ERVsearch

ERVsearch is a ruffus pipeline for identification of endogenous retrovirus like regions in a host genome.

Introduction

ERVsearch provides a comprehensive screen for endogenous retrovirus (ERV) like regions in any genome. The genome needs to be available as a single FASTA file. Reference genome sequences are available from http://www.ensembl.org/info/about/species.html and https://genome-euro.ucsc.edu/index.html. De novo assemblies can also be used.

ERV search will take this genome and output: A table showing regions of the genome related to each of the major retroviral genes (gag, pol and env). This table shows:

- ID, chromosome, start position, end position and length of the ERV region
- The name and source of the most similar previously known retrovirus sequence and its similarity to the region
- The start position, end position and length of the longest open reading frame (ORF) in the region

Summary plots for each gene, consisting of:

- A bar plot showing the number of genes per chromosome
- A histogram of the ORF length
- A pie chart showing distribution between retroviral genera
- A bar plot showing the frequencies of regions related to different known retroviruses

Prerequisites

The pipeline is currently available for Unix systems only.

The ERVsearch pipeline requires the following freely available software.

Python 2 with the following packages:

- ruffus - https://pypi.python.org/pypi/ruffus
- numpy - https://pypi.python.org/pypi/numpy
- pandas - https://pypi.python.org/pypi/pandas
- ete2 - https://pypi.python.org/pypi/ete2
- matplotlib - https://pypi.python.org/pypi/matplotlib

All packages are available via pip and easy_install

- Exonerate: http://www.ebi.ac.uk/about/vertebrate-genomics/software/exonerate
- Bedtools: https://github.com/arq5x/bedtools2
- Samtools: https://sourceforge.net/projects/samtools/files/
- Usearch: http://www.drive5.com/usearch/download.html
- Emboss: http://emboss.sourceforge.net/download/#Stable/
- Mafft: http://mafft.cbrc.jp/alignment/software/linux.html
- FastTree: http://meta.microbesonline.org/fasttree/#Install

Quick Start

After cloning the repository, the program can be used as is (with the above prerequisites installed).

1. Make a copy of the pipeline.ini file (src/pipeline_ERVs/pipeline.ini) in your working directory (the directory in which would would like to store the output).
2. Download a local copy of your genome of interest as a single fasta file. Check if the genome is assembled into chromosomes, scaffolds or contigs.
3. Edit this copy to configure the pipeline for your computer:

   Change the working directory to the absolute path to the working directory.

   Add the name of your genome and the absolute path to the directory where you have saved it

   e.g. hg19.fa saved in /home/myname/genome/hg19.fa would require the following options:

   ```
   genome=hg19
   genome_directory=/home/myname/genome
   ```

   If your genome is assembled into chromosomes, has_chroms should be 1, otherwise it should be 0.

   Change the paths to the ERV input files to the directories in which you have cloned the repository.

   e.g if you cloned to /home/ERVsearch then paths would be

   ```
   path_to_refs=/home/ERVsearch/ERV_db/all_ERVs.fasta
   path_to_phyloseqs=/home/ERVsearch/phylogenies
   sequencedir=/home/ERVsearch/ERV_db
   ```

   Change the paths to the required software.

4. Run the pipeline in the working directory as:
5. `python /path_to_ERVsearch/ERVsearch/src/pipeline_ERVs.py -v`

Inputs

The pipeline requires a set of query sequence files in order to recognise ERV-like regions.

These will be referred to as:

ERV_Amino_Acid_DB - Three fasta file named gags.fa, pols.fa and envs.fa provided with the repository in the ERV_db directory.

All_ERVs_Fasta - A fasta file (all_ERVs.fasta) of nucleic acid sequences of known retroviruses provided with the repository in the ERV_db directory.

Reference_Phylogenies - Known retroviruses have been grouped according to sequence similarity and previous work, these groups are stored in the fasta files in the provided phylogenies directory. Files with the suffix _strays.fasta are mixed sequences from very small groups.

Summary_Phylogenies - Small subsets of known retroviruses selected for each gene and genus of retrovirus to provide an overview of the diversity within the group. These are also stored in the phylogenies directory, as gene_genus.fasta.

Outgroups_tsv - Tab-delimited file provided in the phylogenies directory listing an appropriate outgroup when building a phylogeny for each gene and genus.

