (2014) Prokka: Rapid prokaryotic genome annotation. Bioinformatics, 30, 2068–2069.
- Siguier,P. et al. (2014) Bacterial insertion sequences: Their genomic impact and diversity. FEMS Microbiol. Rev., 38, 865–891.
- Siguier,P. et al. (2006) ISfinder: the reference centre for bacterial insertion sequences. Nucleic Acids Res., 34, D32.
- Team,R.C. (2014) R: A language and environment for statistical computing.
- Toro,N. et al. (2018) Contribution of mobile group II introns to Sinorhizobium meliloti genome evolution. Front. Microbiol., 9, 1–8.
- Toro,N. et al. (2016) The early events underlying genome evolution in a localized Sinorhizobium meliloti population. BMC Genomics, 17, 1–14.
- Treepong,P. et al. (2018) PanISa: Ab initio detection of insertion sequences in bacterial genomes from short read sequence data. Bioinformatics, 34, 3795–3800.
- Vandecraen,J. et al. (2017) The impact of insertion sequences on bacterial genome plasticity and adaptability. Crit. Rev. Microbiol., 43, 709–730.
- Van Der Walt,S. et al. (2011) The NumPy array: A structure for efficient numerical computation. Comput. Sci. Eng., 13, 22–30.
- Xie,Z. and Tang,H. (2017) ISEScan: automated identification of insertion sequence elements in prokaryotic genomes. Bioinformatics, 33, 3340–3347.
- Zomer,A. et al. (2018) Bordetella pertussis population dynamics and phylogeny in Japan after adoption of acellular pertussis vaccines. Microb. genomics, 4, e000180.
- Zulkower,V. and Rosser,S. (2020) DNA Features Viewer, a sequence annotations formatting and plotting library for Python. bioRxiv Prepr.

Downloading and installing ISCompare

2 **ISCompare Requirements:**

Hardware
ISCompare can run on any modern laptop or desktop PC.

Software
ISCompare was tested on Linux and Windows, although it should also run without problems on MacOS.
ISCompare is free and open source and can be downloaded from https://github.com/maurijlozano/ISCompare with any internet browser (For download and install instruction see below).
ISCompare requires Python and NCBI-Blast+ programs. In addition the following python modules are required: Biopython, DNA_features_viewer, Pandas, Numpy and mechanize.

Input data
ISCompare requires the NCBI Assembly accession numbers of the reference and query genomes.

2.1 Access https://github.com/maurijlozano/ISCompare and click on code to clone repository (download as zip)

2.2 Uncompresss and locate the extracted folder in the desired location.

Citation: Ezequiel G. Mogro, Nicolas Ambrosis, Mauricio Lozano (02/26/2021). Finding insertion sequence mobilization events with ISCompare.. https://dx.doi.org/10.17504/protocols.io.bst6nere

2.3 Install Blast. Blast can be downloaded from https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/. In the case of Linux users Blast+ can be downloaded and installed as follows. Open a terminal and type: sudo apt install ncbi-blast+.

2.4 Make sure that blast executables can be accessed from any folder (i.e. are on the PATH environment variable).

Windows users:
i. Open the Start Search, type in “env”, and choose “Edit the system environment variables”
ii. Go to Environment variables
iii. Select PATH and edit
iv. Select New and add as value the complete path of blast binaries (e.g. C:\Program Files\NCBI\blast-2.10.1+\bin)
v. for blast to run correctly two extra environment variables must be created:
• A new BLASTDB environment variable as pointer to database location, with "blast_install_dir\db" as value (e.g. C:\Program Files\NCBI\blast-2.10.1+\db)
• A new BLASTDB_LMDB_MAP_SIZE, with 1000000 as its value (needed to optimize makeblastdb operations when creating new database files)

Linux users:
If blast was installed using apt install, the binaries will be accessible from any folder with no further steps. Otherwise, following line ‘export PATH=$PATH:/blast_install_dir/bin’ should be edited or added to .bashrc file located on the home directory, where blast_install_dir is the blast install folder, and binaries are located on ./bin child folder.

2.5 Install python and the required modules. Linux distributions usually have python preinstalled.
In the case of windows, Python can be downloaded from https://www.python.org/downloads/windows/ and installed as any other program. To install the required modules open a command line window and use:

```
pip3 install ‘module_name’
```

where module_name is the required python module. In the case of windows, some modules need to be installed with pipwin. To install pipwin type:

```
pip install pipwin
```

Then,to install the modules type:

```
pipwin install ‘module_name’
```

If biopython was installed with pipwin, the error 'Bio module not found' could appear. In that case, go to your python install folder, \Lib\site-packages and verify that Biopython module folder is 'Bio' and not 'bio'.

