FlyOrtho: a BLAST-based search tool for rapidly evolving orthologs among fruit flies

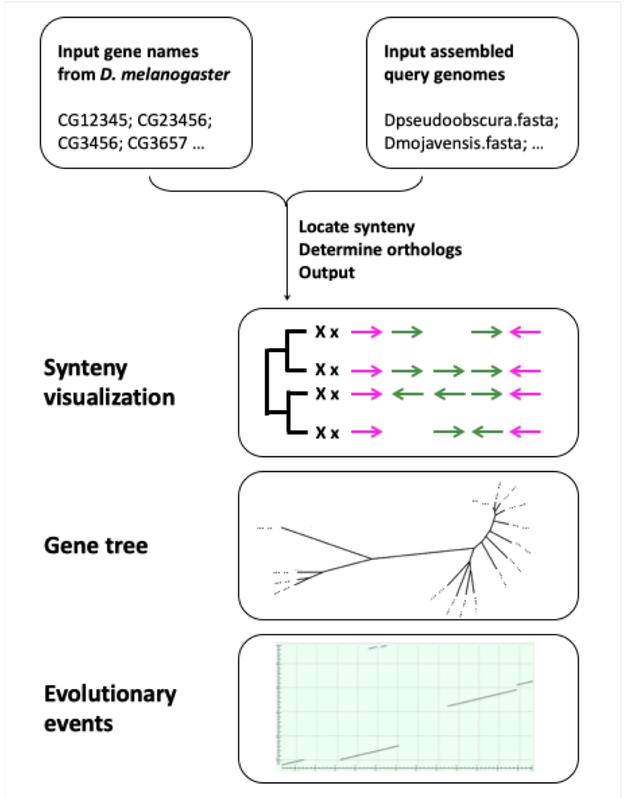
Zinan Wang [1], Arjun Krishnan [2], Henry Chung [1]

[1]Dept. of Entomology & Program of Ecology, Evolutionary Biology, and Behavior; [2]Dept. of Biochemistry and Molecular Biology

Overview

- Correct identification of gene orthologs across species is important to study the evolution of gene functions
- Genes are rapidly evolving; complicated evolutionary events could result in false prediction of gene orthologs using sequence similarity[1]
- Synteny, defined as conserved collocation of genomic elements, is important to determine orthologous genes[2]
- Contemporary synteny-search methods compare whole genome data among species and output large scale of sequences which are predicted being orthologous[e.g., 3-5]. However, these methods are not helpful to determine orthologs of single gene due to low efficiency and occupation of unnecessary computational resources. In addition, these complicated tools constrain the uses from molecular biologists with little computational skills
- In this study, I present a model-species-centric pipeline which integrates BLAST tools and synteny search based on protein coding sequences in assembled genomes regardless of annotation quality. The pipeline requires four major steps to accomplish.

Graphic Abstract



Step 1. To build database for synteny search

A database was built for rapidly extracting gene sequences from *D. melanogaster* genome. This database includes two tables for indexing positions and two files containing genome sequences and protein coding sequences of all genes. Once a targeted gene being input, this database can be used to locate this gene and other genes surrounding it. The protein coding sequences of located genes will be extracted for further steps.

Table 1.

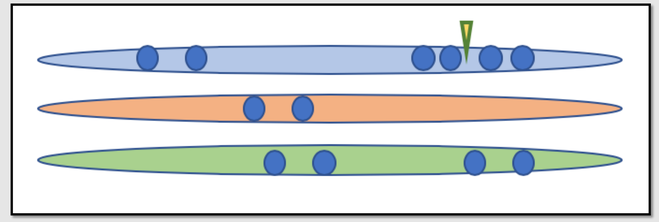
Gene ID	Gene_name	Position	# of CDS

Table 2.

CDS_name	CDS_position	Gene_name

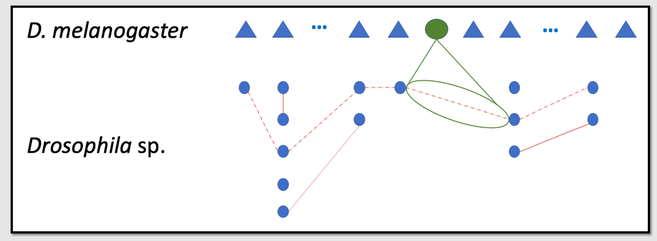
Step 2. To perform tBLASTn for flanking genes

The protein coding sequences are more conserved than regulatory regions[6], which is used to locate conserved synteny in query genomes. We use tBLASTn, a BLAST tool using amino acid sequences to search the protein coding regions in query genomes, to locate flanking genes in query genomes. tBLASTn may generate false positive outputs that distribute on different scaffolds/chromosome. For current step, we keep all outputs.



Step 3. To locate conserved microsynteny

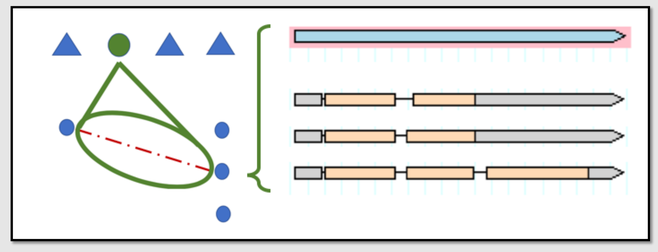
Output from each tBLASTn is transformed to a vertex with scaffold/chromosome and sequence positions being recorded. To locate the orthologous microsynteny for input gene, we connect the vertices for each scaffold/chromosome and find the shortest route with maximum number of vertices and minimum weight sum. Since conserved synteny confers orthologous elements (protein coding sequences), the shortest route here represents orthologous synteny in the query genome. The region between nearest flanking genes in the query genome is the microsynteny containing orthologs of input gene.



Step 4. To locate orthologs in the microsynteny

Once the microsynteny for the input gene has been determined, further operations focus on predicting the gene sequence and protein coding sequence of gene orthologs and detecting potential evolutionary events, including gene loss, gene duplication, and inversion. Detailed output needs more consideration.

The fatty acyl-CoA elongase genes (ELO gene family), a rapidly evolving gene family with complicated evolutionary history, will be used to benchmark this pipeline.



Conclusions

This class project is aiming at developing a model-species-centric pipeline for quickly finding ortholog(s) of one single gene from the genomes of non-model species. Generally, this pipeline uses protein coding sequences from the genome and annotation of model species as basic elements to locate microsynteny in the query genomes. With the help of microsynteny, the orthologous prediction will have much lower false positive rate.

References

[1]Fang et al 2010 Plos Computational Biology 6: e1000703
[2]Ghiurcuta et al 2014 Bioinformatics 12: i9-i18
[3]Baek et al 2016 Evolutionary Bioinformatics 12: S40009
[4]Nguyen et al 2017 Nucleic Acids Research 46:D816-D822
[5]Proost et al 2011 Nucleic Acids Research 40:e11
[6] Cai et al 2019 BMC Genomics 10: 623

[I thank Chen Wang (Dept. of Biology, Miami University, Oxford, Ohio) for providing instructions in Python for this project.]

wangzina@msu.edu