Pipeline Functions

The pipeline is designed to be run from beginning to end, however it is possible to run functions individually or run subsets of functions. As a ruffus pipeline, standard ruffus command line syntax can be used to do this, as described at http://www.ruffus.org.uk/tutorials/new_tutorial/command_line.html. If the pipeline fails, delete output from the most recent step and it will automatically recommence from this step.

genomeToChroms

- Splits the host genome provided by the user into one fasta file for each chromosome.
- If the genome is not assembled into chromosomes, this is specified in pipeline.ini. The pipeline.ini parameter nchroms is then used - the contigs or scaffolds are concatenated into nchroms fasta files. This input type is allows much quicker screening with Exonerate.
- This function generates a series of fasta files which are stored in the host_chromosomes directory.

indexChroms

- Indexes all chromosomes in the host_chromosomes directory (or chromosome constructs generated by genomeToChroms) using Samtools faidx, for fast sequence retrieval later, and saves the index files

runandParseExonerate

- Runs the protein2dna algorithm in the Exonerate software package with the host chromosomes in host_chromosomes as target sequences and the ERV_Amino_Acid_DB fasta files as query sequences.
- Output is parsed to remove sequences shorter than the "overlap" parameter in pipeline.ini and sequences containing introns and combined into a tab-delimited file
- The raw output of Exonerate is stored in the raw_exonerate_output directory.
- This step is carried out with low stringency as results are later filtered using UBLAST and Exonerate (in the runUBLASTCheck and makeGroups steps).
- The parsed output is stored in the parsed_exonerate_output directory as gags_results.tsv, pols_results.tsv and envs_results.tsv.
- A bed file (https://genome.ucsc.edu/FAQ/FAQformat.html#format1) is also generated corresponding to the regions in each parsed output file - these are stored in parsed_exonerate_output as gags.bed, pols.bed, envs.bed.

mergeOverlaps

- Overlapping regions of the genome detected by Exonerate with similarity to the same retroviral gene are merged into single regions. This is performed using bedtools merge on the bed files output by runAndParseExonerate.
- Merged bed files are stored in parsed_exonerate_output as gags_merged.bed, pols_merged.bed and envs_merged.bed.

makeFasta

- Fasta files are generated containing the sequences of the merged regions of the genome identified using mergeOverlaps.
- These are extracted from the host chromosomes using Samtools.
- The output files are stored in parsed_exonerate_output as gags.fa, pols.fa and envs.fa.

makeUBLASTDb

- USEARCH requires an indexed database of query sequences to run.
- This function generates this database for the three ERV_Amino_Acid_DB fasta files.

runUBLASTCheck

- ERV regions in the fasta files generated by makeFasta are compared to the ERV_Amino_Acid_DB files for a second time, this time using USEARCH.
- This allows sequences with low similarity to known ERVs to be filtered out. Similarity thresholds can be set in the pipeline.ini file (usearch_id, min_hit_length and usearch_coverage).
- The output file is a fasta file of sequences with high similarity to known retroviruses in the ERV_Amino_Acid_DB. These are saved in parsed_exonerate_output as gags_filtered.fa, pols_filtered.fa and envs_filtered.fa.
- Raw output is also saved in parsed_exonerate_output as gags_alignments.txt, pols_alignments.txt and envs_alignments.txt.

findBest

- Runs the exonerate ungapped algorithm with each ERV region in the fasta files generated by makeFasta as queries and the All_ERVs_Fasta fasta file as a target, to detect which known retrovirus is most similar to each newly identified ERV region.
- All_ERVs_Fasta contains nucleic acid sequences for many known endogenous and exogenous retroviruses
- The raw output is saved in parsed_exonerate_output as gags_table_matches.tsv, pols_table_matches.tsv and envs_table_matches.tsv.
- Regions with no significant similarity to a known retrovirus are filtered out.
- The most similar known retrovirus to each of the ERV regions is identified. This result is saved as gags_table_bestmatches.tsv, pols_table_bestmatches.tsv and envs_table_bestmatches.tsv in the parsed_exonerate_output directory.

