Efficient evidence-based genome annotation with EviAnn

Aleksey V. Zimin[1,2], Daniela Puiu[1,2], Mihaela Pertea[1,2], James A. Yorke[3] and Steven L. Salzberg[1,2,4,5]

[1]Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD
[2]Center for Computational Biology, Johns Hopkins University, Baltimore, MD
[3]Department of Mathematics, University of Maryland, College Park MD
[4]Department of Computer Science, Johns Hopkins University, Baltimore, MD
[5]Department of Biostatistics, Johns Hopkins University, Baltimore, MD

Abstract

For many years, machine learning-based *ab initio* gene finding approaches have been the central components of eukaryotic genome annotation pipelines, and they remain so today. The reliance on these approaches was originally sustained by the high cost and low availability of gene expression data, a primary source of evidence for gene annotation along with protein homology. However, innovations in modern sequencing technologies have revolutionized the acquisition of abundant gene expression data, allowing us to rely more heavily on this class of evidence. In addition to gene expression data, proteins found in a multitude of well-annotated genomes represent another invaluable resource for gene annotation. Existing annotation packages often underutilize these data sources, which prompted us to develop EviAnn (Evidence-based Annotation), a novel evidence-based eukaryotic gene annotation system. EviAnn takes a strongly data-driven approach, building the exon-intron structure of genes from transcript alignments or protein-sequence homology rather than from purely *ab initio* gene finding techniques. We show that when provided with the same input data, EviAnn consistently outperforms current state-of-the-art packages including BRAKER3, MAKER2, and FINDER, while utilizing considerably less computer time. Annotation of a mammalian genome can be completed in less than an hour on a single multi-core server. EviAnn is freely available under an open-source license from https://github.com/alekseyzimin/EviAnn_release.

Introduction

Two of the most reliable sources of evidence for gene annotation are gene expression data and protein homology. Gene expression data can be captured with very high-throughput RNA sequencing (RNA-seq) technology, which sequences all the genes being expressed in a tissue sample and can easily generate tens of millions of reads per sample. These short reads can be aligned to a genome and then assembled to create an accurate picture of the exon-intron structures of all the genes and splicing variants present in the original tissue. RNA-seq technology has been augmented in recent years with the introduction of long-read sequencing and even direct sequencing of RNA, which can capture full-length transcripts and thereby provide more accurate pictures of the exon-intron structures found in expressed genes. Along with these advances in transcriptome sequencing, the scientific community has now assembled and annotated genomes for thousands of different species. These genomes provide an additional valuable source of annotation evidence, particularly in the form of protein sequences that are well conserved even among distant species.

Transcriptome sequencing protocols directly capture and sequence the mature messenger RNA in a cell, providing some of the best evidence for gene transcripts. However, most of the RNA-seq technologies do not recover complete transcripts, meaning that the sequence fragments need to be assembled,

which in turn may introduce errors. In addition, an RNA-seq experiment only measures the genes expressed in one tissue (or one cell, in some cases), and thus RNA from multiple tissues and conditions is needed to capture all genes and transcripts for a comprehensive annotation.

Most of the current genome annotation systems, including MAKER2 (Cantarel et al., 2008; Holt and Yandell, 2011), BRAKER3 (Gabriel et al., 2024), GeneMark-ETP (Bruna et al., 2024), and FINDER (Banerjee et al., 2021), do not use input evidence directly to produce gene annotation. Instead, they use transcript and protein sequence evidence to train *ab initio* gene finders such as GeneMark, SNAP (Korf, 2004), or AUGUSTUS (Stanke et al., 2006; Hoff and Stanke, 2019). The annotation systems then use the predictions made by these computational gene finders as the basis for their gene models. This approach was originally developed prior to the invention of next-generation RNA sequencing, at a time when expression data was expensive, few annotated genomes were available, and the number of known proteins was relatively small compared to today. The use of an *ab initio* gene finder could, in principle, compensate for gaps in RNA-sequencing data coverage by using machine-learning models to find genes at locations that are not expressed. However, despite many years of algorithm development, *ab initio* gene finding is still highly error-prone for eukaryotic genomes, in which most of the sequence does not encode proteins. In particular, *ab initio* gene finders frequently generate very large numbers of false positive predictions, including erroneous exon-intron combinations and entirely erroneous gene loci. FINDER or BRAKER3 utilize protein homology from closely related species to filter and improve upon predictions made by their *ab initio* gene finders, but homology information cannot always fix all the errors. A further drawback is that *ab initio* gene finders typically only look for protein-coding sequences, which means they simply do not annotate untranslated regions of transcripts (UTRs) or long non-coding RNAs, the latter of which are a vital part of genome annotation today (Schuster and Hsieh, 2019; Cenik et al., 2020; Chatterjee et al., 2001).

In this paper, we describe a novel approach to genome annotation that relies solely on direct evidence of expression or strong protein sequence similarity, avoiding the use of *ab initio* gene finding techniques. This approach is based on accurate processing and cross-referencing of different types of evidence, resulting in a faster and more transparent annotation process, where one can trace the origin of every annotated transcript or coding sequence (or CDS) back to the input data. We have implemented this approach in an open-source software package called EviAnn (Evidence-based Annotation). EviAnn produces genome annotation by combining transcript assemblies created from Illumina RNA-seq data, transcripts from closely related species (if available), and protein sequences from at least one other related species. Because protein sequences are well conserved across the tree of life, if proteins from a closely related species are not available, EviAnn can instead use proteins from curated protein databases such as UniProt (Bairoch et al., 2005). EviAnn's ability to use transcripts and proteins from related species also enables it to "lift over" (or transfer) an annotation from a well-annotated genome to the genome of a close relative.

