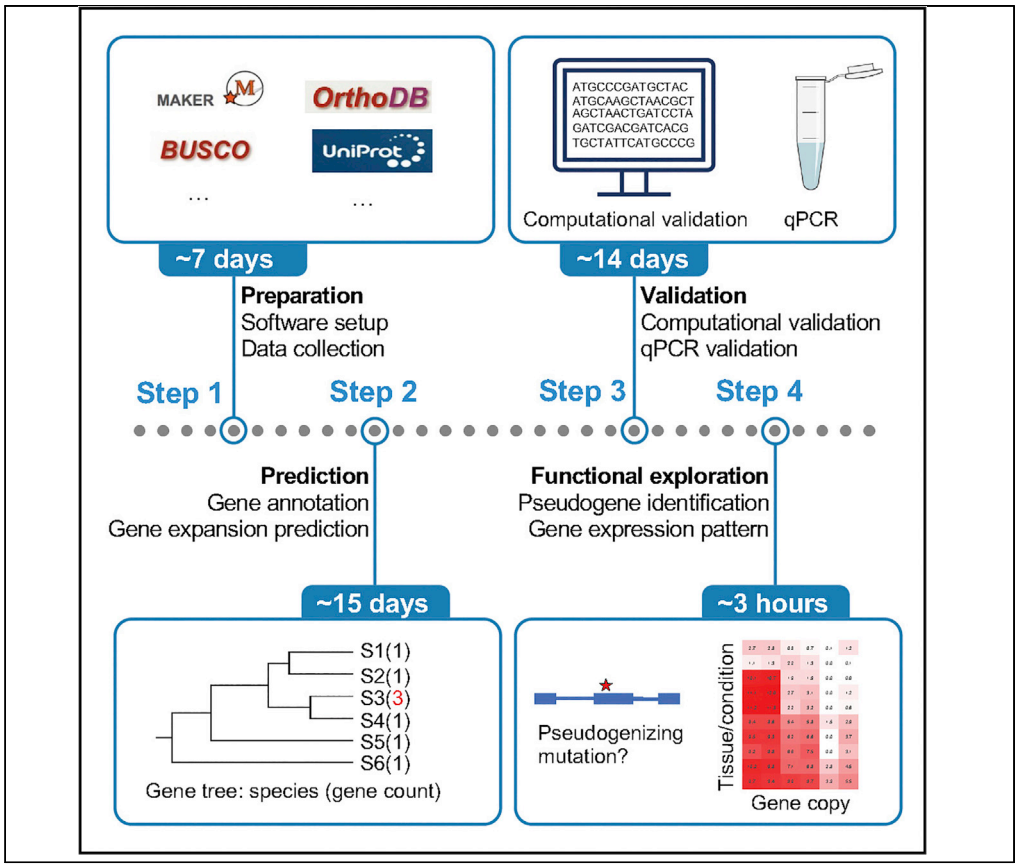
STAR Protocols

CellPress
OPEN ACCESS

Protocol

Protocol for gene annotation, prediction, and validation of genomic gene expansion



Although gene expansion plays an important role in evolution, its identification remains a challenge due to potential errors in genome assembly and annotation. Here, we describe a detailed step-by-step protocol for gene annotation, prediction of genomic gene expansion, and its computational and experimental validation. Finally, we also detail steps to discover functionality of each copy of replicated genes.

Publisher's note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.

Quanwei Zhang, Zhengdong D. Zhang

qwzhang0601@gmail.com (Q.Z.)
zhengdong.zhang@einsteinmed.edu (Z.D.Z.)

Highlights

Description of pipeline of *de novo* genome annotation

Detailed procedure to identify genomic gene expansion

Computational and experimental approach for validation of gene expansion

Protocol for functional exploration of replicated gene copies

Zhang & Zhang, STAR Protocols 4, 101692 December 16, 2022
https://doi.org/10.1016/j.xpro.2022.101692

STAR Protocols

CellPress
OPEN ACCESS

Protocol

Protocol for gene annotation, prediction, and validation of genomic gene expansion

Quanwei Zhang[1,2,*] and Zhengdong D. Zhang[1,3,*]

[1]Department of Genetics, Albert Einstein College of Medicine, Bronx, NY, USA

[2]Technical contact

[3]Lead contact

*Correspondence: qwzhang0601@gmail.com (Q.Z.), zhengdong.zhang@einsteinmed.edu (Z.D.Z.)
https://doi.org/10.1016/j.xpro.2022.101692

SUMMARY

Although gene expansion plays an important role in evolution, its identification remains a challenge due to potential errors in genome assembly and annotation. Here, we describe a detailed step-by-step protocol for gene annotation, prediction of genomic gene expansion, and its computational and experimental validation. Finally, we also detail steps to discover functionality of each copy of replicated genes. For complete details on the use and execution of this protocol, please refer to Zhang et al. (2021).

BEFORE YOU BEGIN

Before you start the protocol, you should have the genome assembly ready to be annotated. For genome assembly, we suggested to combine both short sequencing reads and long-read sequencing data, which can improve the chance to identify gene expansion (Zhou et al., 2020). RNA-seq data from different tissues of a few biological replicates will help to validate the prediction and explore functionalities of each copy of expanded genes across tissues. Please note that this protocol is intended for studies of mammalian genomes.

