8-21-2017

EnTAP: Software to Improve the Quality and Functional Annotation of De Novo Assembled Non Model Eukaryotic Transcriptomes

Alexander Hart
alexander.hart@uconn.edu

EnTAP: Software to Improve the Quality and Functional Annotation of *De Novo* Assembled Non-Model Eukaryotic Transcriptomes

Alexander J. Hart

B.S. University of Connecticut 2017

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

At the

University of Connecticut

2017

Approval Page

Master of Science Thesis

EnTAP: Software to Improve the Quality and Functional Annotation of *De Novo* Assembled Non-Model Eukaryotic Transcriptomes

Presented by
Alexander J. Hart, B.S.

Major Adviser _____
Dr. Jill Wegrzyn

Major Adviser _____
Dr. Kevin Brown

Major Adviser _____
Dr. Rachel O'Neill

University of Connecticut
2017

Acknowledgements

I would first like express my sincere appreciation for my advisor, Dr. Jill Wegrzyn, for being supportive at every step of the way and providing an invaluable source of information and guidance throughout my studies. I am truly privileged to have such a thoughtful and enthusiastic advisor who always took the time to help. I'd also like to thank my thesis committee members, Dr. Kevin Brown, and Dr. Rachel O'Neill who have provided valuable guidance and support.

I owe a great deal of thanks towards the many collaborators on my thesis including, Dr. Claudio Casola, Cera Fisher, Dr. Rachel O'Neill, Erik Visser, and Jeff Mitton who have graciously provided data for the development of my thesis.

I truly appreciate the invaluable help of the members of the Plant Computational Genomics Lab at the University of Connecticut who have worked tirelessly with me in the development process providing crucial suggestions and considerations. I would also like to acknowledge the vital help and support from everyone at the Computation Biology Core.

I would like to thank my family, as without them, I would not be the person I am today. They have been an unwavering source of encouragement and support. Finally, I would like to thank my friends that I have had the pleasure to share time with. Be it be past or present, I am grateful and appreciative that these individuals have played a part.

Table of Contents

List of Figures

List of Tables

1 Introduction:

1.1 Transcriptome Background

Transcriptomes provide an opportunity to assess gene expression at a specific point in time or in response to stimuli, allowing discovery of overrepresented or underrepresented genes [1]. The transcriptome is comprised of the mRNA (messenger ribonucleic acid) transcribed from an organism's genetic code. A single gene can range in length from several hundred to several thousand base pairs, with eukaryotes having total gene numbers ranging from several thousand to tens of thousands [2, 3]. It is but one of several variants of RNA, including, but not limited to, tRNA (transfer RNA) and rRNA (ribosomal RNA), each involved in various stages of transcription and translation. mRNA acts as an intermediary between the genome and protein creation; it is transcribed from the genome and translated into chains of amino acids following a three-base pair to amino acid rule [4]. The vast majority of eukaryotic genomes are typically non-coding. In humans, the gene space represents less than 5% of the genome [5]. The sampled transcriptome is generally even smaller since not all of these genes are actively transcribed in every cell.

Within a single organism, the expressed gene space varies among cells, tissues, time of day, and in response to a variety of abiotic and biotic factors. Assessment of what is known as differential expression allows one to interrogate the differences between individuals, tissues, or in response to specific stimuli. This approach has been widely applied in microbial and eukaryotic systems. Studies assessing differential expression in transcriptomes have been applied to thousands of plants, mammals, fish, insects, fungi, and bacteria in order to gain valuable insight into what genes are influencing aspects of development, defense, and more [6-8]. Differences in

gene expression are controlled through several cellular factors, such as: promoter influence, alternative splicing, and other transcriptional regulators [9]. Analysis of the transcriptome is realized through probe-based methods (Microarrays) and/or direct sequencing (Expressed Sequence Tags and RNA-Seq). Presently, RNA-Seq is the most common approach and has this has been made accessible with the advent of Next-Generation Sequencing (NGS).

1.2 Sequencing Techniques

The most popular technique for evaluating expression was via ESTs (Expressed Sequence Tags) or transcripts typically assessed via Sanger's chain termination method implemented through selective incorporation of chain-terminating dideoxynucleotides (ddNTPs). Sanger sequencing was one of the most widely used techniques, for both genomes and transcriptomes, for over twenty years [10]. This method, while accurate, required three separate reactions with ddNTPs for each nucleotide as well as a separate DNA strand for each base in the sequence. Several incremental improvements were made to Sanger sequencing over the years, including fluorescent tagging to improve base identification [10, 11].

NGS platforms have provided the means to rapidly sequence an incredible amount of data, in parallel, with costs decreasing year after year. NGS platforms today encompass both short and long read technologies. RNA-Seq describes highly parallel assessments of transcriptomes with the use of short reads (generally between 36 and 300 bp) [12]. The most widely used platform, implemented via *sequencing by synthesis* (SBS), is the cyclic reversible termination (CRT) technology developed by Illumina (Table 1). The popular Illumina HiSeq 3000/4000 systems produce short reads (no longer than 300bp or 150bp PE) with a throughput

of 650-750 Gb per flowcell [12]. Illumina achieves these results through a sequencing by synthesis approach that incorporates fluorescently labelled markers as Deoxynucleotide Triphosphates (dNTPs) that act as a chain-terminator (through prevention of a phosphodiester bond forming with the 3' OH group). Once incorporated, the added base is sequenced and the blocker removed. This process generates short reads, either paired-end or single-end, up to 300bp in length [13]. Paired-end reads are generated from sequencing the DNA from both sides, forward and reverse, while a single-end read is the result of unidirectional sequencing [12].

Table 1: Common NGS Technologies (Illumina) for RNA-Seq

Sequencer	Read length (bp)	Throughput	Reads	Runtime	Error Profile	Cost per Gb
Illumina MiSeq v3	75 (PE)	3.3-3.8 Gb	44-50 M	21-56 h	0.1%	$250
	300 (PE)	12.2-15 Gb				$110
Illumina NextSeq 500/550 Mid-Output	75 (PE)	16.25-20.5 Gb	Up to 260 M (PE)	15 h	<1%	$42
	150 (PE)	32.5-39 Gb		26 h		$40
Illumina HiSeq 3000/4000	50 (SE)	105-125 Gb	2.5 B (SE)	1-3.5 d	0.1%	$50
	75 (PE)	325-375 Gb				$31
	150 (PE)	650-750 Gb				$22

*SE, refers to a single end read; PE, refers to a paired end read; AT, refers to Adenine and Thymine nucleotide bases [12, 14, 15]

1.3 Transcriptome Assembly

To date, approximately 5,500 reference genomes have been completed for eukaryotic organisms, while an estimated 7.4 to 10 million species remain uncharacterized [16, 17]. In model systems with a complete reference genome, RNA-Seq analysis involves the alignment of single or paired-end short reads derived from RNA libraries against an annotated genome. If the appropriate sequence depth and mapping rate are achieved for each RNA library, the reads can be quantified at exonic regions and compared across libraries. In the absence of a genome reference, paired-end data serves as input to a *de novo* assembler. This assembler will attempt to generate contigs (genes or gene fragments) from the short reads [1]. Non-model organisms are defined here as those without a high-quality reference genome and the associated genomic resources. Many non-model eukaryotic organisms do not have a close (phylogenetic) relative to serve as a reference genome, thus *de novo* assembly is the only option [18].

Assembling a transcriptome via NGS derived short reads from RNA libraries introduces both computational challenges and multiple sources of error. Some of this error is a result of the sequencing platform itself. As noted in Table 1, each platform is associated with its own error profile. While deep sequencing can assist with resolving some of the bias and error profiles, this remains a challenge when working with shorter reads (< 300bp) [1, 12, 13]. Compared to the genome, the transcriptome is much more complex, with coverage variation resulting from variable expression, splice variants, and chimeric-assemblies [13]. In addition, RNA is much more challenging to work with as it degrades rapidly. RNA samples may also include (in the case of an eukaryote target), material from microbial, fungal, or other closely associated organisms [19, 20]. The library construction and sequence strategy for the RNA libraries may also impact the

assembly. Pooling multiple individuals into a single library (rather than multiplexing each individual in a lane) is often done when sufficient RNA is scarce from a single sample. The process of pooling different genotypes generally results in increased heterozygosity and more fragmented assemblies [21].

In addition to the limitations imposed by the RNA library sampling and library construction, sequencing platform, and biological constraints of transcriptomes - approaches to assemble NGS data also play a role. The most commonly used algorithm for *de novo* assembly of NGS sourced reads (genome or transcriptome) is based on de Bruijin graphs [22]. The implementation of this graph-based approach is the current solution to the deep sequencing available with NGS technologies. While traditional approaches (Sanger derived sequencing) utilize the more accurate, but much memory intensive 'overlap-layout-consensus' method, it is not scalable on the combination of read depth and existing hardware configurations [22]. Nearly all genome and transcriptome assemblers today utilize some form of the de Bruijin graph methodology, including the most popular assembler for transcriptomes, Trinity [18, 23, 24]. In the majority of implementations, the reads are separated into *k*-mers (strings of length *k*) and aligned against each other to create edges, or common bases between *k*-mers. This alignment is not a separate process, but a relationship found during *k*-mer construction. A *k*-mer can be seen as a node on the graph with linkages, or edges, between k-mers giving genomic relationships that allows for construction of the assembly [25]. This methodology is very sensitive to heterozygosity, sequencing errors, and polymorphisms which means the graph splits easily when encountering these. This often results in inflated numbers of "genes" in the final assemblies which result from

fragmented assemblies. A typical final assembly will include several complete (full-length) genes but also many that lack a full frame on one end (5’ or 3’) or both ends.

1.4 Functional Annotation

Transcriptome annotation is the process of functionally annotating assembled transcripts with a myriad of information such as sequence similarity, protein domain identification, gene family assignment, and Gene Ontology term assignment through comparison with several informative databases [26]. The object delivered following *de novo* assembly is simply a text file of sequences with arbitrary names generated by the assembler. The goal is to take that entire set of transcripts (or just those identified as differentially expressed) and determine their function. Before annotation, however, detection of the open reading frame (ORF) can be beneficial to the downstream analysis and annotation.

Open reading frames are determined by the presence of a “start” and “stop” amino acid codon within the transcript comprised of amino acids AUG and UAA, UAG, or UGA, respectively. These codons signal the beginning and ending of translation into proteins. However, complications can arise with the presence of untranslated regions (UTR) on both the 5’ and 3’ ends which is common in *de novo* transcriptome assembly. As a result of amino acids requiring three nucleotide bases in order to be translated, there exists six total reading frames from three in either direction of the strand [11]. Frame selection software attempts to remedy several of these issues while determining complete, internal, and partial genes from transcripts. Selection of a frame can be beneficial to the downstream annotation analysis as it provides a “more likely”

translation of the mRNA than one would find through a frame selected by a similarity searching software.