In this study, we compare EviAnn to current state-of-the-art annotation pipelines including MAKER2, BRAKER3, and FINDER on multiple plant and animal species, including *Arabidopsis thaliana*, *Drosophila melanogaster*, *Populus trichocarpa* (poplar tree), *Danio rerio* (zebrafish), *Gallus gallus* (chicken), and *Mus musculus* (mouse). Our results demonstrate that EviAnn consistently produces gene annotations with superior accuracy and completeness when run on the same data, outperforming all other annotation pipelines. For most transcripts, EviAnn also annotates the 5' and 3' UTR regions using the RNA-seq or transcript-based evidence provided.

Results

Annotation of genomes with RNA-seq and protein data. We compared the annotation produced by EviAnn to three widely used or recently published annotation packages: MAKER2, FINDER, and BRAKER3. For these comparisons, we used data sets from the genomes of six plants and animals: *A. thaliana*, *D. melanogaster*, *P. trichocarpa*, *D. rerio*, *G. gallus*, and *M. musculus*. Five of these genomes represent model organisms that have been well-studied, and their assemblies and annotations have been thoroughly manually curated. The one exception is the poplar tree, *P. trichocarpa*, for which the annotation was created by the Gnomon pipeline at NCBI. We included this genome to illustrate how consistent the automated annotation pipelines are with the annotation produced by NCBI.

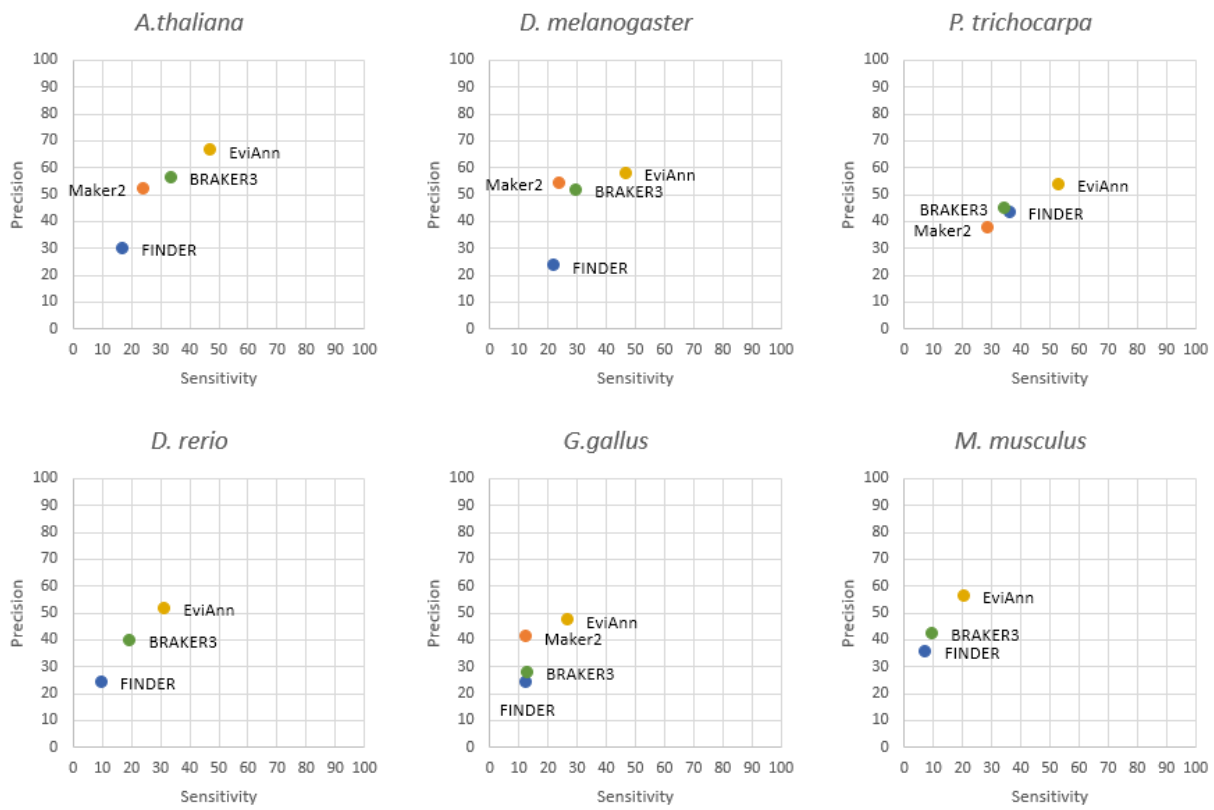

Figure 1. Transcript annotation sensitivity and precision for transcripts in six organisms. Better results are closer to the upper right corner. A transcript was considered correct if all of the introns precisely matched the reference annotation. The low values for sensitivity observed across all packages are likely due to missing annotations of alternative isoforms. MAKER2 results are not shown for *D. rerio* and *M. musculus* because MAKER2 failed to complete the annotation in 1 month on a 24-core Intel Xeon Gold server.

EviAnn annotates both protein-coding genes and long non-coding RNAs (lncRNAs), and therefore we used reference annotations of both types of genes in our evaluations, as opposed to using only protein-coding genes, as has been done in many previous comparisons (Gabriel et al., 2024; Bruna et al., 2024; Banerjee et al., 2021). As inputs to train each of the annotation systems, we used the "close relatives"

data sets from Gabriel et al., 2024. These data sets contain RNA sequencing data available from the NCBI SRA database plus proteins from several closely-related species. We list the accession numbers for each data set in the Data availability section.