Software setup

Timing: ~ 1 week

The user should work with their system administrator to install the following software tools for their specific computational environment. Methods S1 has a list of dependencies.

1. Set up Maker2 pipeline.
   a. Install Maker2.
   Register and download Maker2 from https://www.yandell-lab.org/software/maker.html. Then follow the description in the "INSTALL" file from the downloaded package and install the software.

```
# Go to directory ``maker/src'' of downloaded package

# Run the command below to configure

  perl Build.PL

# Install maker2 with the following command

  ./Build install
```

b. Install BUSCO.

Download the BUSCO package (Simao et al., 2015) at https://busco.ezlab.org/ and the latest orthologous gene sets for the corresponding lineage (https://busco-data.ezlab.org/v5/data/lineages/) from OrthoDB (Kriventseva et al., 2019) at https://www.orthodb.org.

```
# Install BUSCO
# Download the package
  git clone https://gitlab.com/ezlab/busco.git
# Install
  cd busco/
  sudo python3 setup.py
# Check installation
  $ busco -h
```

c. Install RepeatMasker.

Prepare RepeatMasker from http://www.repeatmasker.org/ and also RepBase repeat libraries from https://www.girinst.org/.

```
# Install RepeatMasker
# Unpack the package of RepeatMasker
  cp < RepeatMasker.tar.gz> <path_to_install_RepeatMasker>
  cd <path_to_install_RepeatMasker>
  gunzip <RepeatMasker.tar.gz>
  tar xvf <RepeatMasker.tar>
# Unpack RepBase library
  cp <RepBaseRepeat.tar.gz> <path_to_install_RepeatMasker>/RepeatMasker
  cd <path_to_install_RepeatMasker>/RepeatMasker
  gunzip <RepBaseRepeat.tar.gz>
  tar xvf <RepBaseRepeat.tar.gz>
# Set up by run following command
  perl ./configure
```

d. Install RepeatModeler.

For non-model organisms, de novo transposable elements can be constructed by RepeatModeler, which is available at http://www.repeatmasker.org/RepeatModeler/.

```
# Install RepeatModeler
# Unpack the package of RepeatModeler
  tar -zxvf <RepeatModeler-open-#.#.#.tar.gz>
# Go into the folder and configure it
  perl ./configure
```

2. Tools for analysis.
   a. CAFE5.
   Latest CAFE5 is available at https://github.com/hahnlab/CAFE5.

```
# Install CAFE5

# Download the package

# Go into the folder and run the following command to install

  ./configure

  make
```

   b. GeneWise.
   This package is available at ftp://ftp.ebi.ac.uk/pub/software/unix/wise2/.

```
# Install GeneWise

# Download the package run the following command to install

# Binaries are in src/bin after make

  cd <path_wise2>/src

  make all
```

   c. Kallisto.
   The executable package is available at https://pachterlab.github.io/kallisto/.
3. Tools for validation.
   a. Apollo.
   To install this package run the command below. More details can be found at https://genomearchitect.readthedocs.io/en/latest/Apollo2Build.html.

```
# Install Apollo

  git clone https://github.com/GMOD/Apollo.git Apollo
```

   b. Integrative Genomics Viewer (IGV).
   The package is available at https://software.broadinstitute.org/software/igv/download.
   c. STAR.
   The tool for reads alignment can be installed by the command below.

```
# Install STAR

# Download STAR from https://github.com/alexdobin/STAR

  cd <path_STAR>

  make STAR
```

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Reviewed protein sequences	UniProtKB (Swiss-Prot)	https://www.uniprot.org
Software and algorithms		
BUSCO (v5.3.2)	Simao et al. (2015)	https://busco.ezlab.org/

(*Continued on next page*)

Continued

REAGENT or RESOURCE	SOURCE	IDENTIFIER
RepBase (v27.05)	Bao et al. (2015)	https://www.girinst.org/repbase/
RepeatMasker (v4.1.2)	Tarailo-Graovac and Chen (2009)	https://www.repeatmasker.org/
MAKER (v3.01.03)	Holt and Yandell (2011)	https://www.yandell-lab.org/software/maker.html
OrthoDB (v10.1)	Kriventseva et al. (2019)	https://www.orthodb.org
CAFE 5 (v5)	Mendes et al. (2020)	https://github.com/hahnlab/CAFE5/
GeneWise (v2.2.0)	Birney et al. (2004)	ftp://ftp.ebi.ac.uk/pub/software/unix/wise2/
Kallisto (v0.46.1)	Bray et al. (2016)	https://pachterlab.github.io/kallisto/

STEP-BY-STEP METHOD DETAILS

Genome annotation

Timing: days to weeks (depending on genome size and complexity; e.g., ~15 days for the 3-Gb beaver genome on a 10-node computer cluster with 12-GB memory in each node)

In this step, we predict and annotate genes for the de novo genome assembly of a target species, the focus of a particular genome annotation study. Besides Maker2 pipeline described here, researchers can also consider other pipelines as alternative choice, e.g., BRAKER2 (Bruna et al., 2021), the Ensemble gene annotation system (Aken et al., 2016) etc.