ORFs

- Finds the longest open reading frame in each of the ERV regions in the output table
- This analysis is performed using EMBOSS getorfs
- Raw getorfs output is saved in parsed_exonerate_output as gags_table_orfs.fa, pols_table_orfs.fa and envs_table_orfs.fa.
- The start, end, length and sequence of each ORF are added to the output tables and saved in parsed_exonerate_output as gags_table_orfs.tsv, pols_table_orfs.tsv and envs_table_orfs.tsv.

makeGroups

- The retroviruses in All_ERVs_Fasta have been classified into groups based on sequence similarity.
- Each group is named after a single representative ERV.
- The newly identified ERV regions are classified into the same groups based on the output of the findBest function.
- Each region is assigned to the same group as the retrovirus sequence it was found to be most similar to by findBest.
- The assigned group is added to the output tables, these are saved as gags_table_groups.tsv, pols_table_groups.tsv and envs_table_groups.tsv.

makePhyloFastas

- For each of the retrovirus groups used in makeGroups, a fasta file is available in the phylogenies directory containing a set of related sequences to allow a finer classification of sequences in the group.
- For each group generated by makeGroups, a fasta file is built combining the sequences in the group with the fasta file in Reference_Phylogenies.
- If groups are very large (more than 40 sequences) a random sample of 20 sequences is used to represent the group.
- If combined groups of ERV regions and known retroviruses are very small, the Summary_Phylogenies sequences for the appropriate gene and genus are added to allow a phylogeny to be built.
- Groups are aligned using the MAFFT fftns algorithm.
- A tree is built for each group using the FastTree algorithm, using the -gtr and -nt options.
- An image of each tree is also generated, using the ete2 python package.
- The outputs are saved in the "fastas" directory and "trees" directory under the same of one known retrovirus representing the group . Fasta files are saved in fastas as .fasta, aligned fastas as _ali.fasta. Newick formatted trees are saved in the trees directory as .tre, png images in this directory as .png.

PhyloFastasDone

- Helper function for makePhyloFastas allowing ruffus to detect when it is complete.

makeRepFastas

- Clusters of newly identified sequences are identified in the trees generated by makePhyloFastas.
- For each of these clusters, a single representative sequence is selected
- These are combined with the Summary_Phylogenies sequences to build a single tree representing each group. Fasta files are built containing the sequences,

these are aligned using MAFFT fftns, trees are built with FastTree -gtr -nt and images are generated using the ete2 package.
- The number of sequences in the group is also shown in the phylogeny
- Each group is given a unique ID, these are shown in the phylogenies and the sequences in the group are listed in a .txt file in the groups directory.
- If a sample of sequences was taken in makePhyloFastas (for very big groups), the number of sequences shown in the phylogeny is corrected to account for this. In these cases lists of sequences are not output to the groups directory.
- Fasta files of the sequences are saved in the summary_fastas directory, newick formatted trees and png images of trees are saved in the summary_trees directory.

summarise

- Statistics about the results are saved into the summary.tsv file in the working directory.
- A bar plot is generated showing how many ERV regions were identified on each chromosome for each gene, these are saved in the working directory as chromosome_counts_gag.png, chromosome_counts_pol.png and chromosome_counts_env.png.
- A histogram is generated showing the distribution of maximum ORF lengths for each gene. These are saved in the working directory as as orf_lengths_gag.png, orf_lengths_pol.png and orf_lengths_env.png.
- A pie chart is generated showing the distribution between retroviral genera identified in the host for each gene. These are saved in the working directory as genera_gag.png, genera_pol.png and genera_env.png.
- A bar chart is generated for each gene and genus showing the number of ERVs assigned to each group by makeGroups. These are saved in the working directory as groups_gene_genus.png.

Major Outputs

The main output files for this pipeline are:

A tab-delimited table for each gene with a row for each ERV region in the final filtered output, with the following columns: id - An ID for each ERV region match - The most similar retrovirus in All_ERVs_Fasta to the region score - The UBLAST score for the sequence and its match gene - gag, pol or env genus - retrovirus genus orf_start - the relative position of the start of the longest orf orf_end - the relative position of the end of the longest orf orf_len - the length of the longest orf orf_seq - the sequence of the longest orf group - the group of the sequence in the match column chr - chromosome on which the ERV region is found start - start position of the ERV region end - end position of the ERV region length - length of the ERV region

These are saved as: parsed_exonerate_output/gags_table_groups.tsv, parsed_exonerate_output/pols_table_groups.tsv, parsed_exonerate_output/envs_table_groups.tsv

A png file showing a phylogenetic tree summarising the ERVs found in the host for each gene and genus. These show a circle at each node, sized to represent the number of ERVs in this cluster. The ID of the group represented by the cluster is also shown. Circles are coloured by genus. These are saved as: summary_trees/gene_genus.png