Gene identification is generally performed through similarity searching against one or more public protein repositories of experimentally or computationally derived sequences, maintained via Ensembl or NCBI (National Center for Biotechnology Information)[27]. With this, transcript sequences are aligned against the full database and generally subject to requirements of coverage, identity, and probability of a unique match often computed in terms of E-value. Several open-source tools are available for similarity searching between databases such as BLAST (Basic Local Alignment Search Tool), being among the most popular, DIAMOND (double index alignment of next-generation sequencing data), and RAPsearch2 [28-30]. A variety of methods are incorporated in order to increase sensitivity (lower rates of false positives) and increase speed. The most widely used method of hash table seeding is used within the aforementioned software. This involves splitting sequences into *k*-mers (smaller sequences of length *k*), or seeds, and storing these within a hash table that can be accessed through the *k*-mer [31]. A popular method incorporated to improve traditional seeding is the reduced alphabet alignment. This method attempts to maintain similar sensitivity (or at a minimal loss) while drastically increasing alignment speed. Moreover, integration of spaced seeds, or longer seeds with possible mismatches within its sequence, provide a higher sensitivity at a minimal cost in runtime [29, 31].

In non-model systems, it is likely that several (or even the majority) of assembled transcripts will not have strong alignment to an existing annotated protein in a database. Additional information can be obtained by identifying conserved regions within the amino acid sequence (protein domains) that have been characterized in existing databases. A protein

domain is a functional unit of a sequence that determines a certain part, or structure of a protein, such as the catalytic behavior [32]. These domains can be grouped into families which are essentially a hierarchy of protein inheritance, with the children inheriting certain characteristics from the parents. Several curated databases exist, including SMART, Pfam, Panther which characterize these motifs and their associated annotations [33-35]. Standalone applications that focus on optimizing the search for domains, such as InterProScan, can query multiple databases [36].

Domain assignment can be further leveraged through the use of orthologous gene families. Several curated sources exist for organismal families, including KOG, EggNOG, and OrthoDB [37-39]. These resources differ in orthologous group generation, curation, and querying strength. Curation of these families generally involves the identification of protein domains and the assignment of curated terms and associated pathways. The *ortholog conjecture* is a longstanding postulation within the phylogenetic community that states orthologous genes, or genes that are a result of speciation, are more functionally alike to paralogous genes, genes resulting from duplication [40]. This theory has been generally accepted throughout the years, however there is a significant lack of large-scale studies to accurately test the conjecture. A study was performed that disproved the theory in functionally mapping orthologs to gene ontology terms, while another determined this is an inaccurate means of testing and proved the ortholog conjecture using RNA-seq data [40, 41]. Nonetheless, it has remained a very important part of biological community as a means of functionally annotating non-model organisms that have more informative orthologs.

Gene Ontology (GO) terms have become a standard for characterizing proteins products into three categories: *molecular function*, *biological process*, and *cellular component* [26, 42, 43]. These terms can be derived from well characterized sequence similarity search results, independent protein domain assignments, and/or from orthologous gene families curated with these terms. The terms are always associated with a single sequence or domain and have a confidence score and source that reflects how that information was obtained for a given sequence [43]. Molecular function refers to specific molecular or chemical activity within the cell, such as *flavonoid 3'-monooxygenase activity*. The biological process refers to the biological result of a group of molecular functions, as seen in *regulation of adenosine receptor signaling pathway*. Lastly, the cellular component describes any structure within the cell such as a *nucleus* or *ribosome* that the protein product may be acting on. Standardizing the nomenclature of gene descriptions allows for a comprehensive view of categories of functions and processes impacted by differential expression [43].

The final step in the process of functional annotation often involves the assignment of differentially expressed genes (or all genes in the transcriptome) to a curated biological pathway. The combination of orthologous gene families, protein domains, and ontology terms can assist in recognizing single genes or groups of genes associated with databases hosting pathway information. Key resources for pathway assignment include KEGG (Kyoto Encyclopedia of Genes and Genomes), as well as Reactome and MetaCyc [44-46]. The KEGG database represents many biological systems interplaying with genomic and chemical information through their pathways and relationships through 16 independent databases [44]. Using this information, researchers

are able to obtain data on the relationships among the differentially expressed transcripts that is not evident from similarity search alone.

1.5 Current Challenges and Existing Solutions

There are several key challenges involved in the annotation of non-model eukaryotes that ranges from the error-prone assembly, the less representative databases used for comparison, and the usability of existing software packages. Before functional annotation can begin, the quality of the assembled nucleotide transcriptome, completeness and correct frame identification, must be examined as it can have drastic effects on the downstream annotation of the gene space. Resulting annotations may be based on an improper reading frame or on an artifact of the assembly. Estimates between *de novo* transcriptome assemblers vary considerably (by hundreds or thousands of genes) and the majority of tools for functional annotation do not consider pre-filtering assemblies for artifacts that may impede transcriptome characterization or differential gene expression analysis [23, 47, 48].

Non-model systems tend to have limited genomic resources and generally no reference genome. In many cases, they may belong to a genus or clade with very little genomic information. The farther removed they are from a well annotated reference, the more likely it is that similarity search approaches will not be able to provide an informative match. Mechanisms for evaluating phylogenetic relevance of a match as well as informativeness of the description is also lacking in existing software solutions. In addition, similarity searching is often the rate limiting step in annotating thousands of genes at a time. Traditional tools such as NCBI BLAST can take several days depending on the hardware available to complete a run. While faster and equally sensitive

tools currently exist, few are implemented into pipelines that benefit from multiple rounds of similarity search.

Contamination plays an important role in eukaryotic RNA libraries. Plant, animal, insect, and fish tissues are often in close association with bacteria, fungal, and viral RNA. The process of annotating these proteins and removing them from the reference transcriptome must typically be implemented through a manual and time consuming process by the user. It is rarely well integrated into the pipeline despite the prevalence [42].

Gene ontology terms are a desirable outcome when annotating full transcriptomes and sets of DE genes. When applied comprehensively, this information can be used to examine enrichment. Enrichment analysis allows one to assess which functions or processes appear to be up or down regulated among the libraries compared. In non-model systems, it can be challenging to achieve term assignment where close relatives with a full curated set of terms are lacking. Current functional pipelines do not leverage multiple sources of term assignment in order to improve this annotation. Doing so would allow for more robust and accurate enrichment comparisons. The implementation of classification and term assignment by orthologous genes or protein domains is gaining popularity but the execution is often incomplete.

The usability of annotation software is often an overlooked characteristic; however, it can play a major role in a user's software selection. Usability can be broken down into three main components: installation, ease and flexibility of execution, and interpretation of results. The installation of annotation software often comes with a variety of dependencies in order for the software to perform. This can create challenges for the user and barriers to use. Problematic installation can be observed in the integration of the Gene Ontology database which may require

a full local SQL database. Ease of use and allowing great flexibility in execution can often be challenging [43]. Whether it be a simple command, or a complicated multi-step process to annotate a transcriptome, ease of execution can be a major turning point for many pipelines. A user can get frustrated with software that has an enigmatic execution process or poor documentation. Clear documentation and test datasets are necessary to avoid frustrations during installation and execution [44]. In regards to flexibility, if a user is too confined to a set of databases or options, it will greatly reduce the utility of the software. Above all else, the results of the pipeline must be accurate, sufficient, and easy to understand.

The most popular full-scale pipelines for annotation include Trinotate, Blast2GO, and TRAPID [47-49]. Trinotate incorporates the software package Transdecoder as a means of frame selection as well as integration with protein domain databases such as Pfam, Gene Ontology term assignment, and pathway information [50]. It successfully oversteps a complicated SQL installation via SQLITE. Trinotate suffers from a complex, multi-step, execution that requires integration of the components by the user. It is also fairly slow due to reliance on BLAST over faster methods of similarity search, and a heavy reliance on specific databases (Swiss-Prot and Pfam) [29, 47]. These constraints limit the user in several categories, such ease of use and flexibility. TRAPID attempts to remedy some of these issues by providing a more efficient method of similarity searching via RAPSearch2 and an online interface to analyze datasets, removing the complications involved in installation [48]. Additionally, it provides many methods of visualization for the user and basic statistics on the final annotation. Blast2GO provides a GUI (graphical user interface) in a standalone software package with access to a variety of databases. Users are able to load their datasets and execute NCBI BLAST, InterProScan for protein domain assignment,

Gene Ontology term assignment and pathway assignment. Additionally, Blast2GO provides many methods of data visualization. Blast2GO suffers from a paid subscription service for access to a majority of its functionality as well as very slow similarity searching which is not accessible for users with larger transcriptome datasets. The lack of transcriptome frame selection incorporated into the pipeline creates additional challenges. None of the pipelines available today offer robust options for selection of optimal similarity search hits, such as: filters for taxonomic relationships, contaminant identifications, or selections based on sequence description informativeness.

EnTAP is an open-source annotation pipeline designed to remedy many limitations of existing software solutions and provide greater flexibility, ease of use, speed, and accuracy.

Table 2: Annotation Software Comparison

Metric	**Blast2GO Pro**	**Blast2GO Basic**	**Trinotate**	**EnTAP**	**TRAPID**
Open Source/Free Software		X	X	X	X
Command Line Integration	X		X	X	
Filtering Assembly via Short Read Alignment (Expression)	X			X	
Frame Selection				X	X
Custom Database Selection and Indexing	X		X	X	X
Fast and Sensitive NCBI BLAST Alternative				X	X
Selection of Optimal Hit From Several Databases				X	
Selection of Optimal Hit Based on Informativeness	X			X	
Contaminant Identification and Filtering				X	
Orthologous Gene Family Assignment			X	X	X
Gene Ontology term and pathway assignment sourced from Orthologous Genes				X	X
Provides Graphical User Interface	X	X	X		X

2 Methods

EnTAP was developed in the C/C++ languages utilizing C++11 features. General execution involves two files from the user, a required FASTA formatted transcriptome and an optional BAM or SAM formatted alignment file. The pipeline transitions between several states while maintaining and updating transcript information found at each state throughout the entire execution. Evaluation of the assembly is performed first, composed of two stages: transcript quantification and frame selection. From here, EnTAP transitions to the main annotation execution in similarity search against user-selected databases (up to three), taxonomic filtering and best-hit selection, and finally to orthologous gene family assignment as the basis for Gene Ontology term and pathway assignment. Prior to this execution however, EnTAP must be configured by the user.