1. Mask repetitive elements in the genome.
   a. RepeatMasker.
      i. Construct species-specific repetitive elements by RepeatModeler (version 1.0.10). Details about this step can be find at http://weatherby.genetics.utah.edu/MAKER/wiki/index.php/Repeat_Library_Construction–Basic.

```
# Collect repetitive elements

# Input: genomic sequences

# Output: file ''consensi.fa.classified'', containing the receptive sequences

# path_RM: path of RepeatModeler

# -pa N: how many cores to run

# -engine ncbi: refers to blast program for alignment

  <path_RM>/BuildDatabase -name seqfiledb -engine ncbi <genome.fa>

  <path_RM>/RepeatModeler -database seqfiledb -pa N > seqfile.out
```

      ii. Mask repeat elements by RepeatMasker (Tarailo-Graovac and Chen, 2009), with RepBase repeat libraries (Bao et al., 2015), together with specifies-specific repeating elements constructed in the previous step. Repbase stores consensus sequences of repetitive elements among species. This step masks both common repeat elements among species and species-specific repetitive elements in the assembly.
   b. RepeatRunner.

RepeatRunner is a part of the Maker2 pipeline, which also comes with a list of default transposable protein elements in the FASTA format. Maker2 will automatically search for more divergent transposable protein elements.

Note: Repetitive elements are enriched throughout the genome. Such repetitive elements can cause non-specific gene hits during annotation. By masking repetitive elements, annotation tools can target gene encoding regions more easily.

2. Train the models.
   a. Train Augustus.

```
# Train Augustus

# Input: genome assembly

# Output: trained model

# –long: performs full optimization for Augustus training

  python <directory_of_BUSCO>/BUSCO.py –cup <number_thread> –in <genome_assembly>.fa –out
<output_name> –lineage <directory_BUSCO_lineage_data> –mode genome –long
```

Note: The "–long" parameter turns on Augustus optimization mode for self-training, which can improve the accuracy for non-model organisms. The latest lineage data are available at https://busco-data.ezlab.org/v5/data/lineages/.

   b. Train SNAP.

The details of training SNAP can be found in (Campbell et al., 2014), which recommends three iterations of training. The trained parameter/HMM file from the current round can be used to seed the next round of training.

```
# Train SNAP

# (a) Generate MAKER control files

# Generate three files with suffix ''.ctl'', through which to provide user input

  maker -CTL

# (b) Edit maker_opts.ctl file to provide input parameters

genome=<genome_assembly.fa>

# choose either eukaryotic or prokaryotic

organism_type=<eukaryotic|prokaryotic>

# Expressed sequence tags (ESTs) or assembled mRNA

est=<transcript_evidence.fa>

# Protein sequences from other organisms (e.g., UniProt)

protein=<protein.fa>

# Gene prediction method

# (1st round training derive gene mode from EST, i.e., est2genome=1 or protein evidence, i.e.,
protein2genome=1)

est2genome=1 | protein2genome=1

# (c) Run MAKER

# Run on a single processor by ''maker'' or on ''N'' processors by ''mpirun -n''

  maker | mpirun -n N maker

# (d) Collect annotation result and merge into a single file

  cd <maker.ouput>

  gff3_merge -d <genome_datastore_index.log> -g

# (e) Make a directory for the training

  mkdir <snapTrain1>
```

```
  cd <snapTrain1>
# (f) Generate files required for training
# Generate <genome.ann>, <genome.dna> required to train SNAP
maker2zff <../all.gff>
# ''fathom'' separates annotation into categories
# uni: single gene per sequence
# alt: genes with alternative splicing
# olp: genes overlap others
# err: genes with errors
# wrn: genes with warnings
  fathom -categorize 1000 <genome.ann> <genome.dna>
# ''fathom'' exports the genes
# Generate export.aa, export.ann, export.dna, export.txt
  fathom export 1000 uni.ann uni.dna
# (g) Generate new parameters
  mkdir params
  cd params
  forge ../export.ann ../export.dna
  cd ..
# (h) Generate new HMM
  hmm-assembler.pl <genome> params > <genome.hmm>
  cd ..
# (i) Update maker_opts.ctl & retrain the model from step (c) to (h)
snaphmm=<genome.hmm>
est2genome=0
protein2genome=0
```

3. Gene annotation and functional annotation.
   a. Gene structure annotation.

   In addition to gene prediction models, evidence from orthologous protein sequences and transcriptome assembly could be used to improve annotation quality. Protein sequences of orthologous genes can be obtained from UniProt (The UniProt, 2017). Ones from Swiss-Port have been reviewed and thus are of higher quality. Transcriptome assembly may be available from previous studies or can be assembled de novo from RNA-seq reads by Trinity (Haas et al., 2013). High quality transcriptome assembly can be selected as described in (Zhang et al., 2021).

```
# Gene structure annotation
# (a) Generate MAKER control files
# Generate three files with suffix ''.ctl'', through which to provide user input
  maker -CTL
```

```
# (b) Edit maker_opts.ctl file to provide input parameters

genome=<genome_assembly.fa>

# choose either eukaryotic or prokaryotic

organism_type=<eukaryotic|prokaryotic>

# Expressed sequence tags (ESTs) or assembled mRNA

est=<transcript_evidence.fa>

# Protein sequences from other organisms (e.g., UniProt)

protein=<protein.fa>

# Gene prediction models

snaphmm=<SNAP_trained_model>

augustus_species=<augustus_trained_model>

# (c) Run MAKER

# Run on a single processor by ''maker'' or on ''N'' processors by '''mpirun -n''

  maker | mpirun -n N maker

# (d) Collect annotation result and merge into a single file

  cd <maker.ouput>

  gff3_merge -d <genome_datastore_index.log> -g
```

