FUSTr: a tool to find gene Families Under Selection in Transcriptomes

**Background**

Elucidating patterns and processes involved in the adaptive evolution of genes and genomes of *organisms is fundamental to* developing an understanding of the vast phenotypic diversity found in nature. Recent advances in RNA-Seq technologies have played a pivotal role in expanding molecular evolution knowledge through the generation of an abundance of protein coding sequence data across all levels of biodiversity[1–4]. In non-modeleukaryotic study systems, transcriptomic experiments have become the de facto approach for functional genomics in lieu of whole genome resequencing. This is due largely in part to lower costs [5], better targeting of coding sequences [6], exploration of posttranscriptional modifications and differential gene expression [7,8]. This influx of transcriptomic data has resulted in an ever-expanding need for scalable tools capable of elucidating broad evolutionary patterns in large biodiversity datasets.

The genomic architectures of species across the tree of life is riddled with complexities. Numerous speciation events along with frequent whole genome duplications has given rise to a myriadofmultigene families that span a broad range of biochemical properties. Adaptive gene families receive Darwinian positive selection and are important in the evolutionary history of the organism. Grouping protein encoding genes into their respective families de novo has remained a difficult task computationally . This typically entails homology searches in large amino acid sequence similarity networks with graph partitioning algorithms to cluster coding sequences into transitive groups[12–14]*.* This is further complicated in eukaryotic transcriptome datasets that contain several isoforms via alternative splicing, which cannot be treated as phylogenetically independent homologs,[**??]**. Further analysis of these gene families is also non-trivial, as it requires multiple sequence alignment followed by phylogenetic inference [15–18]. Additional exploration of patterns of molecular evolution in these families is also computationally intensive, requiring robust phylogenetic analysis using codon substitution models with random or mixed effects likelihood methods in addition to MCMC Bayesian statistical frameworks, in order to determine patterns of pervasive diversifying selection or episodic lineage based diversifying selection [19, 20].

        Here we present FUSTr, a tool to address the aforementioned difficulties of characterizing molecular evolution in large biodiversity transcriptomic datasets in a pipeline capable of scaling to multicore high-performance computational facilities. FUStr can be used to characterizing selective regimes on homologous groups of phylogenetically independent coding sequences in transcriptomic datasets and has been verified using Arachnoserver and simulated datasets. The presented pipeline implements simplified user experience with minimized third-party dependencies, in an environment robust to breaking changes to maximize reproducibility over a long-term time scale.

**Implementation**

FUSTr *is written in python, and implements* eight widely used bioinformatics packages: snakemake, Transdecoder, BLAST, Silix, MAFFT, TrimAL, FastTree, PAML, and HYPHY. ~~Data preparation and filtration used to~~ *~~connect~~* ~~these programs is implemented as Python scripts.~~ These scripts and the command line arguments/parameters for each program is wrapped together using the workflow engine Snakemake, to ensure scalability, configurability and reproducibility. A custom anaconda environment has been created to allow for minimal amount of third-party software installation and configuration required to use FUSTr, due to all but two of its third-party dependencies being available in the Bioconda repositories. FUSTr takes as input assembled transcriptomes from any number of taxa in FASTA format. Header patterns are analyzed *to* auto-detect whether or not the given assembly includes isoforms. This is accomplished by comparing the header patterns to common assemblers, and detecting naming convention redundancies commonly used in isoform designations. *This allows FUSTr to support assemblies from a variety of sources*. Open reading frames (ORFs) are then *detected* using likelihood tests of all six reading frames *with* Transdecoder. *This facilitates further analysis, requiring data to be at the codon level.* If thedata contains different isoforms of the same gene, at this point only the isoform with the longest isoform is kept for further analysis to ensure phylogenetic independence. Homology searches are then implemented *via* all against all BLAST of ORFs translated to amino acid sequences, *with an E-value cutoff of @@@.* This *homology network* is then grouped into constituent gene families using transitive clustering with SiliX. Multiple sequence alignments of the amino acid sequences of each family are then generated using MAFFT. Spurious columns in alignments are removed with Trimal. Using the original multiple sequence protein alignment, an *approximate maximum liklihood* phylogeny is inferred with FastTree. Codon multiple sequence alignments are reverse translated from the *protein alignment. Tests of*  pervasive positive selection and episodic diversification are done using PAML and HYPHY with the codon alignments and *tree.* The final output is a summary file describing what gene families were detected , and which ones are under strong selection, and what sites and what branches. FUSTr is freely available under a GNU license and can be downloaded at <https://github.com/tijeco/Fuster>