The full documentation for EnTAP is available at http://EnTAP.readthedocs.io/en/latest/. The code for the current package release (0.5.6.1) is available on GitLab at https://gitlab.com/EnTAP/EnTAP. Several dependencies and libraries are required by EnTAP for full functionality including CMake (MakeFile generation), Boost C++ Libraries (serialization, user input parsing, parallelization), Python, Perl, pstreams (IOStream library for terminal commands), SQLITE (EggNOG database integration) and fast-cpp-csv-parser (TSV file format parsing) [51-56].

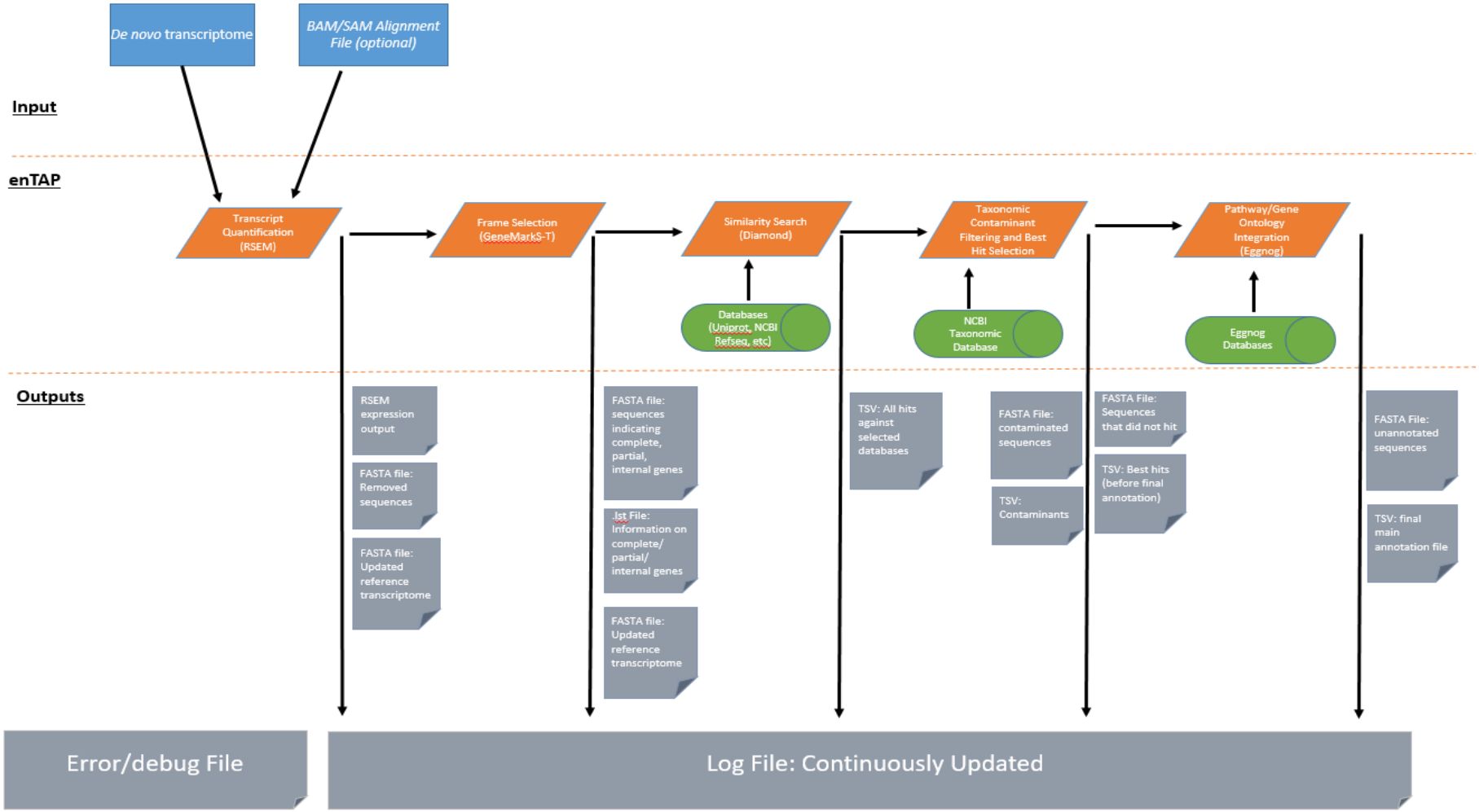

Figure 1: EnTAP Main Annotation Pipieline Overview

2.1 Configuration

The first stage of EnTAP, prior to execution of the main annotation pipeline, involves configuration of accompanying databases and features. This involves serialization of an NCBI taxonomic database, Gene Ontology term database, and indexing of user-selected databases for execution from DIAMOND. Serialization is performed through utilization of Boost libraries [49]. This is the only stage that requires an internet connection in order to download the required datasets.

2.1.1 NCBI Taxonomic Database

EnTAP's taxonomic database derives from the taxonomic database upheld and continually updated by NCBI, or the *Entrez Taxonomy Database*. This database is curated by NCBI taxonomists that upholds species information and phylogenetic classification on a multitude of organisms [57]. The database contains approximately 10% of the described eukaryotic species on the planet as well as almost all of the prokaryotic species analyzed, with unknown or underrepresented species maintaining informal, "placeholder", names within the database. The Taxonomy database has an entry for every organism that has an associated genetic record in any division of NCBI [57]. Incorporation of the taxonomic database permits contaminant filtering and improved selection of the most appropriate similarity search match.

During configuration of EnTAP, this taxonomic database is queried for taxonomic XML configured data on every entry in the database. From here, each query is filtered for NCBI taxonomic ID, species (or match in the database), and phylogenetic lineage. For an entry within the database such as *Homo sapiens*, lineage information would be recorded from "Eukaryota", "Opisthokonta", through the phylogenetic tree to *Homo sapiens*. The extracted information

(entry name, lineage, taxonomic ID) is then configured into an ordered map and serialized in an output file for re-reading into EnTAP during the main execution stage.

2.1.2 Gene Ontology Term Mapping

The software utilized to assign Gene Ontology information within EnTAP does not directly report complete GO descriptions or hierarchical level information (only unique IDs). To contend with this limitation, EnTAP provides the bridge between GO IDs and deeper term information provided from the Gene Ontology Consortium [43]. Additional information reported includes the Gene Ontology category the term was assigned (biological process, cellular component, or molecular function), the hierarchy level of the term, and the full-term description. Similar to the taxonomic database, this information is downloaded directly from the Gene Ontology Consortium, parsed, and arranged within a map of accession keys from GO term IDs. The map is then serialized into a file so it can be read back during the main execution pipeline.

2.1.3 Database Indexing

DIAMOND is included in EnTAP as a similarity search tool. As such, all databases selected by the user will need to be indexed, or formatted, for quick searching speeds. The user merely needs to enter the path of the database and it will be automatically configured for use by DIAMOND. Additional databases through providers such as NCBI can be downloaded in their FASTA format and indexed by Diamond during the configuration. This step should be re-run when a new version of the database(s) are available.

2.2 Evaluating the Assembly

Execution of EnTAP is based primarily around 8 enum states, INIT, EXPRESSION, FRAME_SELECTION, FILTER, SIMILARITY_SEARCH, SIMILARITY_PARSE, GENE_ONTOLOGY, and EXIT. Under default circumstances, the states will transition in the order as previously stated through bit shifts. However, the user has complete control over state execution if they would only like to execute certain parts of the pipeline. Utilizing the "--state" flag, the user can start and stop at certain locations in the pipeline, allowing for greater flexibility. If the user would like to execute similarity search to the end of execution they would flag "--state 4+" to indicate execution of stages four through eight. Alternatively, if they would like to solely run similarity searching, the user might flag "--state 4x," with an "x" denoting exiting of execution. Additionally, with the absence of an alignment file or an amino acid transcriptome, both the frame selection and expression filtering stages can easily be skipped.

The execution phase of the pipeline begins by evaluating the provided *de novo* transcriptome assembly. As noted previously, mis-assemblies and errors in sequencing can result in assembly artifacts. In an attempt to mitigate these issues and provide more accurate input for transcriptome annotation, frame selection and alignment of the reads back to the *de novo* assembled transcriptome are implemented in EnTAP. The detection of coding transcripts can help eliminate potential errors in assembly, trim the Untranslated Regions (UTRs), and remove transcripts where no coding regions were detected. Furthermore, the alignment of reads to the assembled transcriptome is an accepted approach to assess expression and coverage of transcripts and potentially eliminate assembly errors [25].

2.2.1 Frame Selection

Several methods of frame selection exist utilizing different methods of supervised and unsupervised learning. GeneMarkS-T developed at the Georgia Institute of Technology by Mark Borodovsky and Alex Lomsadze, specifically for eukaryotic transcripts, was selected for EnTAP [25]. The GeneMarkS-T algorithm utilizes iterative self-training and a hidden semi-Markov model. Initially, clusters are formed based upon guanine and cytosine (G + C) content in which iterative self-training through a hidden semi-Markov model predicts coding regions which ultimately converges to find the predicted gene. GeneMarkS-T is primarily designed for prediction of eukaryotic protein coding regions making it a viable option during transcriptome analysis. It functions well for short transcripts when compared with TransDecoder due to a lower base pair threshold in addition to overall higher rates of gene prediction [25, 50]. Moreover, GeneMarkS-T provides a means of self-training that is not present in competing software, and therefore does not rely on additional database input [25].

GeneMarkS-T execution is performed with the following command:

- gmst.pl -faa -fnn INPUT_TRANSCRIPTOME

Following successful execution of frame selection, EnTAP will begin to parse all GeneMarkS-T outputs and update an overall ordered hash map, SEQUENCE_MAP, that is initialized at the beginning of the EnTAP execution. This map is comprised of *QuerySequence* objects keyed to the query sequence IDs within the reference transcriptome. Throughout execution, SEQUENCE_MAP is passed by reference to each state to be continually updated with more information. After successful parsing of GeneMarkS-T output, each sequence is updated with a frame selection tag

(Partial 5 Prime, Partial 3 Prime, Internal, or Complete) as well as an amino acid sequence, if applicable. After successful completion of this stage, control switches to the EXPRESSION state.