Note: Details about gene structure annotation (Holt and Yandell, 2011) can be found at http://gmod.org/wiki/MAKER_Tutorial, https://darencard.net/blog/2017-05-16-maker-genome-annotation/, and the protocol (Campbell et al., 2014).

b. Quality measurement and functional annotation.

For each predicted gene, Maker2 provides the annotation edit distance (AED) score, which measures the goodness of fit between its predicted gene structure and its evidence support. The lower the score, the more accurate the prediction. If more than 90% genes with AED scores lower than 0.5, the genome can be considered well annotated. In addition to the AED score, a high proportion of recognizable domains contained in predicted protein – e.g., higher than 50% – also indicates a good annotation. Recognizable protein domains can by scanned by InterProScan (Jones et al., 2014), assigning potential function to predicted genes.

Note: Besides the aforementioned quality measurement, we strongly recommend measuring the completeness of the genome assembly and annotation by checking the existence of a set of Benchmarking Universal Single-Copy Orthologs (BUSCO) (Simao et al., 2015). A high-level completeness of genome assembly and annotation is imperative for a better identification of gene expansion. Based on the result of this analysis, researchers can decide whether they need to further improve the genome assembly before predicting gene expansion. A detailed protocol of BUSCO is available at (Manni et al., 2021).

```
# Run BUSCO

# Input: genome sequence or protein sequence to be measured

# Output: completeness of input regarding to near-universal single-copy orthologs

# -i: input file, either a nucleotide fasta file or a protein fasta file
```

```
# -l: lineage dataset

# -o: folder to save results

# -m: assessment mode (i.e., genome, protein, transcriptome)

  busco -i <DNA.fa|protein.fa> -l <lineage> -o <output> -m <mode>
```

Gene family construction

Timing: hours to days (e.g., ~12 h for 26k predicted proteins)

To identify gene expansion, we first need to assign genes into gene families and then count gene copies of each gene family among species in the study. Next, the table of gene counts will be used to identify gene expansion in a particular species. See Figure 1.

4. Map annotated genes into gene families.

OrthoDB (Kriventseva et al., 2019) can be used to map annotated genes into gene families. OrthoDB is a database of hierarchical catalog of orthologs, including 1,271 eukaryotes in the current release (v10.1). In addition to species included in the database, users can upload and analyze their own protein sequences and get corresponding ortholog information, which include "ortholog group name", "gene name", "score of match", etc.

Note: For a customized analysis of OrthoDB, up to 5 species can be selected as references to derive ortholog information for user's data. Well annotated and evolutionally close species are

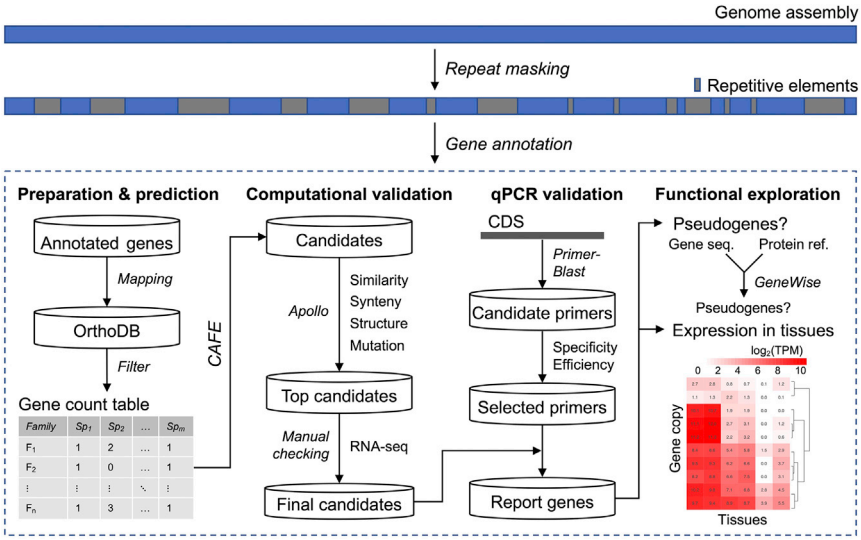


Figure 1. Workflow for gene expansion identification and functional analysis

After gene annotation, a gene count table is prepared through OrthoDB (Kriventseva et al., 2019), and then replicated genes are predicted by CAFE (Mendes et al., 2020). For computational validation, Apollo is used to further improve gene annotation quality if necessary. Protein sequence similarity, synteny, and evolution in gene structure and coding sequences can be used to identify and examine more reliable candidate genes. RNA-seq reads can differentiate gene copies generated by divergent evolution after expansion from false ones resulted from sequencing and/or assembly errors. Final candidates can be further validated by qPCR with primers designed by Primer-Blast (Ye et al., 2012). Their genome-wide specificity is checked by MFEprimer (Qu et al., 2012) and then further inspected by gel electrophoresis and the melting curve analysis. Amplification efficiency should also be checked. For the functional assessment, GeneWise (Birney et al., 2004) can be used to check potential pseudogenization of gene copies, while gene expression pattern of gene copies across different tissues can be measured by Kallisto (Bray et al., 2016) with RNA-seq data.

recommended as potential references. Details about customized analysis can be found at https://www.orthodb.org/orthodb_userguide.html#uploading-and-analyzing-your-own-sequences.