To process RNA-seq reads, EviAnn uses HISAT2 (Kim et al., 2019) to align the reads to the target genome. We provided the same alignments (as BAM files) to BRAKER3 and FINDER. Note that MAKER2 is unable to use aligned RNA-seq data directly. Instead, it uses transcripts as evidence for its initial evidence-based annotation pass. To enable this, we assembled the aligned reads into transcripts with StringTie2 (Kovaka et al., 2019), merged these transcripts with the StringTie2 merge command, and then provided the resulting set of transcripts as "EST" evidence to MAKER2. The MAKER2 pipeline also requires manual intervention to train an external gene finder; in our case, we followed the MAKER2 protocol paper (Campbell et al., 2014) and trained SNAP [REF]. For convenience and for better performance in terms of both speed and accuracy, we developed an automated script to run MAKER2 in parallel on a single multi-core computer. This script, which we called EZMAKER, is included in the EviAnn distribution package as `ez_maker.sh`, and serves as a wrapper to run two-pass annotation (using the

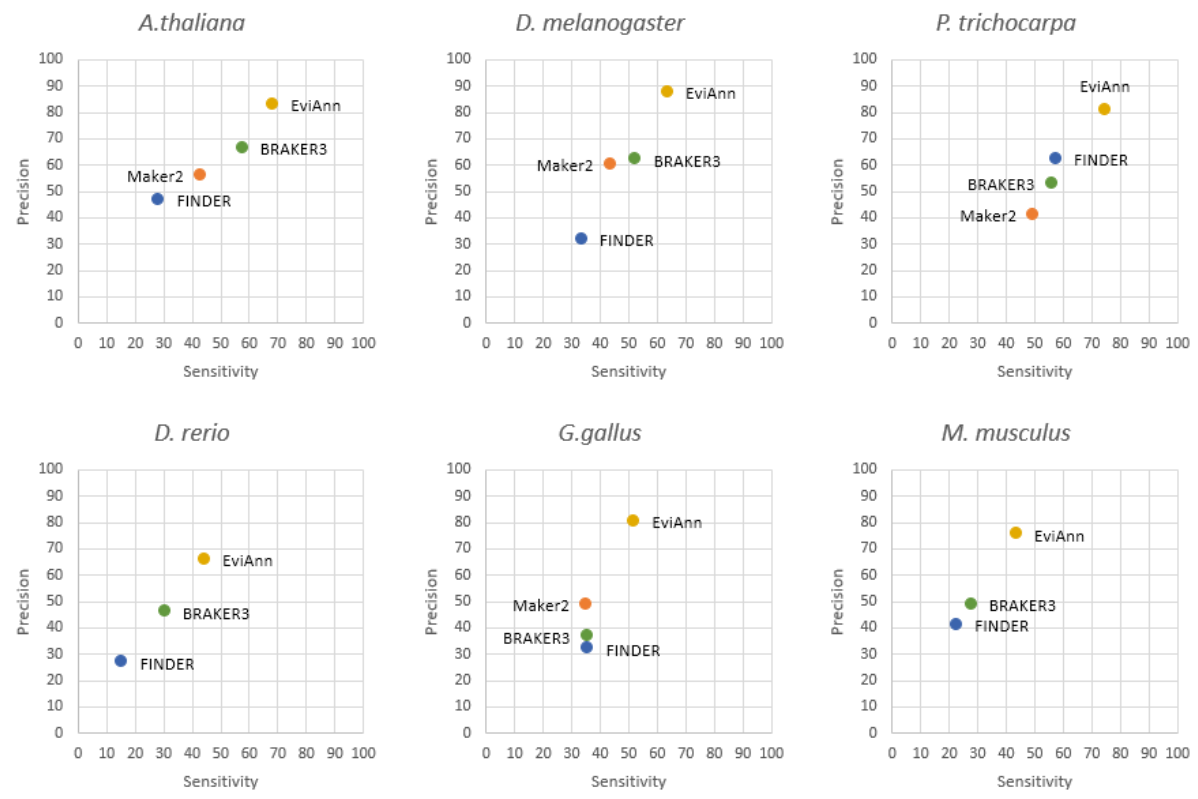

Figure 2. Sensitivity and precision for annotations at the gene level in six species. A gene locus was counted as correct if at least one transcript or CDS at that locus matched a reference transcript at the same locus. Better results are closer to the upper-right corner.

-d switch, as we did here) with MAKER2, incorporating the SNAP *ab initio* gene finding step. This wrapper makes it easier to reproduce the results described here, and in addition it makes MAKER2 faster and easier to run on a single multi-core server. We list the command lines to generate all annotations in the Supplementary Materials.

We used gffcompare (Pertea and Pertea, 2020) to compare the automatically generated annotations to the protein-coding and long non-coding RNA genes in the RefSeq annotations of the reference species. Note that our evaluation specifically assessed the annotated transcripts and CDSs against the complete RefSeq transcripts. Thus, if an annotated transcript was missing an exon in either the 5' or 3' UTR regions, it was not considered a match. We did, however, allow mismatches of up to 100bp in the transcriptional start position and the transcription termination site. We evaluated accuracy on protein-