Execution of the frame selection stage of EnTAP will provide the user with an updated reference transcriptome of complete, partial, and internal genes. Additionally, EnTAP will output each of these in separate files as well as the genes where no open reading frame was found, or the rejected sequences. All files will contain both the nucleotide and amino acid sequences where available. Lastly, statistics on the number of partial, complete, and internal genes will be calculated as well as statistical information on the new reference transcriptome and the rejected sequences including n50, n90, average sequence length (nucleotide base pairs), longest sequence, and shortest sequence. This provides useful information when comparing rejected sequences to kept sequences.

2.2.2 Expression Filtering

The goal of expression filtering, or transcript quantification, is to determine the relative abundance levels of transcripts when taking into account the sequenced reads and how they map back to the assembled transcriptome and using this information to filter out suspect expression profiles possibly originated from poor or incomplete assemblies [58, 59].

Software attempts to remedy the aforementioned problems by creating a statistical model relating to the aligned reads and providing an “accurate” means of quantification. Among these is RSEM, developed by Bo Li and Colin Dewey of the University of Wisconsin-Madison [58]. Algorithmically, RSEM incorporates Expectation-Maximization as a statistical model and progressively iterates to approximate maximum likelihood estimates. RSEM is particularly useful in non-model eukaryotic samples as it does not require mapping back to a reference genome and

provides a means of escaping further complication incurred by reads spanning exon-intron junctions. RSEM was chosen as the main software package for expression filtering, or transcript quantification in EnTAP [58].

Provided the execution phase is provided with both the reference FASTA and an ungapped alignment file (SAM/BAM format), RSEM will be executed with the following commands:

- rsem-sam-validator ALIGNMENT_PATH
    - This command is performed to assess the validity of the input file, whether it be BAM or SAM formatted.
- convert-sam-for-rsem -p THREADS ALIGNMENT_PATH OUTPUT_PATH
    - Following successful validation of the alignment file, it will be converted into BAM format in order for RSEM to read it.
- rsem-prepare-reference TRANSCRIPTOME_PATH REFERENCE_OUT_PATH
    - Preparation of a reference is necessary in order to calculate expression levels of the transcriptome.
- rsem-calculate-expression --bam -p THREADS ALIGNMENT_PATH REFERENCE_OUT_PATH EXPRESSION_OUT_PATH
    - Following this command, expression levels will be calculated for the transcriptome.
    - Additionally, the user has the option to flag “--paired-end” if the reads are paired-end reads.

The default FPKM threshold is set to 0.5, of which the user has control over to change. Again, SEQUENCE_MAP is updated with FPKM and rejected sequence information. Following execution, EnTAP returns a FASTA file of retained sequences and a FASTA file of those sequences removed. Summary statistics on the data before and after RSEM is provided to the master log file. Control is then shifted to the FILTER state which will in turn filter and pick the new reference transcriptome. The retained sequences proceed to the similarity search stage.

2.3 Similarity Search

An essential stage in an annotation pipeline is gene identification, or similarity searching against reference protein sequences from one or more curated databases such as RefSeq. The most popular methods of similarity search include NCBI's BLAST, and recent modifications which enhance speed - DIAMOND, and RAPSearch2 [28-30]. DIAMOND incorporates many of these methods and has been tested to outperform BLASTX with speeds over 20,000 times faster with similar resulting sensitivity [29]. As a result, DIAMOND is incorporated into the EnTAP pipeline as a means of rapid and accurate similarity searching.

2.3.1 Database Selection

The EnTAP pipeline allows the user to configure up to three different protein databases for similarity search execution in EnTAP. If new databases or new version of existing databases are needed, the configuration stage must be run with the new source FASTA file. The pipeline will identify the header format that is common to NCBI and to Ensembl automatically in order to capture species and the protein term description. Selection of the database is defined by the

user and the recommendation is to select the most curated set of databases that likely represent the closest species. More curated databases have less redundancy and more full-length annotations; however, they may limit the scope too much for certain species. Should the user be interested in pathways, one of the database selections should include a well resolved model species where extensive pathway information will be available.

2.3.2 Selecting an Optimal Alignment

Following database selection and alignment, the most optimal alignment is selected for each database and from these, a best hit is selected for each query sequence. Due to the nature of similarity searching, several alignments can be found for a given query sequence. The selection of the best-hit is unique to the EnTAP software and considers multiple factors. These parameters include: alignment scores (E-value), query coverage, contaminant status, taxonomic relationship to query species, and informativeness of target description.

Within a database, selection begins with an E-value comparison between the query sequence and the target sequence. If the range of these two values is below a threshold, contaminant status is evaluated. However, if the hits are out of this range, the lowest E-value hit is chosen. The user is permitted to flag taxonomic contaminants during execution as hits they would prefer were identified and removed from the transcriptome reference. This provides a phylogenetic filter, such that if a user selects “fungi” as a contaminant, all hit species (derived from the reference database) with a lineage to fungi will not be favored as optimal hits. As previously described, the taxonomic incorporation into optimal alignment selection is based upon a serialized version of the NCBI taxonomic database. If the species is not found within the

taxonomic database, EnTAP attempts to use the genus to determine taxonomic lineage. Comparison of query coverage is similar to the treatment of E-value in which a specified range is used to determine if an alignment should be selected or if additional comparisons are needed. If the sequences are within a predefined coverage range, a score is calculated for each hit based upon taxonomic relation to the queried species (if provided by user) and informativeness of the hit. An alignment is tagged as uninformative if a match is found between a pre-defined list of common descriptors, such as *predicted protein* and *hypothetical protein*. Selection for informativeness can provide more functional information for each protein within a similar range of alignment scores.

A summary of the process of hit selection can be seen below in Figure 2. Hits being compared enter the process with metrics involved in the selection process including E-value (*eval*), contaminant status (*is_contam*), coverage (*cov*), and "taxonomic score" (*tax_score*). Additionally, there are several constants associated with the selection process including E_VAL_DIF, COV_DIF, and INFORMATIVE_SHIFT. E_VAL_DIF represents the required difference in E-values between hits in order to continue to the next stage of selection; assigned a value of seven. This prevents very good hits within a database from being filtered out by possible contaminant status or taxonomic relationship. In a similar vein, COV_DIF represents the coverage difference between hits that must be attained in order to continue in the selection process; assigned a value of five. Finally, INFORMATIVE_SHIFT is used within the function, *calculate_tax_score()*. This function incorporates (if inputted by the user) the taxonomic relationship of hits compared to that of species being annotated through accessing the taxonomic database. Each similarity in lineage increases a hit's taxonomic score, while being an

informative hit increases this further by a value of five. This accomplished selection of the hit based upon similar taxonomic lineage to the species being annotated as well as informativeness of the hit.

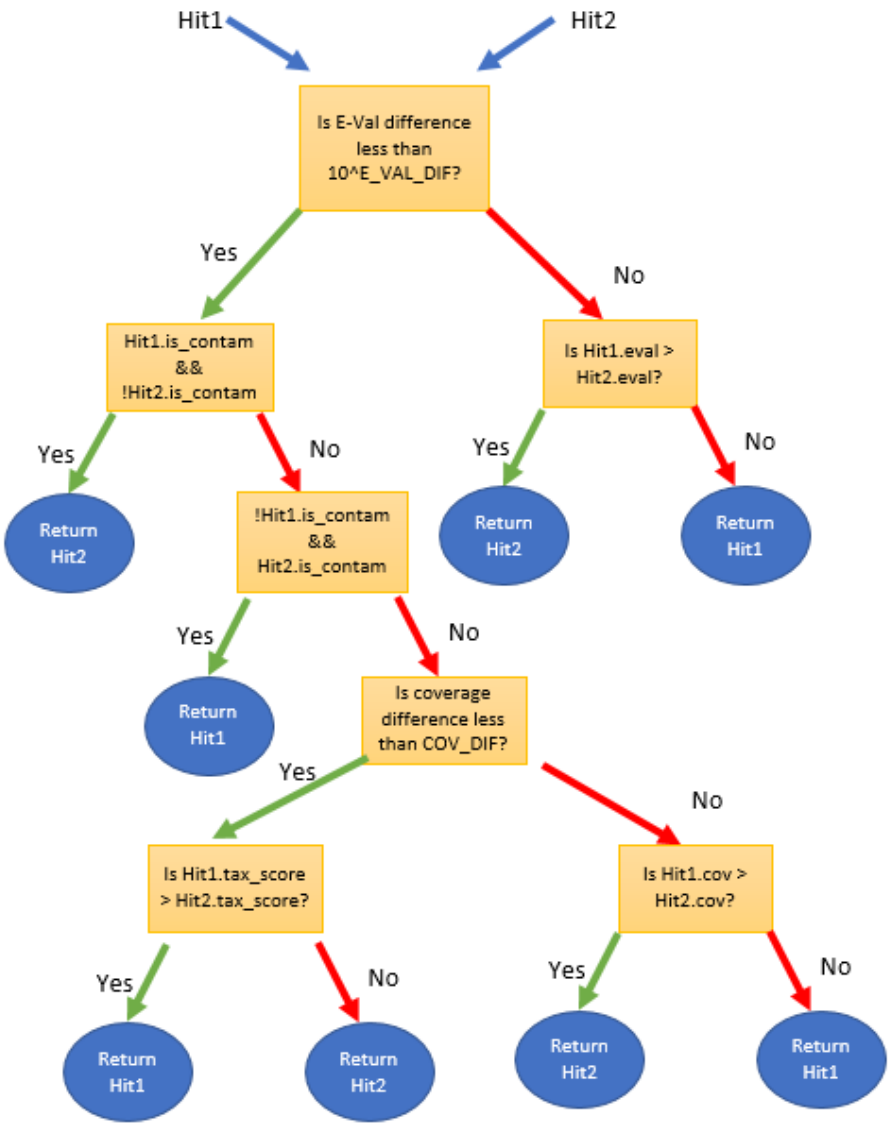

Figure 2: Best-Hit Selection

The following commands are used for similarity searching:

- diamond blastp USER_DATABASE --query-cover COVERAGE --more-sensitive --top 3 -q INPUT_TRANSCRIPTOME -o OUTPUT_FILE -p THREADS -f 6 qseqid sseqid pident length mismatch gapopen qstart qend sstart send evalue bitscore qcovhsp stitle
    - By default, COVERAGE is set to 50, limiting results that only appear over 50 coverage. The user has control over this parameter.
    - The user has control over the THREADS variable.
    - The "-f" flag provides a specified format from the DIAMOND execution in which pertinent information can be extracted from to perform hit selection.
    - The "--top 3" flag is used for selecting the top 3% of hits to be reported.
    - The "--more-sensitive" flag is used to increase the sensitivity of the search while increasing runtime.