Basic protocol: Automatic search of Differentially Located Insertion Sequences (DLIS)

3 In the automatic mode, ISCompare only require the accession numbers of the query and reference genomes. ISCompare uses Biopython module to download the genomes in genbank and fasta format. Then it runs an ISFinder server blast search, and downloads the sequences in fasta format of all the found ISs. Finaly, ISComapre algorithm is run to detect DLISs.

3.1 Open a terminal (command line window) and change to the ISCompare root folder.

3.2 Make the ISCompare.py file executable. In linux it can be done by running the following command:

Citation: Ezequiel G. Mogro, Nicolas Ambrosis, Mauricio Lozano (02/26/2021). Finding insertion sequence mobilization events with ISCompare.. https://dx.doi.org/10.17504/protocols.io.bst6nere

```
chmod +x ISCompare.py
```

3.3 Type:

```
./ISCompare.py -h
```

to test the correct installation and display the program help.

3.4 Accession numbers for the genomes to compare can be easily obtained from NCBI genomes (Link). For assemblies with multiple replicons or scaffolds, the assembly accession number should be used.

3.5 To run ISCompare on the command line window type:

```
python ISCompare.py -Q ACC_NUM_Query -R ACC_NUM_Ref  -I -e your@email -o
resultsDirName
```

For linux command line you can also run:

```
./ISCompare.py -Q ACC_NUM_Query -R ACC_NUM_Ref  -I -e your@email -o
resultsDirName
```

Support Protocol 1: Search for differentially located ISs using local files

4 In this support protocol we are going to search for DLISs using a local IS database and local query and reference genome files in genbank format. In addition we will indicate ISCompare to clean temporal files and produce a PDF plot of the IS genomic surrounds.

4.1 Download the sequence in fasta format of the ISs of interest. Concatenate those sequences in multifasta file.

4.2 Download the genomic sequence of both the reference and query genomes in genbank file format.

4.3 Place all the files in the ISCompare root folder (Otherwise a full path to the genome and IS database files will be required)

4.4 Open a terminal (command line window) and change to the ISCompare root folder.

4.5 To run ISCompare on the command line window type:

```
python ISCompare.py -q Query_genome.gb -r Ref_genome.gb  -i ISdatabase.fasta -o
ResultsDirName -c -p
```

For linux command line you can also run:

```
./ISCompare.py -q Query_genome.gb -r Ref_genome.gb  -i ISdatabase.fasta -o
ResultsDirName -c -p
```


Citation: Ezequiel G. Mogro, Nicolas Ambrosis, Mauricio Lozano (02/26/2021). Finding insertion sequence mobilization events with ISCompare.. https://dx.doi.org/10.17504/protocols.io.bst6nere

Support protocol 2: Using the shift mode to identify differentially located Group II Introns

5 Some bacterial genomes present regions with a high density of insertion sequences where it's common to find two or more adjacent ISs that are, in general, difficult to analyse since blast searches will yield more than one good match. The same happens in the case of ISs which are inserted into or near repeated sequences. For example, bacterial Group II introns are mobile elements that are inserted in most cases within ISs, and thus present repeated sequences on both flanks. On such cases ISCompare can achieve better results using the shift mode.

5.1 Compile a multifasta file containing the sequences of all variants of Group II introns which can be found on the genomes to analyse. This file will be used as the IS database.

5.2 Download the query and reference genomes in genbank format [or get the accession numbers]

5.3 Place all the files in the ISCompare root folder (Otherwise a full path to the genome and IS database files will be required)

5.4 Open a terminal (command line window) and change to the ISCompare root folder.

5.5 Run ISCompare with the following command:

```
python ISCompare.py -q Query_genome.gb -r Ref_genome.gb  -i Isdatabase(Group II
Introns).fasta -o ResultsDirName -c -p -S 3000
```

or in linux

```
./ISCompare.py -q Query_genome.gb -r Ref_genome.gb  -i Isdatabase(Group II
Introns).fasta -o ResultsDirName -c -p -S 3000
```

Here, the -S 3000 argument indicates ISCompare to shift the beginning and end of the blast hits for group II introns in 3000 basepairs. This allows ISCompare to get flanks that will not contain the ISs halfs located on both ends of the Group II introns.