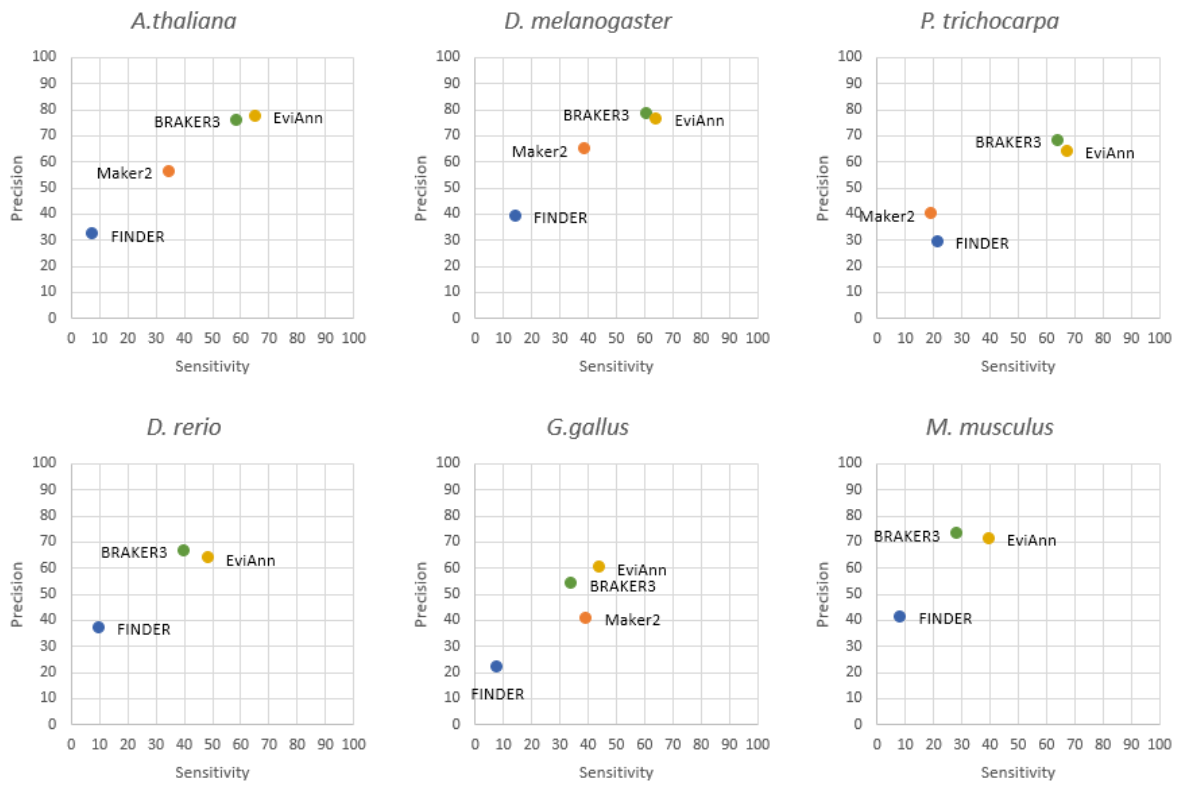

Figure 3. Sensitivity and precision for annotations of protein-coding sequences in six species. A protein-coding annotation was considered correct if all the introns between protein-coding regions (CDSs) precisely matched the reference annotation. Noncoding exons were not counted for this evaluation. Better results are closer to the upper-right corner.

coding sequences (CDSs) separately.

We assessed the annotations produced by the different pipelines using two measures: sensitivity (Sn = number of features correctly annotated/number of total correct features) and precision (Pr = number of features correctly annotated/number of total annotated features). We considered three feature categories: transcripts, CDSs, and genes (or gene loci). A gene was considered correctly annotated if at least one transcript or CDS at the gene locus matched the reference annotation, as determined by either an intron chain match (for multi-exon transcripts) or exon coordinate overlap (for intronless transcripts). A CDS match was considered valid only if the start and end coordinates and all intron coordinates matched exactly. Note that in Figures 1-3, we do not show MAKER2 results for *D. rerio* and *M. musculus* because MAKER2 failed to complete the annotation in one month on a 24-core Intel Xeon Gold server.

Figure 1 shows that at the level of transcripts annotated by the four packages, EviAnn consistently had the highest sensitivity and precision across all six model organisms. We note that all pipelines demonstrated fairly low sensitivity due to the relatively small amount of RNA-seq data used; while the reference annotation often contains multiple transcripts per gene, in most cases either zero or one isoform was present per gene in the RNA-seq data. Figure 2 shows the sensitivity and precision at the level of gene loci. Here we count a gene locus as being annotated correctly if at least one transcript or CDS at that locus matched a transcript from the reference. Not surprisingly, we observe much higher sensitivity across all test data sets. Figure 3 shows the sensitivity and precision when considering only protein-coding regions (CDSs) annotated by the four programs. Here we counted each distinct CDS only once; i.e., if a gene locus contained multiple transcripts with the same CDS, they all counted as one. In all our experiments, EviAnn had higher sensitivity than any of the other methods, usually by at least 10% and sometimes much more. In nearly all comparisons, EviAnn also had the highest precision, with the exception of CDS prediction, where its performance was roughly similar to the performance of BRAKER3.

To determine whether EviAnn would produce more comprehensive annotation when provided with more RNA-seq data, we ran a separate experiment for *A. thaliana* using 116 RNA-seq samples, along with the same set of proteins from related species. With this larger dataset, EviAnn's accuracy on transcripts increased dramatically, from 47% sensitivity and 66% precision (Figure 1) to 71% and 75%, respectively.

It is important to note that FINDER and BRAKER3 rely on the AUGUSTUS and GeneMark *ab initio* gene finders, neither of which can annotate UTR regions of genes. Consequently, all "exon" features in the output files produced by FINDER and BRAKER3 correspond only to the protein-coding portions of exons. Virtually all transcripts in the organisms studied here begin and end with untranslated regions (UTRs), with coding sequences (CDSs) starting in the first or subsequent exons and ending with a stop codon, which typically occurs in the last exon but can occur earlier. Annotation files normally describe the coding regions as CDS features that are separate from exon features, although their positions overlap and internal coding exon boundaries are often identical to CDS boundaries. In recent publications describing these systems, the authors reported transcript-level accuracy based only on the coding sequences, incorrectly inflating accuracies when in fact many transcripts were entirely missing their untranslated regions (UTRs). In our comparisons here, we have corrected this reporting error, and in Figures 1 and 3 we evaluate accuracy on transcripts separately from accuracy on CDS regions. In addition to the inability of FINDER and BRAKER3 to annotate the UTRs that flank most eukaryotic transcripts on both ends, they also lack the ability to annotate any long non-coding RNA genes, which number in the thousands for many plants and animals. By using evidence from RNA-seq data, EviAnn also annotates long non-coding RNAs, which contributes to its higher sensitivity on transcripts.