Upon successful execution of similarity search, hits are analyzed for each database using the selection method detailed above. Initially, tab separated output files from DIAMOND are parsed using the fast-cpp-csv-parser library. Hits are contained within a *QuerySequence* object and variables, such as: taxonomic lineage, species, e-value, coverage, and contaminant status are set before addition to an ordered hash map with the query sequence ID as keys. As the file is parsed, the database hash map is continuously updated with new *QuerySequence* objects. The best-hit selection method is contained with the *QuerySequence* object through an overridden comparison method allowing *friendly* access to the private member variables stored within either object that are pertinent to best-hit selection. Once the same sequence ID is found within the map, a

comparison is made between both sequences and the map entry points to the better hit of the two. This process continues until the file, and all subsequent database files, are successfully parsed. Database ordered maps are generated separately and contained within a vector to allow for a statistical analysis of hits from each database, including: species, contaminant, and informativeness. Upon completion of this stage, *QuerySequence* objects are flagged in order for a separate comparison to be made not based on e-value (as this value does not carry over between databases) and compared to find the overall hit for each sequence. Again, statistics are calculated for the overall hits found after compiling the results from each database.

The EnTAP output from this stage is rather extensive by providing information at each part in the process. Nucleotide and protein FASTA files are provided for the user for all best-hits, sequences that were tagged as contaminants (within the best-hits), best-hits excluding contaminants, and sequences that did not hit against the database. This is particularly useful as the user may want to run a future analysis excluding the contaminated sequences. These files are provided for each database and for the combined best-hits. Additionally, multiple TSV files are generated representing the same metrics as previously stated. These TSV files contain information on the DIAMOND run as well as species and database origin information. Following completion of this stage, EnTAP transitions to GENE_ONTOLOGY.

2.4 Orthologous Gene Families

The EggNOG database provides a source of orthologous groups generated from the clustering of 2031 prokaryotic and eukaryotic genomes. These orthologous groups are categorized into a controlled vocabulary of 107 taxonomic levels. The orthologous genes are

mapped to the Gene Ontology database, several protein domain databases, and biological pathways (KEGG) [37].

The EggNOG-mapper is integrated into the EnTAP pipeline and can assign an average of 32 more terms per protein with speeds up to 15 times faster than NCBI BLAST annotation and 2.5 times faster than utilizing InterProScan [60]. This software depends on SQLITE functionality in Python as well as databases downloaded from the EggNOG servers.

EggNOG-mapper is ran by the following command:

- emapper.py -i TRANSCRIPTOME_INPUT --output OUTPUT_PATH --cpu THREADS -m diamond
    - The "-m," flag specified with "diamond" instructs EggNOG-mapper to compare against the Eggnog databases through DIAMOND similarity search under high sensitivity. This provides a rapid and accurate annotation of the dataset.

As like before, the SEQUENCE_MAP is updated with orthologous family information, including the particular group that was aligned to as well as the gene. Statistics are calculated to determine the number of successful orthologous mappings found within the transcriptome.

2.4.1 Gene Ontology Terms

Applicable Gene Ontology terms from all three categories are assigned based upon orthogroup assignments in the previous step. The EnTAP pipeline integrates term descriptions and hierarchy information (based upon the directed acyclic graph setup of Gene Ontology) from the previously configured term mapping database. A visualization of the Gene Ontology hierarchy

can be seen below in Figure 3 with levels increasing by one as you traverse downwards. After successful integration of orthologous groups, the previously serialized Gene Ontology mapping database is read back into memory through the Boost C++ libraries as an ordered map of structures containing level, term, and categorical information (biological process, molecular function, or cellular component). The mapping database is needed due to the lack of information on Gene Ontology IDs provided by the EggNOG-mapper, with IDs being the only reported information. The user can generate term assignments at all levels or at specified levels which can be used as input to enrichment packages through a "--level" flag. The Gene Ontology term information is separated into the hierarchical categories for the user in the final output and labelled with a description as well as the level.

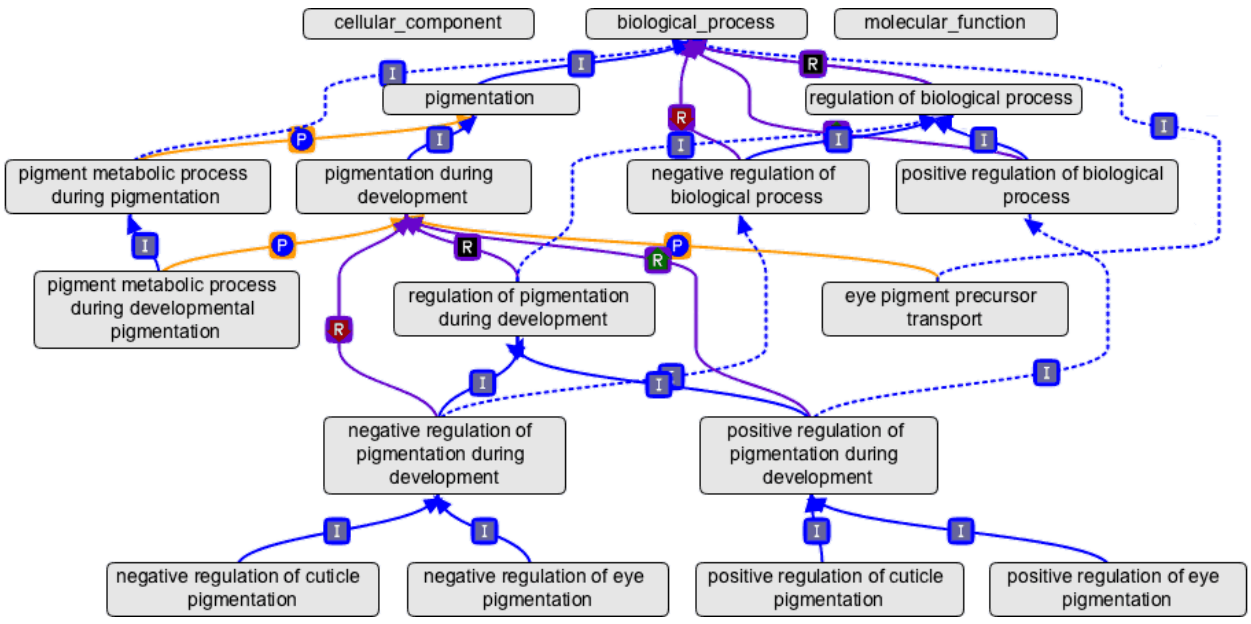

Figure 3: Gene Ontology Hierarchy[43]

2.4.2 Biological Pathways

The EggNOG-mapper provides biological pathway information from the KEGG database. Each pathway is associated with a unique identifier and orthologous groups can be associated with one or more pathway. As with each stage of the pipeline, the SEQUENCE_MAP is updated with any relevant pathway information and a statistical analysis is performed denoting the number of sequences that were assigned pathway terms and those that were not in addition to the total terms assigned.

2.5 Final Output

Upon completion of EnTAP, the user will receive several final annotation files in TSV format summarizing information gleaned from each stage of the pipeline such as hits through similarity searching and orthologous assignments, with accompanying GO and pathways term assignments (seen in Figure 1). Unannotated and annotated sequences are provided in FASTA format of nucleotide and amino acid, where available. Additionally, overall statistics are calculated for the execution and printed to a final log file.

2.6 Methodology for Evaluating Performance

2.6.1 Description of Transcriptome Set

The transcriptome dataset used for evaluation and benchmarking was sourced from *Pinus flexilis* (limber pine) needle tissue. This species represents a true non-model with no reference genome and no significant genomic resources for its very diverse genus. Short reads were generated from a single library of four pooled individuals on a single lane of an Illumina HiSeq

2500 (100bp PE). The resulting raw reads were subject to quality control (trimming and read removal) with Sickle with a minimum length threshold of 40bp and a minimum Phred-scaled quality score of 30 [61]. A total of 124694125 reads passed quality control These reads were assembled through Trinity (version 2.4.0) with a minimum contig length of 350bp resulting in 30,291 final transcripts [23] . A summary of initial dataset preparation can be seen below in Table 1.

Table 3: Pinus flexilis RNA-Seq Summary

Process	Parameters	Results
Quality Control (Sickle)	Minimum Length: 40bp Minimum Quality: 30	124694125 paired-end reads
de novo Assembly (Trinity)	Minimum Contig Length: 350bp	30,291 assembled sequences

2.6.2 Evaluations Conducted

Benchmarking comparisons were conducted among EnTAP (pre-release version 0.5.6.1), Blast2GO (standalone pro version 4.1.9), and Trinotate (v3.0.2). These three packages were selected for the closest overlap in terms of features and flexibility (Table 2). Several metrics were considered including installation procedures, flexibility, speed, annotation rate, accuracy of annotation (contaminant detection, informativeness, and phylogenetic relevance), and accessibility. Installation procedures can be broken down into several categories such as database downloading and configuration, any dependencies that may be required, and how complicated the installation may be. The analysis of software flexibility will involve flexibility within similarity searching such as hit selection, protein database selection, and parameter

flexibility. The speed of software execution can be slightly difficult as the aforementioned software packages each perform moderately different functions utilizing different software, so comparisons are made within parts of the overall pipeline. Annotation rate is based upon analysis of the final transcriptome that will be used for annotation after frame selection or expression filtering. Due to the final transcriptome possibly getting smaller due to rejected sequences where a frame wasn't found, annotation rate comparisons are made sequence-to-sequence to give fairness between software.

The EnTAP run is based upon running frame selection, protein similarity search (against both Swiss-Prot and RefSeq Complete), and EggNOG-mapper in the University of Connecticut's BBC cluster utilizing 8 dual-core 2.0 GHz Intel Xeon processor with 64 GB of RAM. Additionally, this benchmark included a taxonomic contaminate filter of fungi, bacteria, and insecta, an e-value cutoff of 10^{-5}, a similarity search hit limitation of 5 hits, and a minimum coverage of 50%.

The Blast2GO Pro benchmark ran nucleotide "blastx-fast" on Blast2GO's high-performance CloudBlast service (only available to paid Pro users) with similar settings to the EnTAP run against the Uniprot and RefSeq Complete databases. This is the fastest BLAST service Blast2GO provides. Blast2GO annotation was performed with default metrics of an annotation cutoff of 55, GO weight of 5, no taxonomic filter, E-value-hit-filter of 1E-6, no HSP-hit coverage cutoff, and a hit filter of 500.

The Trinotate comparison ran Transdecoder for frame selection and ran against the Swiss-Prot database with similar parameters to the previous runs.