**Validation**

**We tested** FUSTr with curated spider toxin proteins from ArachnoServer This allowed for verification of the proper *de novo* gene family reconstruction as well as *finding families already known to be under strong selection from literature.* which has a stringent *protein family curation* and contains several protein families under strong selection. Simulations were also used to validate FUSTr. to validate homology searches and to find families under selection previously studied. We recovered ### gene families from Arachnoserver XX% of which matched perfectly with Arachnoserver’s naming convention. Of those gene families recovered, 2 were identified to be under strong positive selection. To further demonstrate FUSTr’s abilities to properly identify simulations …..

**Conclusions**

FUStr has been demonstrated to be a useful tool for finding groups of functionally relavant prteins in biological databases, and has also been determined to be robust to highly diverged simulated data under varying selective regimes.

In order to further increase the reproducibility, third party dependencies has been reduced from nine to only three, all necessary dependencies have been wrapped up in an Anaconda environment.

**Pre-filter data**

Input data consists of assembled transcriptome fasta files, nucleotide data. These files are first filtered to remove sequences just containing Ns as well as to removing any haphazard text found in sequences (that may be artifacts of previous assembling procedures) to ensure proper downstream analysis. Header patterns in the inputs are simulteneosly autodetected to sort out all unique and redundant identifiers for downstream isofom filtration.

|  |  |  |
| --- | --- | --- |
| gene | AOXIE | id=a |
| gene | 111 | id=b |
| gene | 2342 | id=a |
| gene | 2342 | id=b |
| gene | 2342 | id=c |

* Algorithms, bitches and money, layout of typical gene and isoform header patterns

**Predict coding sequences**

Coding sequences are determined using Transdecoder, which predicts orfs using all 6 possible reading frames, keeping the best orf per transcript. Genes containing several isoforms will only keep longest isoform for downstream stuff. protein sequences, stuff. Keep only longest isoform.

**Group by homology**

Sequence homology networks are determined using an all against all BLAST search of the amino acid sequences. Sequences are grouped into homologous groups using Silix, only adding sequences to a group that have at least ### sequence coverage, ### sequence identity, &&&& other stuff. All against all blast. Silix. Families.

**Multiple-sequence alignment and phylogenetic inference**

Homologous groups containing at least 15 amino acid sequences are aligned using Mafft (accurate and fast, auto), and then trimmed using trimal. Fastree is used to infer phylogenetic tree (accurate and fast) Mafft, trimal, codon masking. Codon sequences are masked over amino acid alignment.

**Pervasive positive selection**

Codon alignments. CODEML, pervasive positive selection.Codeml and FUBAR??

~~There are several established phylogenetic frameworks to then test the families for various forms of selection utilizing  rates of nonsysnonymous to synonymous substitution. Determining selective regimes of specific amino acid~~ *~~residue~~* ~~sites that~~ *~~may be adaptive~~* ~~involves tests of pervasive positive selection within a gene family using either fixed or random effects likelihood models to help ellucidate specific amino acid residue sites undergoing diversifying or purifying selection (FUBAR, M8). Finding specific lineages that may have undergone~~ *~~adaptive surges~~* ~~(niche diffenertiation, sexual selection, predator prey arms races, novel innovations, lots of evolution terms) requires tests for episodically diversifying lineages (MEME,BUSTED,CodemlBranchSpecific) or evolution along specific branches in gene family…~~

**Episodic positive selection**

Hyphy/MEME, FUBAR, BUSTED

**Validation**

Simulations, GLOve??? Bacteria stuff (POTION), and arachnoserver (compare to young clade paper).

Results

**Simulations**

We did stuff, yeah.

**Empirical results**

Arachnoserver, toxinbase, bacteriashit, goldenstandard.

**Conclusions**

~~An influx of~~ *~~terabytes~~* ~~of transcriptomic data has resulted in an ever expanding need for scalable~~ tools capable of elucidating broad patterns of molecular evolution within the genomic architecture of taxa spanning throughout the tree of life.