Guideline for understanding the results

6 The main output of ISCompare is the FinalResults.csv table. In addition, the tables QueryIS.csv, RefIS.csv, ConsecutiveIS.csv and two optional files, SLIS.csv (-rs) and graphicReport.pdf (-p), are produced.
The FinalResults.csv table contains all the information related to the DLIS found on the query and reference genomes, their genomic coordinates and annotation. All the analysed ISs are classified in 8 selfexplicative categories:

1. DLIS
2. Verify manually, possible false positive caused by consecutive ISs
3. Verify manually, Possible false positive produced by consecutive IS [2 RAFs found, only one with an IS]
4. Verify manually, possible false positive produced by repeated sequences [Different QIFs match with same region]
5. Verify manually, possible false positive produced by QIF match on scaffold end
6. Discarded for analysis. There were multiple blastn hits, and left and right pairs could not be determined
7. Discarded for analysis, blastn hits with low query coverage
8. Not found by Blastn.'

Of these, the DLIS category is the only one with a high confidence. All the other categories are reported, because their

Citation: Ezequiel G. Mogro, Nicolas Ambrosis, Mauricio Lozano (02/26/2021). Finding insertion sequence mobilization events with ISCompare.. https://dx.doi.org/10.17504/protocols.io.bst6nere

manual inspection could help improve the sensitivity of the method.

The ***FinalResultsTable.csv*** has the following fields:

Fields related to IS location
ISID: Name of the detected IS (Extracted from the headers of the ISDatabase.fasta file)
Isstart / Isend: genomic location of the IS in the query or reference genome.
Query.ID1: Accession number of the query sequence (Different scaffolds or replicons present a different ID) for the first/left IS flank.
Start1 / End1: genomic location of the first/left IS flanks
Fields related to flank locations
Query.ID2: Accession number of the query sequence (Different scaffolds or replicons present a different ID) for the second/right flank (Only when the two IS flanks are mapped in two different scaffolds)
Start2 / End2: genomic location of the second/right IS flanks
REF.ID1: Accession number of the reference sequence (Different scaffolds or replicons present a different ID) for the first/left IS flank.
REF.Start1 / REF.End1: genomic location of the first/left IS flanks
REF.ID2: Accession number of the query reference (Different scaffolds or replicons present a different ID) for the second/right IS flank.
REF.Start2 / REF.End2: genomic location of the second/right IS flanks
Fields related to DLIS description
Description: indicates whether the DLIS was detected in the query or reference genome
Observations: DLIS category [1 to 8, as previously described]
IS_Match_Type: complete IS hit / partial IS hit
Fields related to flanks annotation
QUERY_Flank1.locus_tag: locus tag of the genes present in the left IS flank in the query genome.
QUERY_Flank1.Product: description of the protein product for the genes in the left IS flank in the query genome.
QUERY_Flank2.locus_tag: locus tag of the genes present in the right IS flank in the query genome.
QUERY_Flank2.Product: description of the protein product for the genes in the right IS flank in the query genome.
REF_Flank1.locus_tag:locus tag of the genes present in the left IS flank in the reference genome.
REF.Product_Flank1: description of the protein product for the genes in the left IS flank in the reference genome.
REF_Flank2.locus_tag: locus tag of the genes present in the right IS flank in the reference genome.
REF.Product_Flank2: description of the protein product for the genes in the right IS flank in the reference genome.
Scaffold_Size: size of the scaffold analysed.

Using the -p option, the information of the main table is used to generate a PDF document containing, for each of the ISs found, a figure of the genomic context of the query genome with the IS location and the corresponding QIFs (Figure 1).

Citation: Ezequiel G. Mogro, Nicolas Ambrosis, Mauricio Lozano (02/26/2021). Finding insertion sequence mobilization events with ISCompare.. https://dx.doi.org/10.17504/protocols.io.bst6nere