3 Results and Discussion

3.1 Installation Comparison

The EnTAP installation procedure begins with the user downloading the latest release source code from the GitLab page (seen in Appendices 5.1). Software contained with the EnTAP pipeline is included within the original EnTAP download (with the exception of GeneMarkS-T which requires separate licensing that is free for academic use). The user is then prompted to compile DIAMOND and RSEM with the other packages being primarily script based (as opposed to compiled packages) and not requiring compilation. Alternatively, if the user already has these packages installed on their system, they need to change the execution paths on a configuration text file provided by EnTAP and skip this stage. Due to the fact that a user may not want to include every part of the pipeline, they are not required to install every package, only the ones they would like to incorporate into their annotation. After successful installation of pipeline software, EnTAP must be installed. As EnTAP is designed to run primarily within a Unix environment, it must be compiled from source code. This procedure is very simple as it incorporates CMake to generate a MakeFile while checking required dependencies [56]. The user will run “cmake” and “make,” or “make install,” to complete the EnTAP installation.

Trinotate has a similar procedure to that of EnTAP in requiring the installation and compilation of several supporting software such as Transdecoder, Trinity, and SQLITE. Trinotate, being primarily command-line based as with EnTAP requires source code compilation.

Blast2GO has a very different installation process compared to EnTAP and Trinotate that does not require user compilation of individual packages from source code. To install Blast2GO, the user must download the installation executable from the Blast2GO website and install it onto their local system.

Due to the contrasting nature of the packages, with EnTAP and Trinotate primarily being used within a Unix environment and Blast2GO being standalone software with an emphasis on a Graphical User Interface (GUI), the installation procedure is more intensive for EnTAP and Trinotate. However, the process is streamlined as much as possible for the user with flexibility in what packages are required to install.

3.2 Flexibility

When examining flexibility in terms of protein reference database selection, there is significant flexibility within the EnTAP pipeline allowing any FASTA formatted database to be indexed by EnTAP (utilizing DIAMOND) and permitted to be ran during similarity search. Blast2GO has a limited number of databases accessible through its cloud BLAST feature, although it does allow custom database creation in the Pro version.

EnTAP selects the best hit from each database during similarity search. That is, if the user wishes to compare their transcriptome against several databases, EnTAP will select an overall best hit from the combined searches. Additionally, this best hit is weighted based on: E-value, query coverage, contaminant status, taxonomic relationship, and informativeness. Contaminant identification is a very flexible feature allowing for any taxonomic lineage (within the NCBI taxonomic database) to be deemed a contaminant and unfavored. Blast2GO provides very basic

selection within a single database based on alignment scores alone. This primarily relies on thresholds to select hits concerning e-value and coverage. Additionally, the user is permitted to enter a basic taxonomic filter (hits will be confined to this taxon) and the ability to remove any sequences that contain a certain keyword or phrase within them. It should be noted however, that this only works based upon the description pulled from the databases and will not filter out a specific taxonomic lineage.

Parameter flexibility of core features is similar between packages such as coverage cutoffs and minimum e-value selection during similarity search. Although, Blast2GO provides some additional parameter specification for annotation of results, such as e-value and coverage cutoffs (similar to that of similarity search) and additional output formats.

3.3 Speed

The benchmark comparison seen below in Figure 4 and Table 4 represents runtimes of each stage of the EnTAP pipeline compared to each stage of the Blast2GO and Trinotate pipelines. It should be noted that, due to the additional frame selection step in EnTAP and Trinotate, the transcript number is varied between both runs with EnTAP executing the rest of the pipeline with 23696 sequences and Blast2GO with 30291 sequences. All times are reported as wallclock times. Similarity searching was not run against the RefSeq Complete database for Trinotate since it is designed to use Swiss-Prot.

The results depicted below in Figure 4 and Table 4 demonstrate the runtime disparity between EnTAP, Blast2GO Pro, and Trinotate, primarily seen in similarity searching. Overall runtimes for Blast2GO, EnTAP, and Trinotate were 36.25 hours, 9.56 hours, and 13 hours,

respectively. A faster runtime was observed through Blast2GO and Trinotate in mapping the transcripts to GO terms and pathway information. This can be attributed to the fact that EnTAP utilizes EggNOG-mapper to provide an additional similarity search against orthologous genes, while Blast2GO and Trinotate maps the similarity search results directly to GO terms through the use of NCBI mapping files. Because of this, if the transcriptome has very few similarity search results from well curated model species, the final annotation will have little contributions from GO or pathway databases.

The disparity among runtimes can largely be attributed to the different methods of similarity searching. Blast2GO's Pro service incorporates a high-performance cloud computing BLAST service to align against reference protein databases, while EnTAP incorporated DIAMOND as a means of similarity search and Trinotate utilizes BLAST. In addition, BLAST2GO's compute time relies on their external servers that cannot be configured or optimized by the end user.

Table 4: Wallclock Benchmark Between Pipelines

Stage	Blast2GO Pro	EnTAP	Trinotate
Similarity Search (RefSeq Complete)	31.5	2.32	N/A
Similarity Search (Swiss-Prot)	1.5 h	0.07 h	12.1 h
GO Term Annotation	3.25 h	7.17 h	1.13 h
Overall	36.25 h	9.56 h	13 h

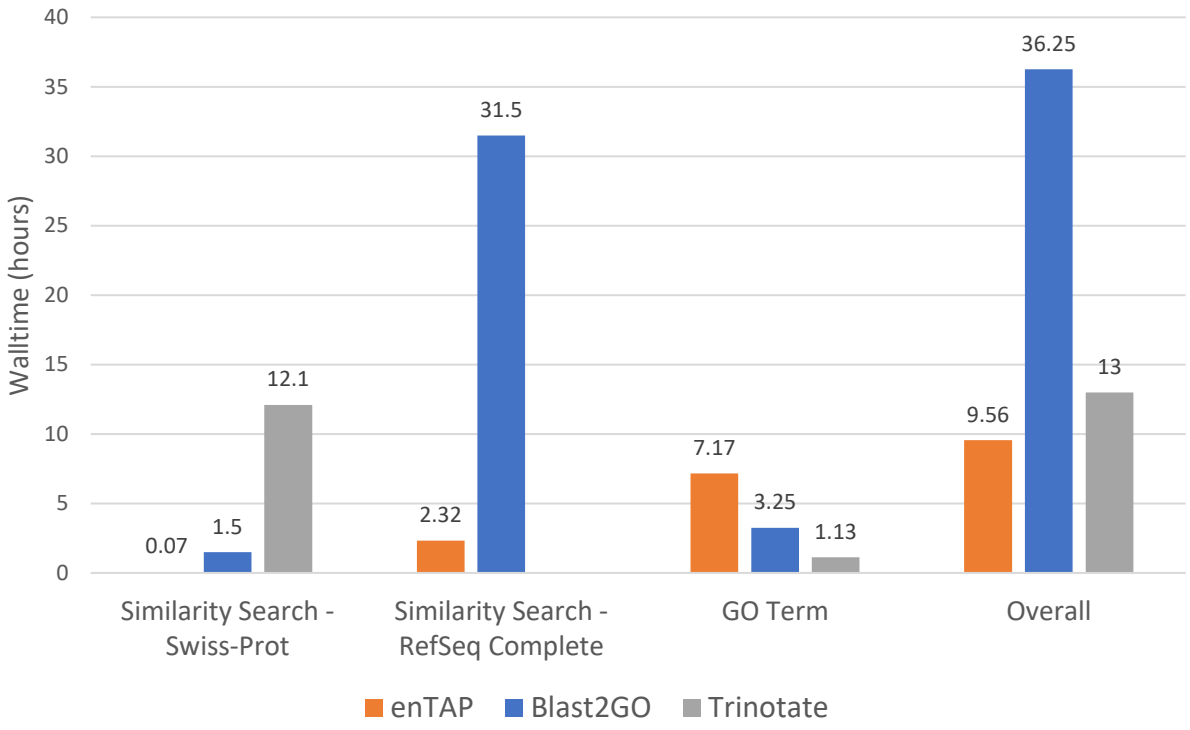

Figure 4: Wallclock Benchmark between Pipelines

3.4 Annotation Rate

The rate of annotation between pipelines can be seen in Figure 5 and Table 5 below. As previously mentioned, these comparisons are made primarily through Swiss-Prot since all three pipelines can be compared with this database. EnTAP and Trinotate generate more hits when compared with Blast2GO. A total of 16218 (approximately 68.4% of the total set ran through similarity searching) sequences hit using the EnTAP pipeline and 19062 (62.93%) with the Trinotate pipeline, compared to 9060 (approximately 29.9% of the total set ran through similarity searching) sequences with Blast2GO. The similarity search results affect downstream annotation

substantially with Blast2GO as it relies on mapping of the BLAST hits to assign Gene Ontology and pathway terms. When comparing against the sequences originally ran through similarity search, this resulted in a 27.1% annotation rate. However, it should be noted that annotation of the similarity search results (sequences that did hit against the database) was rather high with an annotation rate of approximately 90%. Comparatively, EnTAP had an annotation rate of approximately 51.5% when comparing against the entire transcriptome and an annotation rate of 75.3%. The rate of annotation of the overall transcriptome is significantly lower with Blast2GO, however the annotation rate of successful BLAST hits is much higher with Blast2GO due to the mapping methodologies of successful hits having an associated annotation.