Gene family expansion identification

Timing: several hours (e.g., ~3 h for 19k sets of gene homologs from 18 species on a 10-node computer cluster with 12-GB memory in each node)

This step will report expanded gene families and corresponding species.

5. Prepare the gene count table.

With gene families on its rows and species on its columns, this table records the count of each orthologous gene in each species. We recommend removing or further refining gene families with more than 100 genes if present in some species (see the link to a tutorial in Note below), because gene families with a large variance in the gene count can cause erroneous parameter estimation.

6. Predict expanded gene families.

CAFE first calculates an error model for the prediction to account for potential errors in genome assembly and/or annotation and then carries out corrections based on the error model before calculating the gene family size of ancestor nodes. It thus can estimate gene family evolutionary rate more accurately. CAFE 5 (Mendes et al., 2020), the latest version, can account for evolution rate variation among families with improved performance on parallelization.

Note: A tutorial for the CAFE tool can be found at http://evomicsorg.wpengine.netdna-cdn.com/wp-content/uploads/2016/06/cafe_tutorial-1.pdf and https://github.com/hahnlab/CAFE5. It is a dilemma to further filter less reliable gene copies. Stringent thresholds will help to obtain most reliable predictions, while it may lead to lose some true positives. A reasonable threshold may depend on a particular project. After identifying most reliable gene families showing potential expansion, we recommend to further check all filtered predictions manually, as shown below, in those targeted gene families.

7. Additional filtering.
   a. Gene copies with low protein sequence similarity to other genes within a gene family could be removed (Keane et al., 2015; Zhang et al., 2021).
   b. The validity of gene prediction can be verified by RNA-seq data generated from either separate or pooled samples of many tissues. Marked by a lack of gene expression support, questionable predictions should be filtered, so more reliable ones can be prioritized for experimental validation (Zhang et al., 2021).

Note: Assembly and/or annotation errors could cause false prediction of gene expansion. It is time-consuming to manually check all predictions. To obtain reliable predictions, further filtering should be considered. Kallisto (Bray et al., 2016) was recommended to measure gene expression of genes with paralogs, as it shows better performance on paralogs.

Gene family expansion validation

Timing: several days (e.g., ~14 days for computational validation of ~200 candidate gene families and qPCR experimental validation of ~20 selected ones)

Above, we show how to filter out less reliable predictions through computational analysis. Nevertheless, false predictions could still present. More detailed checking is required to ascertain true ones. Here we described some steps to further validate the potential gene expansions (Figure 1).

8. Computational and manual checking.
   a. Improve the gene annotation.

   For some predicted gene copies, the annotation of their gene structure could be problematic (e.g., wrong prediction of splicing sites). We recommend improving the annotation quality of gene copies in targeted potentially expanded gene families. Apollo (Dunn et al., 2019) is a convenient tool to adjust and improve gene structure annotation.

   Note: BAM files of RNA-seq read alignments provide good evidence to improve gene structure annotation. In addition, alignment of coding sequences between studied species and well annotated species (e.g., human and mouse) could provide useful information and help to identify local regions with poor annotation quality (Abascal et al., 2010).

   b. Synteny analysis of gene loci.

   Synteny analysis provides information about the conservation of homologous genes and gene orders between genomes of different species. At gene loci of targeted genes, it provides clues about whether the extra gene copies could come from genome assembly errors. Any genome browser – e.g., IGV (Thorvaldsdottir et al., 2013) – can be used for this analysis.

   c. Gene structure, coding sequence analysis of predicted genes.

   After gene duplication, each copy of the gene may evolve independently and shows differences in their gene structure and coding sequences. Existing of such evolutionary changes among different gene copies could indicate true positives rather than false identification due to errors in genome assembly and annotation. Some tools are helpful for such analysis.

      i. GSDS (Hu et al., 2015) can generate plots of gene structure with the input of gene structure information in BED format or GTF format. It is accessible at http://gsds.gao-lab.org/.
      ii. TranslatorX (Abascal et al., 2010) can align multiple sequences guided by amino acid translations. Accessible at http://www.translatorx.co.uk/, it can be used to view evolutionary changes in coding sequences among different copies of the expanded gene.

   d. RNA-seq supports of predicted genes.

   It is possible that the difference in coding sequences may come from sequencing errors rather than evolutionary changes. The aligned RNA-seq reads are useful to distinguish between them. Sequence differences are most likely real evolutionary changes if they are present in both genomic sequence and RNA-seq reads. Otherwise, they are probably sequencing errors in the genome assembly as they are not verified by corresponding transcript sequences.