Methods

Input data. As part of the process for annotating a genome, EviAnn uses the genome sequence itself, RNA-seq data or transcripts from one or more closely related species, and alignments of protein sequences to the genomic DNA. RNA-seq data must be grouped in such a way that each input file (or pair of files in the case of paired-end sequencing) represents a single RNA-seq experiment; e.g. a single condition for a single tissue. EviAnn supports Illumina RNA-seq, PacBio IsoSeq, and Oxford Nanopore direct RNA/cDNA sequencing data, as well as mixed data sets, where the same sample has been sequenced by both short- and long-read technologies. EviAnn can also use transcripts from different but closely related species, which must be supplied in separate files. EviAnn supports both fasta/fastq and

BAM file inputs. For fastq inputs, EviAnn uses HISAT2 to align short-read sequencing data to the genome, and it uses minimap2 to align transcripts and/or long reads. EviAnn uses the resulting spliced alignments of reads and transcripts as input to the transcriptome assembly program StringTie2 (Kovaka et al., 2019). Alignments from short-read RNA-seq experiments are assembled separately from long read data. For RNA-seq experiments, EviAnn only keeps transcripts that are present in at least 2% of the input experiments and with TPM or FPKM values of at least 0.5. EviAnn then uses gffcompare (Pertea and Pertea, 2020) to merge all transcript assemblies, eliminating redundant and exactly contained transcripts, into a single candidate transcript file. After merging, EviAnn assigns a TPM value to each transcript, equal to the maximum TPM value for the transcript across all input samples, and encodes this value along with the number of samples in the transcript name.

EviAnn uses protein sequences from one or several closely related species to find genes that are not expressed in the transcriptome sequencing data, to correct errors in the assembled transcripts, to identify and filter out non-functional transcripts, and to assign coding sequences to the protein-coding transcripts. Proteins are aligned to genomic DNA using miniprot (Li et al, 2023). Figure 4 shows an overview of the EviAnn annotation pipeline. Below we provide detailed descriptions of the steps.

Merging transcript and protein alignments. EviAnn runs gffcompare with the protein alignment file (which shows how protein sequences align to the sequence of the reference genome) as a guide to assign an aligned CDS to each assembled transcript. EviAnn then examines all transcripts that share at least one intron junction (or that overlap, in the case of intronless transcripts) with the protein alignment. EviAnn may keep more than one transcript corresponding to the same protein if the isoforms have differences in the UTR regions. If there are multiple transcripts that correspond to the same protein at the same locus, EviAnn computes a transcript "reliability" score R for all such transcripts. R is computed as $N*log_2(2+T)$, where N is the number of RNA-seq experiments in which the transcript was observed, and T is the TPM value. This score gives greater weight to transcripts that are expressed in multiple tissues and at higher levels. For each gene, EviAnn computes a maximum score R_{max}, and keeps only the transcripts whose R score is at least $\sqrt{R_{max}}$, a threshold that was chosen empirically. A small portion of the transcripts may end up without any corresponding intron chain matches to an aligned protein, and we label these as “potentially noncoding” transcripts.

At the completion of this stage, we have three categories of genes:

1. Genes with transcripts that have a complete intron chain match to a CDS from an aligned protein ("complete");
2. Genes with transcripts where none of the transcripts have a corresponding protein alignment ("transcript-only"); and
3. Genes whose coding sequences are annotated based only on protein alignments, with no RNA-seq support and no UTR regions ("protein-only").

Next, we screen the transcripts and CDSs obtained from protein alignments to remove transcripts with low-quality splice sites. We model the patterns around the donor and acceptor splice sites using Markov chains, as follows. First, we label as *reliable* all complete transcripts that do not have introns in the UTR regions, and whose CDSs contain both start and stop codons and have no frameshifts or in-frame stops. We used gffcompare to compute intron precision for the reliable transcripts in all data sets used for evaluation in this manuscript against the RefSeq annotations and found that intron precision typically exceeds 99.8%. This implies that 99.8% of the introns in the reliable transcripts matched RefSeq introns exactly. We then use all unique introns in reliable transcripts to compute Markov chain model weights for donor and acceptor splice sites for the target genome. For the donor site, we constructed a Markov chain spanning 16 positions, with the consensus GT (the beginning of the intron)