Table 5: Annotation Rate Between Pipelines

Stage	Blast2GO (Pro)	EnTAP	Trinotate
Sequences	30291	23696 (6595 lost due to frame selection)	30921
Similarity Search - Swiss Prot Hits	9060 (29.9%)	16218 (68.4%)	19062 (62.9%)
GO Term Annotation Rate	8195 (27.1%)	12211 (51.5%)	7186 (37.7%)

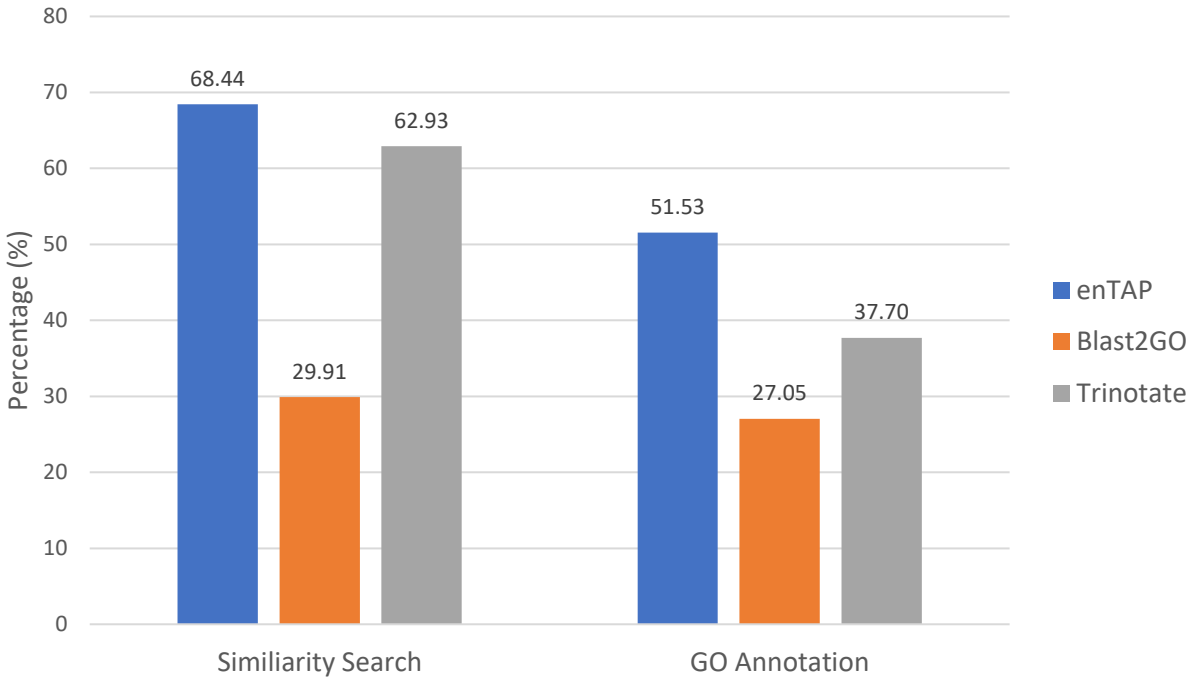

Figure 5: Annotation Rates as a Percentage of Reference Transcriptome

The disparity in annotation rate can be better understood by examining the quality of the hits and why so many sequences were without a hit using similarity search of the Blast2GO pipeline. The input transcriptome was frame selected through GeneMarkS-T in the EnTAP and Transdecoder in the Trinotate benchmark and a protein alignment was done between the

reference databases and the frame selected sequences. Alternatively, using a nucleotide similarity searching method, the software does not incorporate open reading frame detection as an independent step which may impact the accuracy. Although many of the sequences processed with EnTAP were lost (6595, approximately 22%), the annotation rate among those remaining was high. This can be directly attributed to a higher quality hit (higher coverage of the sequence) associated with frame selection through GeneMarkS-T. To further evaluate, the coverage threshold of 50 was removed for another run of Blast2GO against the Uniprot Swiss-Prot database to examine how large of a difference the resulting annotation rate would be. The results of this can be seen below in Table 6 again with a side-by-side comparison to EnTAP (EnTAP maintains the 50% coverage threshold mentioned previously). The rate of hits through similarity search nearly doubled with the exclusion of the coverage threshold for Blast2GO. This can be attributed to a large number of sequences with low quality alignments to Swiss-Prot (less than 50% coverage).

Table 6: Annotation Rate without 50% Coverage on Blast2GO

Stage	Blast2GO (Pro) (no threshold)	EnTAP (50 coverage)
Sequences	30291	23696 (6595 lost due to frame selection)
Similarity Search - Swiss Prot Hits	18877 (62.3%)	16218 (68.4%)
Similarity Search - Swiss Prot No Hits	11414(37.7%)	7478 (31.6%)
GO Term Annotation Rate	17476 (57.6%)	12211 (51.5%)

Although it is rather difficult to determine whether a non-model annotation is "correct," a further comparison in the quality of the results from similarity searching can be made based upon contamination status of hits. Considering the species dataset, limber pine, it is reasonable to evaluate potential contaminants from fungal and bacteria lineages [62, 63]. The relative contaminant percentage for EnTAP, Blast2GO, and Trinotate were 3.44%, 7.56%, and 8.92, respectively. Since neither Blast2GO, nor Trinotate incorporate a phylogenetic contaminant filter, a higher contaminant rate can be seen in these executions leading to lower quality hits. The top three contaminants of the Blast2GO run, accounting for nearly 206 hits, include *Schizosaccharomyces pombe*, *Saccharomyces cerevisiae*, and *Synechocystis sp. PCC 6803*, each having lineage tracing back to either bacteria or fungi. Additionally, the top three contaminants produced by the EnTAP similarity search (and still kept) account for 176 contaminant hits include *Schizosaccharomyces pombe (strain 972 / ATCC 24843)*, *Saccharomyces cerevisiae (strain ATCC 204508 / S288c)*, and *Bacillus subtilis (strain 168): 35(6.27%)*. A summary of the contaminant information can be seen below in Figure 6.

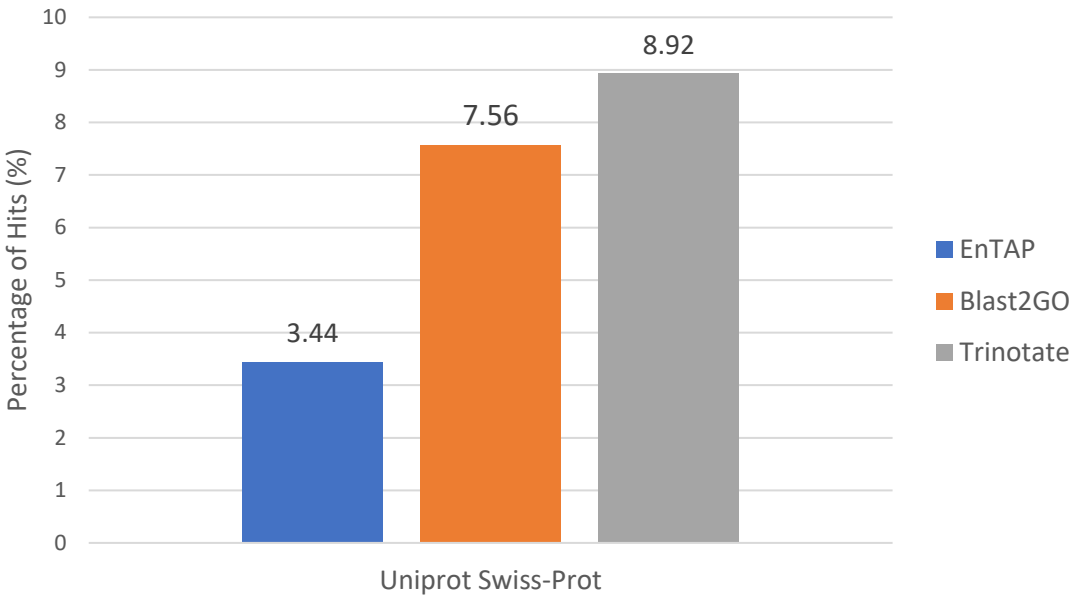

Figure 6: Contaminant Comparison Between Pipelines

A final comparison was made between the informativeness of the hits when comparing sequences successfully aligned with the reference database of EnTAP and Blast2GO runs. Due to the relatively small nature of the Swiss-Prot database, informative hits were rather similar with EnTAP producing approximately 3.54% uninformative hits and Blast2GO resulting in 2.94%. As can be seen, Blast2GO has a higher informative hit-rate compared to that of EnTAP, however the nature of the Uniprot database can be a very likely contributor with EnTAP favoring taxonomic relevance over a more descriptive result.

4 Conclusion

Non-model transcriptome annotation has many challenges and caveats that must be considered in pipeline construction and execution. From an assembly possibly riddled with sequencing fragments and errors, to the incomplete databases used for comparison with non-model species, a thorough analysis of strengths, weaknesses, and possible solutions to many of these challenges must be considered. EnTAP is designed to overcome many of these outlined challenges by providing a means to transcriptome filtering and frame selection prior to annotation. Similarity searching against reference protein databases is performed very rapidly while maintaining a great deal of sensitivity, while Gene Ontology and pathway information is assigned to the reference transcriptome by orthologous group matching. At each step, EnTAP provides useful statistical information and a plethora of outputs to show the user how their data is being manipulated. EnTAP's annotation and hit rates rival that of Blast2GO's (pro version) and Trinotate's by providing a similarity searching hit rate of 68.4% of the transcriptome compared to 29.9% with that of Blast2GO and 62.9% with Trinotate, having far fewer false positive contaminants included. Informative hits were higher with Blast2GO with a rate of 2.94% compared to 3.54% which can likely be contributed to the nature of the database. Furthermore, EnTAP saw annotation rates of 51.5% compared to Blast2GO's 27.1% and Trinotate's 37.7%. By providing the user with unique features such as additional means of evaluating the assembly, upstream of annotation, and similarity search hit selection based on phylogenetics and informativeness, EnTAP provides a fast, accurate, user-friendly, and reliable alternative to the current software solutions for transcriptome annotation.

5 Appendices

5.1 Source Code

The latest EnTAP source code can be found at:
https://gitlab.com/EnTAP/EnTAP

5.2 Documentation

The latest EnTAP documentation can be found at:
http://EnTAP.readthedocs.io/en/latest/

6 References

[1] A. Conesa, P. Madrigal, S. Tarazona, and D. Gomez-Cabrero, "A survey of best practices for RNA-seq data analysis," *Genome Biology,* 2016.

[2] P. Higgs, *RNA Secondary Structure: physical and computational aspects*. United Kingdom: Cambridge University Press, 2000.

[3] G. Cooper, *The Cell: A Molecular Approach*, 2nd ed. 2000.

[4] R. Lewis, *Human Genetics*, 11 ed. (WCB Cell & Molecular Biology). McGraw-Hill Education, 2014.

[5] J. Adams, "Transcriptome: Connecting the Genome to Gene Function," *Nature Education,* vol. 1, no. 1, p. 195, 2008.

[6] I. Nookaew *et al.*, "A comprehensive comparison of RNA-Seq-based transcriptome analysis from reads to differential gene expression and cross-comparison with microarrays: a case study in Saccharomyces cerevisiae," (in eng), *Nucleic Acids Res,* vol. 40, no. 20, pp. 10084-97, Nov 2012.

[7] Q. Q. Wang, F. Liu, X. S. Chen, X. J. Ma, H. Q. Zeng, and Z. M. Yang, "Transcriptome profiling of early developing cotton fiber by deep-sequencing reveals significantly differential expression of genes in a fuzzless/lintless mutant," (in eng), *Genomics,* vol. 96, no. 6, pp. 369-76, Dec 2010.