   Note: IGV (Thorvaldsdottir et al., 2013) can be used to view RNA-seq reads coverage at unique sites among copies by using BAM file of RNA-seq read alignment. By default, STAR will only report alignments of reads with at most 10 mapped loci, which can be adjusted by the parameter "–outFilterMultimapNmax", in case predicted gene copy number of target gene family is greater than 10.

9. Gene copy number validation by qPCR.
   a. Selection of reference genes.

To quantify the gene copy number by qPCR, it is important to select reference genes with normal copy numbers. It is reasonable to select genes with no predicted expansion in the studied genome and no known gene expansion in other closely related species. Orthologous databases are useful to identify such genes. For example, for each gene family, OrthoDB (Kriventseva et al., 2019) shows number of species with certain number of copies. Genes with consistent copy numbers across all the species are good candidates as reference genes. It is also recommended to select reference genes from both autosomes and the X chromosome, with 2 or 3 candidates each. If the genomic DNA of a male individual was used for the experiments, candidates from autosome and the X chromosome can be validated for each other due to expected double copy number on autosome compared with that on X chromosome.

b. Primer design.

Design primers for genes under expansion could be challenging, because different gene copies evolve differently. Highly conserved coding sequences are good candidates for designing primers. Below shows the procedure to design and select primers.

i. We use Primer-Blast (Ye et al., 2012) to design primer candidates in coding sequences of target genes, and set conserved regions among different copies as the searching regions.
ii. Specificity of primers – i.e., no other targets except for the copies of the expanded gene – can be first checked by Primer-Blast against all coding sequences of the genome, and then by MFEprimer (Qu et al., 2012) on the genomic DNA level.
iii. Final candidates can be further inspected by gel electrophoresis and the melting curve analysis.

c. qPCR validation.

The first step is to check whether the amplification efficiency for each candidate primer is the same, which can be achieved by a standard curve analysis with different amount of DNA input for the replicates. The relative gene copy number can then be calculated by ΔC_t (Zhang et al., 2021).

Note: When genomic DNA of a male individual is used, reference genes from both autosome and X chromosome should be used for qPCR validation. For each gene, 2 or 3 candidate primers should be designed, and among them the more accurate ones should be selected.

Functional explore of expanded genes

Timing: several hours (e.g., ~3 h for 25 RNA-seq samples on a 10-node computer cluster with 12-GB memory in each node)

It is important to check whether different copies of an expanded gene function in similar way as the ancient copy. Thus, it is important to check pseudogenization and transcriptional level of each copy of the expanded gene. See Figure 1.

10. Pseudogene identification.
    a. Check whether a copy of an expanded gene could be a processed intronless pseudogene through mRNA retrotransposition. Some genes, however, do have only a single exon. To identify processed pseudogenes, one can search whether the gene is known to have a single exon in evolutionarily closely species in a gene ortholog database, such as OrthoDB (Kriventseva et al., 2019).
    b. Then check whether a gene copy was pseudogenized through mutations, such as frameshift indels. GeneWise (Birney et al., 2004) can identify such mutations, when it is used to scan the

gene locus – e.g., gene body with upstream and downstream 5-kb – against the coding sequence of the functional orthologous gene (e.g., a human ortholog) as the reference. GeneWise is used in our pipeline. Alternatively, researchers can use any other tools, e.g., PseudoPipe (Zhang et al., 2006). The genome sequence at the gene locus can be retrieved by the bedtools (Quinlan and Hall, 2010).

Note: The pipeline to identify pseudogene is straightforward and relatively simple here. However, it is sufficient for analyzing a small number of genes. More complicated methods for genome-wide identification of pseudogenes can be found in other studies (Sisu et al., 2020).

```
# Run bedtools

# Input: genome sequence, gene coordinate (±5 kb) in bed format

# Output: genome sequence in gene coordinate (±5 kb)

# fi: the genomic sequence

# -bed: gene coordinate in bed format (extend to upstream/downstream 5 kb)

# -s: force strandedness. Return reverse complement if the gene is on the antisense strand.

# -name: use ''name'' column in bed file as fasta headers of output

  bedtools getfasta -fi <genomic.fa> -bed <geneCoordinate.bed> -s -name > genomic-region.fa
```

```
# Run GeneWise

# Input: protein sequence of homolog, DNA sequence of genomic region of a predicted gene copy

# Output: report mutations that pseudogenized the gene copy

# -sum: show summary output

# -pretty: show pretty ascii output

# -pseudo: mark genes with frameshifts as pseudo genes

# -genes: show gene structure

# -cdna: show predicted cDNA sequence

# -trans: show protein translation

# -pep: show predicted peptide

# -para: show parameters

# -both: check both strand

# -quiet: no report on stderr

  genewise <protein.fa> <genomic-region.fa> -sum -pretty -pseudo -genes -cdna -trans -pep
-para -both -quiet > out.gw
```

11. Gene expression.

The transcription level of each copy of an expanded gene can be checked to explore its functionality. RNA-seq data from different tissues can be used to examine gene expression pattern across tissues and identify tissue-specific expression of some copies. Kallisto (Bray et al., 2016) shows better performances on paralogs.

