**Tree Construction**

The orthologues of the F-actin binding motif-containing protein were retrieved from the OrthoDB database in FASTA format. This step ensured a comprehensive and accurate set of homologous sequences for subsequent analyses. The sequences were aligned using the AlignSeqs function from the DECIPHER package, configured with 100 iterations and 200 refinements while utilizing all available processors **(Code 1)**. The aligned sequences were then used to construct a phylogenetic tree with IQ-TREE 2, employing the WAG model for amino acid substitution. The robustness of the tree was assessed with 1000 bootstrap replicates **(Code 2)**.

**Isoform Extraction and Filtering**

Isoform sequences were extracted by filtering the OrthoDB sequences based on specific patterns associated with known isoform identifiers **(Code 3)**. This process was followed by the exclusion of non isoforms **(Code 4)**, to also gather a set of canonical proteins, resulting in a refined set of sequences that could be analyzed further. The headers of these sequences were then modified to ensure compatibility with downstream analyses **(Code 5)**.

**Isoform Clustering Analysis and Tree Statistics**

Phylogenetic analysis was performed to assess the clustering of isoforms within the phylogenetic tree, with distinct colors used to highlight isoforms within the tree visualization **(Code 6)**. This analysis helped determine the degree of divergence among isoforms and revealed varying patterns of clustering. Statistical values such as mean, median, and standard deviation of branch lengths and bootstrap values were calculated to further analyze the tree **(Code 7)**. Where necessary, branches exceeding a specified threshold were pruned to ensure accurate phylogenetic interpretations **(Code 8)**.

**Taxonomic Annotation and Isoform Divergence**

Following pruning, the phylogenetic tree was annotated with taxonomic information to determine the most recent common ancestor (MRCA) for each isoform cluster and canonical sequences **(Code 9)**. Export the clade here (See next paragraph)! To complement this, a taxon-based phylogenetic tree was generated and plotted with isoforms, providing an additional perspective on isoform distribution across taxa **(Code 10)**. The annotated trees were color-coded based on isoforms to highlight significant divergence events and to assist in understanding the evolutionary trajectory of these sequences **(Code 11)**.

**Isoform-Specific Clade Export and Annotation**

For deeper analysis, isoform-specific sequences were exported into separate tree files. These tree files were then converted into corresponding FASTA files, allowing for consistent cross-dataset comparisons **(Code 12)**. This approach was particularly useful in identifying and annotating sequences that, while not properly annotated in public databases, could still be confidently assigned to an isoform based on their phylogenetic clustering.

**-----------------------------------------------------------------------------------------------**

**Motif Extraction and PSSM Analysis**

The OrthodB sequences were then analyzed to extract the F-actin binding motif using a regular expression pattern, coupled with an IUPRED Disorder and Anchor, as well as an single PSIPRED random Coil score filter to reduce false positives **(Code 13)**. The motifs obtained were used to generate a Position-Specific Scoring Matrix (PSSM) based on the PAM30 substitution matrix **(Code 14)**. The sequences were re-filtered using this PSSM, with those meeting the specified thresholds being retained for further analysis. Cutoffs were precalculated in preliminary results. A second round of PSSM generation and motif filtering was conducted to refine the results further **(Codes 15 and 16)**.

**Determination of the Most Recent Common Ancestor (MRCA) in Motif-Containing Sequences**

To elucidate the taxonomic distribution of the F-actin binding motif, we identified the most recent common ancestor (MRCA) of motif-containing sequences within the phylogenetic tree. Isoform-specific tree files were utilized as additional input to accurately determine the MRCA for each isoform, leveraging the advantage of sequence-based clustering over reliance on annotated isoforms. This approach allowed for more precise tracking of evolutionary relationships, including sequences that were not annotated but nonetheless part of the isoform clade **(Code 17)**. The script also provided insights into motifs that could not be mapped to the tree, along with their taxonomic information.

**Validation of Motif Distribution in a Taxon-Based Tree**

To ensure the accuracy of the taxonomic characterization, we validated the motif distribution by plotting the motifs within a taxon-based phylogenetic tree. This involved overlaying the motif-containing sequences onto the taxon-based tree to verify the correctness of the MRCA identification and the overall motif distribution. Isoform-specific FASTA files were included to refine the plot further. The resulting tree plot visually confirmed that the motif distribution aligned with expected taxonomic relationships, reinforcing the robustness of our previous analyses **(Code 18)**.

**Motif Development Across Isoforms and Taxonomic Classes**

To investigate the development of the motif across different isoforms and taxonomic classes, we analyzed the motif distribution within the phylogenetic tree. Isoform-specific analyses were conducted using previously isolated clades, generating frequency matrices for amino acid positions within motifs and comparing these matrices within and across isoforms. This allowed for a focused examination of the motif's evolution within each isoform and class, ensuring that evolutionary patterns were accurately represented **(Code 19)**.