[8] C. Schunter, S. V. Vollmer, E. Macpherson, and M. Pascual, "Transcriptome analyses and differential gene expression in a non-model fish species with alternative mating tactics," (in eng), *BMC Genomics,* vol. 15, p. 167, Feb 2014.

[9] S. Gilbert, *Developmental Biology*, 6th ed. 2000.

[10] J. M. Heather and B. Chain, "The sequence of sequencers: The history of sequencing DNA," (in eng), *Genomics,* vol. 107, no. 1, pp. 1-8, Jan 2016.

[11] R. Lewis, *Human Genetics*, 11 ed. (WCB Cell & Molecular Biology). McGraw-Hill Education, 2014.

[12] S. Goodwin, J. D. McPherson, and W. R. McCombie, "Coming of age: ten years of next-generation sequencing technologies," vol. 17, ed: Nature Reviews Genetics, 2016, pp. 333-351.

[13] E. Gongora-Castillo and C. R. Buell, "Bioinformatics challenges in de novo transcriptome assembly using short read sequences in the absence of a reference genome sequence," *Royal Society of Chemistry,* no. 30, p. 490, 2012.

[14] (2017). *System specifications for the NextSeq Series*.

[15] *Ion Proton System Specifications*.

[16] JGI GOLD: Genomes Online Database [Online]. Available: https://gold.jgi.doe.gov/statistics

[17] C. Mora, D. P. Tissensor, S. Adl, A. G. B. Simpson, and B. Worm, "How Many Species Are There on Earth and in the Ocean?," *PLoS Biology,* vol. 9, no. 8, 2011.

[18] B. J. Haas, A. Papanicolaou, M. Yassour, M. Grabherr, P. D. Blood, and J. Bowden, "Do novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis," *Nature Protocols,* vol. 8, pp. 1494-1512, 2013.
[19] I. G. Romero, A. A. Pai, J. Tung, and Y. Giland, "RNA-seq: impact of RNA degradation on transcript quantification," *BCM Biology,* vol. 12, no. 42, 2014.
[20] D. R. Smith, "RNA-Seq data: a goldmine for organelle research," *Briefings In Functional Genomics,* vol. 12, no. 5, pp. 454-456, 2013.
[21] M. Konczal, P. Koteja, M. T. Stuglik, J. Radwan, and W. Babik, "Accuracy of allele frequency estimation using pooled RNA-Seq," (in eng), *Mol Ecol Resour,* vol. 14, no. 2, pp. 381-92, Mar 2014.
[22] Z. Li *et al.*, "Comparison of the two major classes of assembly algorithms: overlap-layout-consensus and de-bruijn-graph," (in eng), *Brief Funct Genomics,* vol. 11, no. 1, pp. 25-37, Jan 2012.
[23] M. G. Grabherr *et al.*, "Full-length transcriptome assembly from RNA-Seq data without a reference genome," (in eng), *Nat Biotechnol,* vol. 29, no. 7, pp. 644-52, May 2011.
[24] X. Ren *et al.*, "Evaluating de Bruijn graph assemblers on 454 transcriptomic data," (in eng), *PLoS One,* vol. 7, no. 12, p. e51188, 2012.
[25] S. Tang, A. Lomsadze, and M. Borodovsky, "Identification of protein coding regions in RNA transcripts," *Nucleic Acids Research,* 2015.
[26] S. Das and D. L. Mykles, "A Comparison of Resources for the Annotation of a De Novo Assembled Transcriptome in the Molting Gland (Y-Organ) of the Blackback Land Crab," *Integrative and Comparative Biology,* pp. 1-10, 2016.
[27] N. A. O'Leary *et al.*, "Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation," (in eng), *Nucleic Acids Res,* vol. 44, no. D1, pp. D733-45, Jan 2016.
[28] "Basic Local Alignment Search Tool (BLAST)," ed: NCBI: National Center of Biotechnology Information.
[29] B. Buchfink, C. Xie, and D. H. Huson, "Fast and sensitive protein alignment using DIAMOND," (in eng), *Nat Methods,* vol. 12, no. 1, pp. 59-60, Jan 2015.
[30] Y. Zhao, H. Tang, and Y. Ye, "RAPSearch2: a fast and memory-efficient protein similarity search tool for next-generation sequencing data," (in ENG), *Bioinformatics,* vol. 28, no. 1, pp. 125-6, Jan 2012.
[31] H. Li and N. Homer, "A survey of sequence alignment algorithms for next-generation sequencing," (in eng), *Brief Bioinform,* vol. 11, no. 5, pp. 473-83, Sep 2010.
[32] A. Mitchell, H.-Y. Chang, and L. Daugherty, "The InterPro protein families database: the classification resource after 15 years," *Nucleid Acids Research,* vol. 43, no. D1, 2015.
[33] I. Letunic, T. Doerks, and P. Bork, "SMART: recent updates, new developments and status in 2015," (in eng), *Nucleic Acids Res,* vol. 43, no. Database issue, pp. D257-60, Jan 2015.
[34] R. D. Finn, P. Coggill, R. Y. Eberhardt, and S. R. Eddy, "The Pfam protein families database: towards a more sustainable future," *Nucleic Acids Research,* vol. 44, 2016.
[35] H. Mi, S. Poudel, and A. Muruganujan, "PANTHER version 10: expanded protein families and functions, and analysis tools," *Nucleid Acids Research,* vol. 44, no. D1, 2016.

[36] P. Jones *et al.*, "InterProScan 5: genome-scale protein function classification," (in eng), *Bioinformatics,* vol. 30, no. 9, pp. 1236-40, May 2014.
[37] J. Huerta-Cepas *et al.*, "eggNOG 4.5: a hierarchical orthology framework with improved functional annotations for eukaryotic, prokaryotic and viral sequences," (in eng), *Nucleic Acids Res,* vol. 44, no. D1, pp. D286-93, Jan 2016.
[38] M. Y. Galperin, K. S. Makarova, Y. I. Wolf, and E. V. Koonin, "Expanded microbial genome coverage and improved protein family annotation in the COG database," (in eng), *Nucleic Acids Res,* vol. 43, no. Database issue, pp. D261-9, Jan 2015.
[39] E. M. Zdobnov *et al.*, "OrthoDB v9.1: cataloging evolutionary and functional annotations for animal, fungal, plant, archaeal, bacterial and viral orthologs," (in eng), *Nucleic Acids Res,* vol. 45, no. D1, pp. D744-D749, Jan 2017.
[40] X. Chen and J. Zhang, "The ortholog conjecture is untestable by the current gene ontology but is supported by RNA sequencing data," (in eng), *PLoS Comput Biol,* vol. 8, no. 11, p. e1002784, 2012.
[41] N. L. Nehrt, W. T. Clark, P. Radivojac, and M. W. Hahn, "Testing the ortholog conjecture with comparative functional genomic data from mammals," (in eng), *PLoS Comput Biol,* vol. 7, no. 6, p. e1002073, Jun 2011.
[42] C. R. Primmer, S. Papakostas, E. H. Leder, M. J. Davis, and M. A. Ragan, "Annotated genes and nonannotated genomes: cross-species use of Gene Ontology in ecology and evolution research," *Molecular Ecology,* vol. 22, no. 12, pp. 3216-3241, 13 June 2013.
[43] G. O. Consortium, "Gene Ontology Consortium: going forward," (in eng), *Nucleic Acids Res,* vol. 43, no. Database issue, pp. D1049-56, Jan 2015.
[44] M. Kanehisa, M. Furumichi, M. Tanabe, Y. Sato, and K. Morishima, "KEGG: new perspectives on genomes, pathways, diseases and drugs," (in eng), *Nucleic Acids Res,* vol. 45, no. D1, pp. D353-D361, Jan 2017.
[45] R. Caspi *et al.*, "The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases," (in eng), *Nucleic Acids Res,* vol. 42, no. Database issue, pp. D459-71, Jan 2014.
[46] A. Fabregat *et al.*, "The Reactome pathway Knowledgebase," (in eng), *Nucleic Acids Res,* vol. 44, no. D1, pp. D481-7, Jan 2016.
[47] "Trinotate: Transcriptome Functional Annotation and Analysis," ed: Trinity Group.
[48] M. Van Bel, S. Proost, C. Van Neste, D. Deforce, Y. Van de Peer, and K. Vandepoele, "TRAPID: an efficient online tool for the functional and comparative analysis of de novo RNA-Seq transcriptomes," (in eng), *Genome Biol,* vol. 14, no. 12, p. R134, Dec 2013.
[49] S. Götz *et al.*, "High-throughput functional annotation and data mining with the Blast2GO suite," *Nucl Acids Res,* 2008.
[50] (2015). *Transdecoder (Find Coding Regions Within Transcripts)*. Available: https://transdecoder.github.io/
[51] B. Strasser. *Fast C++ CSV Parser*. Available: https://github.com/ben-strasser/fast-cpp-csv-parser
[52] *Read the Docs*. Available: https://readthedocs.org/
[53] *GitLab*. Available: https://gitlab.com/
[54] *boost C++ Libraries*. Available: http://www.boost.org/
[55] *Pstreams*. Available: https://github.com/jwakely/pstreams

[56] *CMake*. Available: https://cmake.org/
[57] S. Federhen, "The NCBI Taxonomy Database," *Nucleic Acids Research,* vol. 40, no. D1, 2011.
[58] B. Li and C. N. Dewey, "RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome," (in eng), *BMC Bioinformatics,* vol. 12, p. 323, Aug 2011.
[59] R. Bohnert and G. Rätsch, "rQuant.web: a tool for RNA-Seq-based transcript quantitation," (in eng), *Nucleic Acids Res,* vol. 38, no. Web Server issue, pp. W348-51, Jul 2010.
[60] J. Huerta-Cepas *et al.*, "Fast genome-wide functional annotation through orthology assignment by eggNOG-mapper," (in eng), *Mol Biol Evol,* Apr 2017.
[61] *Sickle - A windowed adaptive trimming tool for FASTQ files using quality*. Available: https://github.com/najoshi/sickle
[62] N. J. Brereton *et al.*, "Comparative Transcriptomic Approaches Exploring Contamination Stress Tolerance in Salix sp. Reveal the Importance for a Metaorganismal de Novo Assembly Approach for Nonmodel Plants," (in eng), *Plant Physiol,* vol. 171, no. 1, pp. 3-24, May 2016.
[63] X. G. Hu *et al.*, "De Novo Transcriptome Assembly and Characterization for the Widespread and Stress-Tolerant Conifer Platycladus orientalis," (in eng), *PLoS One,* vol. 11, no. 2, p. e0148985, 2016.