EXPECTED OUTCOMES

This protocol can be used to identify gene copy number expansion and examine whether gene copies are likely functional and how they are transcribed across different tissues (see Figure 1 for the pipeline).

LIMITATIONS

The identification of expanded genes highly depends on the quality of genome assembly and annotation, which is often limited for de novo assembly of a non-model organism. So manually checking (step 8) and deeper sequencing are recommended for more accurate identification of such genes. Without genome sequences and annotation of evolutionally close species, both the prediction and validation of gene expansion will be challenging. However, for slowly evolving genomes, annotation from certain model organisms could still be useful. In addition to orthologous genes from other species, RNA-seq data are also helpful for manual checking. Please note that our protocol is intended for studies of mammalian genomes, which are evolutionally stable with low evolutionary rate. While it may be applied to the genomes of lower species, its performance needs to be examined and validated.

TROUBLESHOOTING

Problem 1

At step 1b, there are issues with setting up BUSCO.

Potential solution

BUSCO and its dependencies can also be installed by using Docker container or Conda. More directions can be found in (Manni et al., 2021), and specific issues can be raised at https://gitlab.com/ezlab/busco/-/issues.

Problem 2

At step 2a, training Augustus takes too much time.

Potential solution

The Perl module "Parallel::ForkManger" is needed for the function "optimize_augustus.pl" to run in parallel.

```
# Install the package to train Augustus in parallel

sudo apt-get install libparallel-forkmanager-perl
```

Problem 3

At step 4, while OrthoDB (Kriventseva et al., 2019) can be used to construct gene families for de novo genome assembly and annotation, it is not clear how to select reference genomes and what is the best practice if only a small number of reference genomes can be selected.

Potential solution

To limit computational complexity, users can select up to 5 reference genomes to construct gene families for de novo gene annotation. We recommend selecting well-annotated genomes of species evolutionally close to the target species. To achieve this, they should first check the publications of those genomes for their quality. Next, they should check UniProt (The UniProt, 2017) to find out how many proteins, reviewed or not, are available for the target species. Finally, they can measure the completeness of genome assembly and annotation using BUSCO (Simao et al., 2015).

To reduce potential bias due to limited reference genomes, users should repeat the whole procedure 3 times, randomly replacing 2–3 reference genomes in each iteration, and only keep consistent results mapped to the same gene families.

Problem 4

At step 6, genes with high evolutionary rates have higher probability gone through gene expansion/contraction in a studied species. Due to the high evolutionary rates, such expansion/contraction changes are usually not species-specific. It could be challenging to identify potential connection between such evolutionary changes and species-specific traits.

Potential solution

First, we can check whether any genes have notably gone through much more significant expansion in the study-focused species than the others. Second, we can check whether expansion of certain genes in multiple species could potentially connect to certain traits shared among them. Third, gene expansion that is specific to the study-focused species could also be explored with a similar pipeline described in this protocol (more details can be found in (Zhang et al., 2021)).

Problem 5

At step 11, some genes may be expressed only in certain tissues. If such tissues are not included in the study, RNA-Seq reads from those tissue-specific genes may be unavailable for validation.

Potential solution

RNA-seq data from multiple tissues or pooled tissues could help to cover most tissue-specific genes. Although RNA-seq data are optional, they are helpful for both validation and functional annotation, such as expression patterns of all gene copies across tissues/conditions.

RESOURCE AVAILABILITY

Lead contact

Further information or requests should be directed to and will be fulfilled by the lead contact, Zhengdong D. Zhang (zhengdong.zhang@einsteinmed.edu).

Materials availability

This study did not generate new unique reagents.

Data and code availability

Our code and scripts used in this protocol are available in Methods S2 and also on GitHub at https://github.com/zdz-lab/gene-expansion (DOI at Zenodo: https://doi.org/10.5281/zenodo.6784092).

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.xpro.2022.101692.

ACKNOWLEDGMENTS

This work was supported by NIH grant P01AG047200 and a grant from the Irma T. Hirschl Trust to Z.D.Z.

AUTHOR CONTRIBUTIONS

Q.Z. prepared the material and manuscript. Z.D.Z. supervised this study and revised the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

Abascal, F., Zardoya, R., and Telford, M.J. (2010). TranslatorX: multiple alignment of nucleotide sequences guided by amino acid translations. Nucleic Acids Res. *38*, W7–W13.

Aken, B.L., Ayling, S., Barrell, D., Clarke, L., Curwen, V., Fairley, S., Fernandez Banet, J., Billis, K., Garcia Giron, C., Hourlier, T., et al. (2016). The Ensembl gene annotation system. Database.

Bao, W., Kojima, K.K., and Kohany, O. (2015). Repbase update, a database of repetitive elements in eukaryotic genomes. Mob. DNA *6*, 11.

Birney, E., Clamp, M., and Durbin, R. (2004). GeneWise and genomewise. Genome Res. *14*, 988–995.

Bray, N.L., Pimentel, H., Melsted, P., and Pachter, L. (2016). Near-optimal probabilistic RNA-seq quantification. Nat. Biotechnol. *34*, 525–527.

Bruna, T., Hoff, K.J., Lomsadze, A., Stanke, M., and Borodovsky, M. (2021). BRAKER2: automatic eukaryotic genome annotation with GeneMark-EP+ and AUGUSTUS supported by a protein database. NAR Genom. Bioinform. *3*, lqaa108.