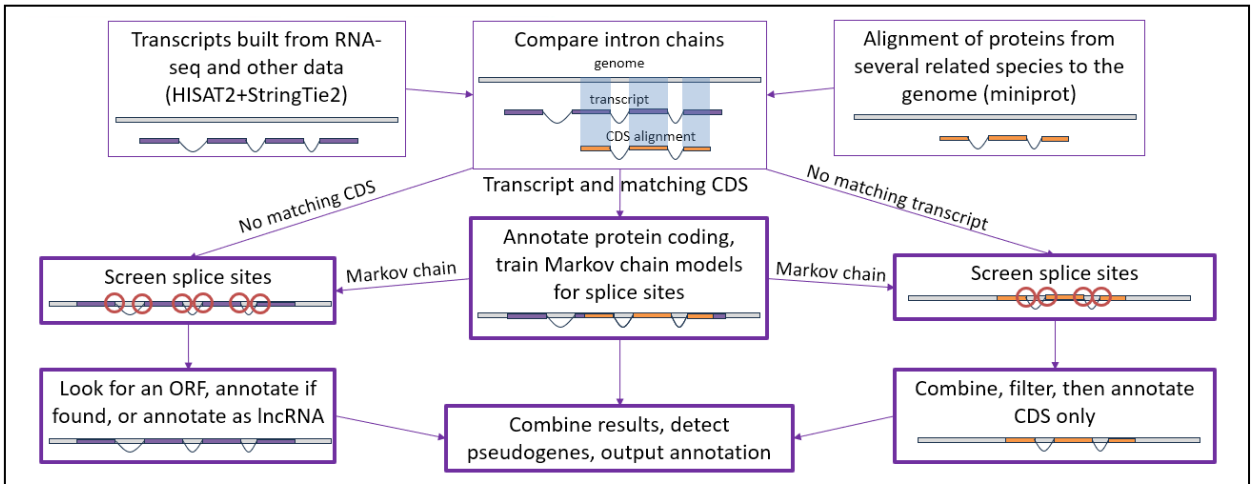

Figure 4. A simplified block diagram of the EviAnn pipeline.

at positions 4-5. For acceptor sites, we created a 30-bp Markov chain where the consensus AG was at positions 26-27. We use higher-order Markov chains when sufficient numbers of splice sites (training data) are available, as follows. If the data contains fewer than 1024 splice sites, we use a 0th-order chain (also referred to as Positional Weight Matrix, or PWM), with 4 probabilities at each position. If the data contains 1024-4096 splice sites, we use a 1st-order Markov chain, with 16 probabilities at each position. If the data contains >4096 sites, we use a 2nd-order Markov chain, with 64 probabilities per position. We then use the Markov chain to score every splice junction in the set of all transcripts and protein alignments. We assign a score to each intron that is equal to the sum of Markov chain scores for donor and acceptor sites for the intron. We then assign a score to each transcript equal to the minimum intron score over all introns in the transcript. We eliminate transcripts and aligned CDSs where the minimum intron score is below a threshold set to 4, which is an empirically determined value that provides the best performance across all test data sets. We note that we do not absolutely require that all introns have the GT-AG consensus sequences, to allow some transcripts with non-canonical splice sites, whose introns otherwise score above the threshold.

Complete gene loci for protein-coding genes have at least one transcript and a CDS alignment that shares at least one splice junction with the transcript or, in case of intronless transcripts, containment of the CDS. If any transcript at a given locus has a matching CDS, then other transcripts that lack CDSs are ignored. Sometimes one or more junctions in an intron chain for a transcript might disagree with its best matching CDS. In the case of disagreements such as this, EviAnn will construct the longest possible transcript model with the largest number of exons derived from the transcript and the CDS alignments,

as shown in Figure 5. For each transcript, EviAnn first assumes that the intron junctions derived from the transcript are correct, and it infers putative start and end coordinates of the open reading frame (ORF) from the corresponding start and stop of the CDS (derived from the protein alignment). EviAnn then verifies that the ORF has no in-frame stop codons, its length is divisible by 3, it starts with a start codon

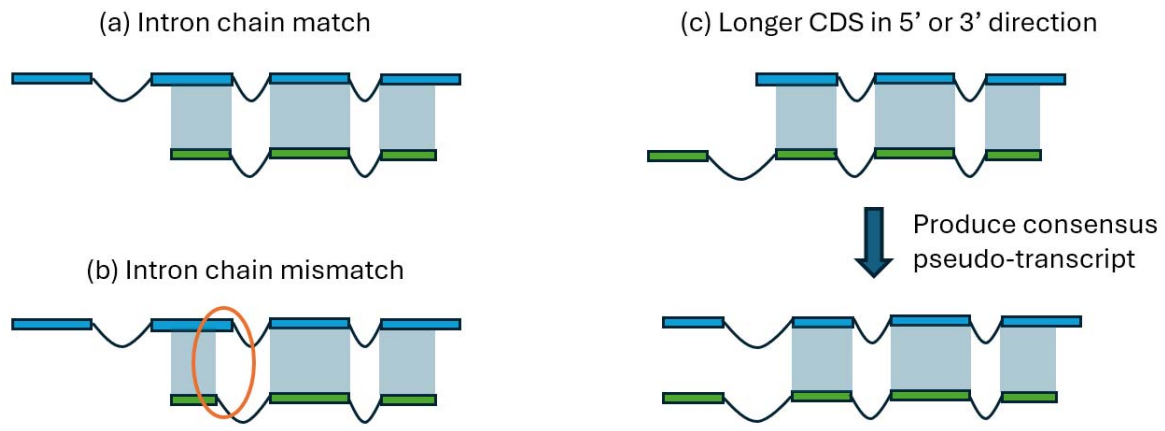

Figure 5. Resolution of conflicts between intron chains of aligned CDSs from related species' proteins (green) and transcripts (blue) assembled from the RNA-seq data. In all cases, we use the frame specified by the start of the aligned CDS and only annotate transcript/CDS pair if we can locate a complete ORF on the transcript/CDS. (a) no conflicts in the intron chain: use the transcript/CDS pair as is, look for a complete ORF. (b) intron chain mismatch: trust the transcript intron chain first, look for a complete ORF on the transcript, annotate if found; delete transcript and use the CDS if no complete ORF is found and there is a complete ORF in the aligned CDS. (c) partial intron chain match: look for the first matching splice junction between the CDS and the transcript and produce a consensus pseudo-transcript using exons from the transcript and CDS. Not pictured here is the scenario where the transcript is contained in the CDS, we simply discard the transcript and use the exons from the CDS.

(ATG), and ends with a stop codon (TAA, TAG, or TGA). We observed that most ORFs have correct start and stop codons and no in-frame stops at this step.