**Sequence Logo Generation**

To visualize the conserved motifs within the different taxonomic classes, sequence logos were generated from the Position-Specific Scoring Matrices (PSSMs). The logos were saved as PNG files for high-quality visualization **(Code 20)**.

**Construction of an Exemplary Phylogenetic Tree**

To illustrate the taxonomic relationships among the sequences, an exemplary phylogenetic tree was constructed using data retrieved from the NCBI taxonomy database, focusing on specific taxonomic groups of interest. The tree was built and saved in Newick format, including internal node names to clarify the taxonomic hierarchy. This step provided a clear representation of the evolutionary relationships within the selected taxonomic groups, serving as a reference for the motif and isoform analyses **(Code 21)**.

**-----------------------------------------------------------------------------------------------**

**Conversion of Clade-Specific Isoform Trees to Sequence Data for Subsequent Analysis**

To refine the isoform analysis, we extracted sequence information directly from clade-specific phylogenetic trees. The extracted clade in .treefile format was converted into FASTA format, allowing the inclusion of sequences that, while not annotated as isoforms in public databases, were confidently identified based on their phylogenetic clustering. We used this sequence data to compare against annotated sequences from the NCBI database. By substituting general PSSM data with more specific isoform PSSM sequences, we assessed whether additional sequences similar to specific isoforms were identified, thereby expanding our understanding of isoform diversity **(Code 22)**.

**Retrieval of Accession Numbers and DNA Sequences from NCBI**

Further refinement involved retrieving accession numbers and corresponding DNA sequences from the NCBI database for specific motifs. The script checked whether the motif was present in the translated DNA sequence and extracted the nucleotide sequence corresponding to the motif. This step was essential for constructing nucleotide-based phylogenetic trees and other nucleotide-level analyses, such as dN/dS ratio calculations **(Code 23C)**. This code is extracting the Full length CDS, as well as the Motif with flanking codons.

**Filtering Ambiguous DNA Sequences and Removing Stop Codons**

Finally, DNA sequences were filtered by removing those with ambiguous nucleotides, trimming sequence IDs, and removing stop codons, ensuring that sequences were of a length divisible by three **(Code 24)**. Also it was assured that in the end the same and identical headers were present in the full length and trimmed sequences **(Code 25)**. This filtering step was crucial to prepare clean sequences for downstream analysis.

**Codon Alignment with Prank and subsequent IQTree2**

A codon sequence alignment was performed with Prank using the following command:   
/home/pythagoras/Programs/PRANK/prank.linux64.170427/prank/bin/prank -d=ARHGEF11\_FL\_DNA\_Sequences\_Common.fasta -codon -F

Then IQTree2 was run on the full length aligned sequence:   
iqtree2 -s /home/pythagoras/Documents/PhD/Evolution/FL/ITPKA/ITPKA/IQTree\_output/output.flank.best.fasta -m GTR -bb 1000 -nt 11

Jalview was used to visualize the trimmed motif sequence and T-Coffee with a gap opening penalty of -20 and gap extension penalty of -5.

**dN/dS Analysis using Hyphy**

Hyphy fel --alignment /home/pythagoras/Documents/PhD/Evolution/FL/ITPKA/ITPKA/ITPKA\_best.fasta --tree /home/pythagoras/Documents/PhD/Evolution/FL/ITPKA/ITPKA/output.best.fasta.treefile

**Ancestor Sequence reconstruction using Hyphy**

hyphy /home/pythagoras/Downloads/FitMG94.bf --alignment /home/pythagoras/Documents/PhD/Evolution/FL/Shroom3/dNdS/Flankoutput1.best.fasta --tree /home/pythagoras/Documents/PhD/Evolution/FL/Shroom3/dNdS/FLoutput.best.fasta.treefile --save-fit /home/pythagoras/Documents/PhD/Evolution/FL/Shroom3/dNdS/Shroom3\_best1.fit

hyphy /home/pythagoras/Downloads/AncestralSequences.bf --fit /home/pythagoras/Documents/PhD/Evolution/FL/Shroom3/dNdS/Shroom3\_best1.fit --output /home/pythagoras/Documents/PhD/Evolution/FL/Shroom3/dNdS/Shroom3\_best1.json

**Code 1: Multiple Sequence Alignment with DECIPHER**

* **Function:** This code aligns a set of amino acid sequences using the AlignSeqs function from the DECIPHER package, applying 100 iterations and 200 refinements with parallel processing.
* **Inputs:** FASTA file in aa format: ITPK\_OrthodDB.fasta
* **Outputs:** Aligned Sequences FASTA File: ITPK\_aligned\_protein\_sequences.fasta

**Code 2: Phylogenetic Tree Construction with IQ-TREE 2**

* **Function:** This command constructs a maximum likelihood phylogenetic tree using the IQ-TREE 2 software with the WAG substitution model, performing 1000 bootstrap replicates to assess the tree's robustness.
* **Inputs:** alignedFASTA File: ITPK\_aligned\_protein\_sequences.fasta
* **Outputs:** 
  + .treefile: The final phylogenetic tree in Newick format.
  + .log: Log file of the IQ-TREE run.
  + .iqtree: Summary file containing details about the model selection, log-likelihood, and bootstrap analysis.