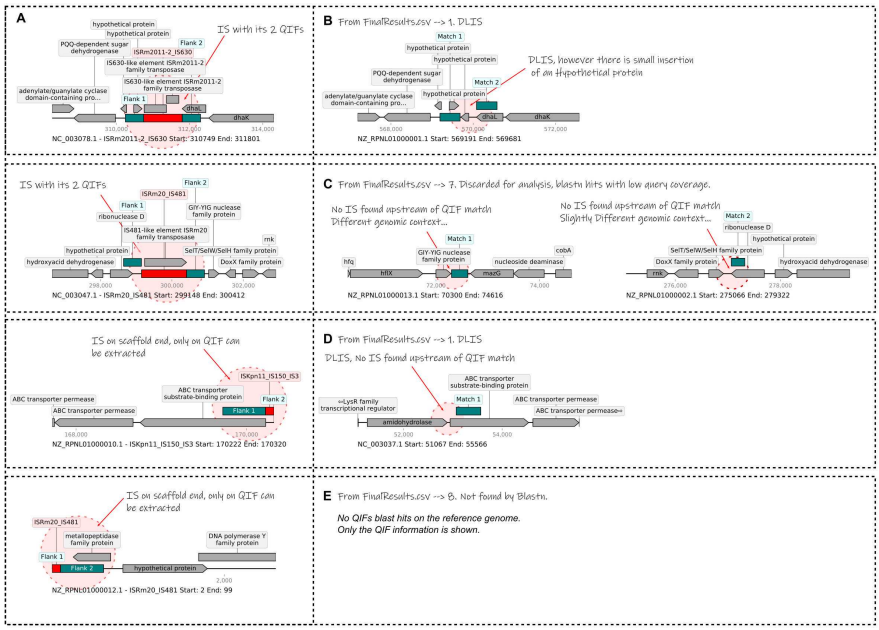

Figure 1. PDF graphic report. Examples of the figures included in the PDF report. A. The genomic context of the query genome with the found IS and its corresponding QIFs. B. The two QIFs were unequivocally matched to two reference genome positions separated by less than 10,000 basepairs. C. The two QIFs were unequivocally matched to two reference genome positions separated by more than 10,000 basepairs or located on different scaffolds or replicons. D. Only one QIF was unequivocally matched to a location in the reference genome. E. No QIF was unequivocally matched to a location in the reference genome.

In addition, the context of the reference genome corresponding to the matched QIFs is shown. Several cases may arise: first, if the two QIFs were unequivocally matched to two positions in the reference genome separated by less than 10,000 basepairs a single reference genome context plot will be drawn (Figure 1.B); second, if the two QIFs were unequivocally matched to two positions in the reference genome separated by more than 10,000 basepairs or located on different scaffolds or replicons, two reference genome context plot will be drawn (Figure 1.C); third, if only one of the QIFs was unequivocally matched to a single position in the reference genome, a single reference genome context plot will be drawn (Figure 1.D); and last, if the QIFs were not matched to positions in the reference genome, only the query genome context will be drawn (Figure 1.E).
These plots can be used to easily verify that the predicted DLIS are correct, and to try to manually assign DLISs that the program could not resolve (See Commentary, Critical Parameters and troubleshooting).

Critical Parameters and troubleshooting.

7 ISCompare works very good for the pairwise comparison of complete genomes of different strains from the same bacterial specie, achieving a high sensitivity and precision for the detection of DLISs. However, in the case of draft assemblies, the sensitivity could be significantly lower, in special for highly fragmented assemblies. In those cases a manual inspection of the graphic report is recommended, since it could improve the results. The results of a first ISCompare run could show many ISs under the 'verify manually' or 'discarded' categories, which are related to the presence of consecutive IS, repeated sequences or cases where there were no significant blast hits. If there were many consecutive ISs or repeated sequences an ISCompare run in the shift mode is recommended as a complementary analysis. The presence of many ISs in the 'discarded' category in general means that the analysed genomes are too different.
Although few, false positive predictions could arise as a result of differences in the genomic context. On such cases, both QIFs could match correctly to positions in the reference genome that are close together and do not have ISs, but which presents differences in their distant genomic context.

Advanced parameters

8 Although ISCompare arguments were adjusted to obtain the best results, there are several options to fine tune the analysis. A description of all the settable options is displayed with the -h option and explained on the ISCompare github webpage.

The most relevant options that can be fine tuned are:

Citation: Ezequiel G. Mogro, Nicolas Ambrosis, Mauricio Lozano (02/26/2021). Finding insertion sequence mobilization events with ISCompare.. https://dx.doi.org/10.17504/protocols.io.bst6nere

-E evalue cutoff for Blast searches
-s the Query ISs flank length to extract (QIF length) in basepairs. By default is set to 500 basepairs, however in some cases increasing this value can achieve better results.
-s2 Reference genome Anchor Flank length (RAF length). By default this value is set to 500 basepairs. Bigger values could increment the number of false negatives. In general a value smaller or equal to the QIF length is recommended.
-S Shift mode. Expands the IS blast hit coordinates in a defined number of nucleotides. This mode is recommended when the analysed genomes contain high density of insertion sequences, or to correctly identify DLISs which are adjacent to multicopy genes. The Shift value must be greater than the length of the repeated sequence.

Citation: Ezequiel G. Mogro, Nicolas Ambrosis, Mauricio Lozano (02/26/2021). Finding insertion sequence mobilization events with ISCompare.. https://dx.doi.org/10.17504/protocols.io.bst6nere