If an ORF has an in-frame stop or its length is not divisible by 3, EviAnn uses TransDecoder (Haas, 2023) to look for a complete ORF that has an amino-acid match to the original aligned protein. If TransDecoder is unable to find an ORF that satisfies these criteria, the transcript is discarded. If a start codon is missing, EviAnn searches for the furthest start codon in the 5' direction, starting with the implied beginning of the CDS, and if not found, it searches in the 3' direction. If a start codon is found and the stop codon is missing, EviAnn looks along the transcript sequence for a first stop codon downstream from the start codon. No frameshifts are allowed when searching for start or stop codons. If EviAnn fails to find a complete ORF, and there was a disagreement between the intron chains for the transcript and its best matching CDS, it discards the transcript and attempts to find a complete ORF in the aligned CDS alone, repeating the above process for the CDS.

For transcript-only gene loci; i.e., those with no protein sequence matches, EviAnn considers whether these might contain novel protein-coding genes or instead represent noncoding RNA genes. EviAnn uses TransDecoder to find complete ORFs in these transcripts, and if it finds an ORF spanning > 75% of the transcript, that gene is annotated as protein-coding, and all other transcripts at that locus that do not contain a CDS are discarded. If none of the transcripts at a locus have a valid ORF that satisfies the above conditions, EviAnn declares the locus to be non-coding and annotates all transcripts at that locus as long non-coding RNAs.

At protein-only loci, we annotate genes based solely on the protein alignments. Because these loci do not have transcript alignments, any transcripts that we annotate will be missing their UTRs. For this class of genes, we only use protein alignments that yield a complete open reading frame, one that spans the entire source protein without any frameshifts in the coding region.

Finally, we look for UTR regions that overlap the CDS regions of other genes. Transcripts with such UTRs may be mis-assembled or *readthrough* transcripts that include parts of two neighboring genes. For these cases, EviAnn splits the readthrough transcripts into chunks, each containing a corresponding CDS.

EviAnn combines the transcript, CDS, and UTR annotations to produce the final annotation. Table 2 provides a breakdown of the numbers of each transcript type for the six species used in our experiments. In most cases, both transcript and protein evidence were used by EviAnn, and the number of transcript-only gene loci was relatively small. A higher number of transcripts with protein-only evidence indicates that the RNA-seq data was not sufficiently deep to capture the full transcriptome of that species. EviAnn includes a label in its output file that indicates the type of evidence used for each annotated transcript, as well as a list of the IDs of transcripts and proteins used as evidence for each annotation.

Table 2. Breakdown of the number of protein-coding transcripts annotated with different evidence types for six plant and animal genomes. For all species, most transcripts and CDSs are supported by both transcript and protein evidence.

Genome	**Number of transcripts or coding sequences**		
	Evidence: complete (UTRs annotated)	**Evidence: transcript only (UTRs annotated)**	**Evidence: protein only (UTRs not annotated)**
A. thaliana	29608	74	10273
D. melanogaster	21799	299	2857
P. trichocarpa	41881	111	13107
D. rerio	21262	180	18189
G. gallus	36941	224	7996
M. musculus	27874	192	16632

Processed pseudogene detection and optional functional annotation. Processed pseudogenes are copies of spliced mRNAs (i.e., transcripts with introns removed) that have been incorporated into the genome sequence. While some of these genes may have a valid ORF, they are generally non-functional and should not be annotated as protein-coding genes. At the last step of the EviAnn annotation, we used NCBI blastp (Camacho et al., 2009) to align all proteins from intronless transcripts to all proteins coded by multi-exon transcripts. If an intronless protein aligns with >90% similarity over >90% of its length to a multi-exon protein, we label it as a pseudogene, adding a “pseudo-true” label, and EviAnn does not output CDS records for these transcripts.

Optionally, if the user provides the -f switch to EviAnn, it aligns all annotated proteins to the UniProt-SwissProt protein database with blastp. We then use the name of the best matching protein to assign a name to the newly annotated protein, with the note "similar to" prepended.

Timings. We measured timings for EviAnn, BRAKER3, MAKER2, and FINDER on a Linux server equipped with two 12-core Intel Xeon Gold 6248R CPUs and 1 TB of RAM. We ran all programs using 24 cores. Table 3 lists the timings for EviAnn and other programs. These timings exclude the step of aligning RNA-seq reads to the genome, which was shared by all programs. EviAnn ran the fastest on all genomes, performing from 14 times faster (on mouse, as compared to FINDER) to hundreds of times faster (as compared to MAKER2).

EviAnn's core algorithms, which produce the annotation by examining transcripts and proteins, are very fast, taking up less than 1% of the total run time. The most time-consuming step in EviAnn's execution is alignment of proteins to the genome, which is done very efficiently with miniprot.

Table 3. Wall-clock time in hours for the annotation pipelines EviAnn, BRAKER3, FINDER, and MAKER2. The timings exclude the time spent on aligning RNA-seq reads to the genome. The fastest result is shown in bold. MAKER2 results for *D. rerio* and *M. musculus* are not shown because the program did not finished after more than one month of run time.

	A. thaliana	*D. melanogaster*	*D. rerio*	*P. trichocarpa*	*G. gallus*	*M. musculus*
EviAnn	**0.3**	**0.3**	**1.0**	**0.4**	**0.4**	**0.8**
BRAKER3	13.2	7.3	29	10.8	72	51
FINDER	6.8	8.4	63	9.5	38	11
MAKER2	168	34	N/A	81	148	N/A

Discussion

EviAnn uses a combination of transcription evidence and protein sequence alignments to produce automated, high-quality eukaryotic genome annotation, outperforming most existing packages in almost all measures of accuracy. EviAnn's ability to use transcripts and proteins from close relatives enables the annotation of novel genomes even when expression data is unavailable. With the ever-increasing number of well-assembled and well-annotated genomes available from public databases, we expect EviAnn to become even more useful over time.