Campbell, M.S., Holt, C., Moore, B., and Yandell, M. (2014). Genome annotation and curation using MAKER and MAKER-P. Curr. Protoc. Bioinformatics *48*, 4.11.1–4.11.39.

Dunn, N.A., Unni, D.R., Diesh, C., Munoz-Torres, M., Harris, N.L., Yao, E., Rasche, H., Holmes, I.H., Elsik, C.G., and Lewis, S.E. (2019). Apollo: democratizing genome annotation. PLoS Comput. Biol. *15*, e1006790.

Haas, B.J., Papanicolaou, A., Yassour, M., Grabherr, M., Blood, P.D., Bowden, J., Couger, M.B., Eccles, D., Li, B., Lieber, M., et al. (2013). De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. Nat. Protoc. *8*, 1494–1512.

Holt, C., and Yandell, M. (2011). MAKER2: an annotation pipeline and genome-database management tool for second-generation genome projects. BMC Bioinf. *12*, 491.

Hu, B., Jin, J., Guo, A.Y., Zhang, H., Luo, J., and Gao, G. (2015). Gsds 2.0: an upgraded gene feature visualization server. Bioinformatics *31*, 1296–1297.

Jones, P., Binns, D., Chang, H.Y., Fraser, M., Li, W., McAnulla, C., McWilliam, H., Maslen, J., Mitchell, A., Nuka, G., et al. (2014). InterProScan 5: genome-scale protein function classification. Bioinformatics *30*, 1236–1240.

Keane, M., Semeiks, J., Webb, A.E., Li, Y.I., Quesada, V., Craig, T., Madsen, L.B., van Dam, S., Brawand, D., Marques, P.I., et al. (2015). Insights into the evolution of longevity from the bowhead whale genome. Cell Rep. *10*, 112–122.

Kriventseva, E.V., Kuznetsov, D., Tegenfeldt, F., Manni, M., Dias, R., Simão, F.A., and Zdobnov, E.M. (2019). OrthoDB v10: sampling the diversity of animal, plant, fungal, protist, bacterial and viral genomes for evolutionary and functional annotations of orthologs. Nucleic Acids Res. *47*, D807–D811.

Manni, M., Berkeley, M.R., Seppey, M., and Zdobnov, E.M. (2021). BUSCO: assessing genomic data quality and beyond. Curr. Protoc. *1*, e323.

Mendes, F.K., Vanderpool, D., Fulton, B., and Hahn, M.W. (2020). CAFE 5 models variation in evolutionary rates among gene families. Bioinformatics *36*, 5516–5518.

Qu, W., Zhou, Y., Zhang, Y., Lu, Y., Wang, X., Zhao, D., Yang, Y., and Zhang, C. (2012). MFEprimer-2.0: a fast thermodynamics-based program for checking PCR primer specificity. Nucleic Acids Res. *40*, W205–W208.

Quinlan, A.R., and Hall, I.M. (2010). BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics *26*, 841–842.

Simão, F.A., Waterhouse, R.M., Ioannidis, P., Kriventseva, E.V., and Zdobnov, E.M. (2015). BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs. Bioinformatics *31*, 3210–3212.

Sisu, C., Muir, P., Frankish, A., Fiddes, I., Diekhans, M., Thybert, D., Odom, D.T., Flicek, P., Keane, T.M., Hubbard, T., et al. (2020). Transcriptional activity and strain-specific history of mouse pseudogenes. Nat. Commun. *11*, 3695.

Tarailo-Graovac, M., and Chen, N. (2009). Using RepeatMasker to identify repetitive elements in genomic sequences. Curr. Protoc. Bioinformatics *Chapter 4*, Unit 4.10.

The UniProt, C. (2017). UniProt: the universal protein knowledgebase. Nucleic Acids Res. *45*, D158–D169.

Thorvaldsdottir, H., Robinson, J.T., and Mesirov, J.P. (2013). Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. Brief. Bioinform. *14*, 178–192.

Ye, J., Coulouris, G., Zaretskaya, I., Cutcutache, I., Rozen, S., and Madden, T.L. (2012). Primer-BLAST: a tool to design target-specific primers for polymerase chain reaction. BMC Bioinf. *13*, 134.

Zhang, Q., Tombline, G., Ablaeva, J., Zhang, L., Zhou, X., Smith, Z., Zhao, Y., Xiaoli, A.M., Wang, Z., Lin, J.R., et al. (2021). Genomic expansion of Aldh1a1 protects beavers against high metabolic aldehydes from lipid oxidation. Cell Rep. *37*, 109965.

Zhang, Z., Carriero, N., Zheng, D., Karro, J., Harrison, P.M., and Gerstein, M. (2006). PseudoPipe: an automated pseudogene identification pipeline. Bioinformatics *22*, 1437–1439.

Zhou, X., Dou, Q., Fan, G., Zhang, Q., Sanderford, M., Kaya, A., Johnson, J., Karlsson, E.K., Tian, X., Mikhalchenko, A., et al. (2020). Beaver and naked mole rat genomes reveal common paths to longevity. Cell Rep. *32*, 107949.