EviAnn was designed with the goals of transparency and efficiency in mind. When EviAnn is provided assembled transcripts as input, any transcript that aligns well with the DNA of the target genome, and that has a valid open reading frame whose translation matches a known protein, will be included in the output. The system's output includes labels that describe the evidence that was used to support each transcript, allowing users to easily see whether a particular transcript is supported by expression data, by protein alignments, or both.

Data Availability

All data used in this manuscript is publicly available from the NCBI RefSeq, GenBank, or SRA databases. We list the specific accession numbers for all data sets used for the evaluations in Supplementary Table 1.

Acknowledgements

This work was supported in part by NSF grant IOS-2432298, and by NIH grants R01-HG006677, R35-GM130151, and R35-GM156470.

References

1. Schuster SL, Hsieh AC. The untranslated regions of mRNAs in cancer. Trends in cancer. 2019 Apr 1;5(4):245-62.
2. Cenik C, Derti A, Mellor JC, Berriz GF, Roth FP. Genome-wide functional analysis of human 5'untranslated region introns. Genome biology. 2010 Mar;11:1-7.
3. Chatterjee S, Rao SJ, Pal JK. Pathological mutations in 5′ untranslated regions of human genes. eLS. 2001:1-8.
4. Cantarel BL, Korf I, Robb SM, Parra G, Ross E, Moore B, Holt C, Alvarado AS, Yandell M. MAKER: an easy-to-use annotation pipeline designed for emerging model organism genomes. Genome research. 2008 Jan 1;18(1):188-96.
5. Holt C, Yandell M. MAKER2: an annotation pipeline and genome-database management tool for second-generation genome projects. BMC bioinformatics. 2011 Dec;12(1):1-4.
6. Hoff KJ, Lange S, Lomsadze A, Borodovsky M, Stanke M. BRAKER1: unsupervised RNA-Seq-based genome annotation with GeneMark-ET and AUGUSTUS. Bioinformatics. 2016 Mar 1;32(5):767-9.
7. Lukashin AV, Borodovsky M. GeneMark. hmm: new solutions for gene finding. Nucleic acids research. 1998 Feb 1;26(4):1107-15.
8. Besemer J, Lomsadze A, Borodovsky M. GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions. Nucleic acids research. 2001 Jun 15;29(12):2607-18.
9. Brůna T, Lomsadze A, Borodovsky M. GeneMark-EP+: eukaryotic gene prediction with self-training in the space of genes and proteins. NAR genomics and bioinformatics. 2020 Jun;2(2):lqaa026.
10. Brůna T, Lomsadze A, Borodovsky M. GeneMark-ETP significantly improves the accuracy of automatic annotation of large eukaryotic genomes. Genome Research. 2024 Jun 12.
11. Banerjee S, Bhandary P, Woodhouse M, Sen TZ, Wise RP, Andorf CM. FINDER: an automated software package to annotate eukaryotic genes from RNA-Seq data and associated protein sequences. BMC bioinformatics. 2021 Dec;22:1-26.
12. Korf I. Gene finding in novel genomes. BMC bioinformatics. 2004 Dec;5(1):1-9.
13. Stanke M, Keller O, Gunduz I, Hayes A, Waack S, Morgenstern B. AUGUSTUS: ab initio prediction of alternative transcripts. Nucleic acids research. 2006 Jul 1;34(suppl_2):W435-9.
14. Hoff KJ, Stanke M. Predicting genes in single genomes with AUGUSTUS. Current protocols in bioinformatics. 2019 Mar;65(1):e57.
15. Bairoch A, Apweiler R, Wu CH, Barker WC, Boeckmann B, Ferro S, Gasteiger E, Huang H, Lopez R, Magrane M, Martin MJ. The universal protein resource (UniProt). Nucleic acids research. 2005 Jan 1;33(suppl_1):D154-9.
16. Gabriel L, Brůna T, Hoff KJ, Ebel M, Lomsadze A, Borodovsky M, Stanke M. BRAKER3: Fully Automated Genome Annotation Using RNA-Seq and Protein Evidence with GeneMark-ETP, AUGUSTUS and TSEBRA. bioRxiv. 2023 Jun 12.
17. Kim D, Paggi JM, Park C, Bennett C, Salzberg SL. Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. Nature biotechnology. 2019 Aug;37(8):907-15.
18. Kovaka S, Zimin AV, Pertea GM, Razaghi R, Salzberg SL, Pertea M. Transcriptome assembly from long-read RNA-seq alignments with StringTie2. Genome biology. 2019 Dec;20(1):1-3.

19. Campbell, M. S., et al. 2014. Genome Annotation and Curation Using MAKER and MAKER-P. Curr. Protoc. Bioinform. 48:4.11.1-4.11.39
20. Pertea G, Pertea M. GFF utilities: GffRead and GffCompare. F1000Research. 2020;9.
21. Gertz EM, Yu YK, Agarwala R, Schäffer AA, Altschul SF. Composition-based statistics and translated nucleotide searches: improving the TBLASTN module of BLAST. BMC biology. 2006 Dec;4(1):1-4.
22. Gotoh O. Direct mapping and alignment of protein sequences onto genomic sequence. Bioinformatics. 2008 Nov 1;24(21):2438-44.
23. Li H. Protein-to-genome alignment with miniprot. Bioinformatics. 2023 Jan 1;39(1):btad014.
24. Haas, BJ. https://github.com/TransDecoder/TransDecoder
25. Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, Madden TL. BLAST+: architecture and applications. BMC bioinformatics. 2009 Dec;10:1-9.