**Code 03: Isoform Filtering from OrthoDB Sequences**

* **Function:** This script filters sequences from an OrthoDB FASTA file by searching for specific keywords related to the isoform in the sequence headers.
* **Inputs:**
  + FASTA File: ITPK\_OrthoDB.fasta containing all OrthoDB sequences.
  + Hardcoded Parameters:
    - Search Pattern: "(itpka|ITPKA|inositol-trisphosphate 3-kinase A)" (regular expression pattern to filter sequences by keyword in headers).
* **Outputs:** Filtered Isoform FASTA File: ITPKA\_OrthoDB.fasta containing sequences that match the specified keywords.

**Code 04: Exclusion of Specific Isoforms from OrthoDB Sequences**

* **Function:** This script filters sequences from an OrthoDB FASTA file, excluding those that match specific keywords related to the ITPK isoforms (A, B, C) in the sequence headers.
* **Inputs:**
  + FASTA File: ITPK\_OrthoDB.fasta containing all OrthoDB sequences.
  + Hardcoded Parameters:
    - Exclusion Pattern: r"\b(inositol-trisphosphate 3-kinase [ABC]|description":"Inositol-trisphosphate 3-kinase [ABC]|itpk[abc]|ITPK[ABC])\b" (regular expression pattern to exclude sequences by keyword in headers).
* **Outputs:** Filtered Non-Isoform FASTA File: Only\_ITPK\_OrthoDB.fasta containing sequences that do not match the specified keywords.

**Code 05: Header Modification for Isoform Sequence Files**

* **Function:** This script modifies the headers of isoform sequences by replacing spaces and special characters with underscores across the entire description, ensuring compatibility with downstream tools.
* **Input:** FASTA File: ITPKA\_OrthoDB.fasta containing isoform sequences that need header modification.
* **Outputs:** Modified Isoform Sequence FASTA File: ITPKA.fasta containing sequences with fully modified headers.

**Code 06: Phylogenetic Tree Plotting with Color-Coded Isoforms**

* **Function:** This script loads a phylogenetic tree, annotates it with isoform information from corresponding FASTA files, and plots the tree with isoforms color-coded. The tree is also saved with modified leaf names to reflect the isoform annotations.
* **Inputs:**
  + Newick Tree File: ITPK\_Isoform\_aligned\_protein\_sequences.fasta.treefile containing the phylogenetic tree.
  + FASTA Files:
    - ITPKA.fasta for ITPKA isoform.
    - ITPKB.fasta for ITPKB isoform.
    - ITPKC.fasta for ITPKC isoform.
  + Hardcoded Parameters:
    - ITPKA: red
    - ITPKB: green
    - ITPKC: blue
* **Outputs:**
  + Annotated Phylogenetic Tree: A visual tree with isoforms color-coded and annotated based on their sequence identifiers.
  + Modified Tree File: **ITPK\_SequenceAligned\_tree\_plot\_with\_isoforms.treefile** containing the tree with updated leaf names.

**Code 07: Tree Statistics for Cleaning**

* **Function:** This script calculates branch lengths and bootstrap values from a phylogenetic tree, suggests a cutoff for branch lengths based on the standard deviation, and visualizes these statistics.
* **Inputs:**
  + Newick Tree File: ITPK\_SequenceAligned\_tree\_plot\_with\_isoforms.treefile containing the phylogenetic tree.
  + Hardcoded Parameters:
    - Standard Deviation Factor: 7 (used to suggest a cutoff for branch lengths based on mean + 7 \* std\_dev).
  + **Outputs:**
    - Statistics and Suggested Cutoff: The script prints and visualizes:
      * Mean, median, standard deviation, max, and min of branch lengths.
      * A histogram of branch lengths with the suggested cutoff line.
      * Mean, median, standard deviation, max, and min of bootstrap values (without suggesting a cutoff).

**Code 08: Pruning Phylogenetic Tree Based on Branch Length**

* **Function:** This script prunes branches from a phylogenetic tree if their branch length exceeds a specified threshold. It also cleans up the tree by removing empty or improperly structured nodes and allows for visualizing the pruned tree.
* **Inputs:**
  + Newick Tree File: ITPK\_SequenceAligned\_tree\_plot\_with\_isoforms.treefile containing the phylogenetic tree to be pruned.
  + Threshold Distance: XX (branch lengths greater than this value are pruned).
* **Outputs:** Pruned Phylogenetic Tree: ITPK\_cleaned\_tree.treefile containing the tree after pruning and cleaning.

**Code 09: Describing Taxonomic Development and Isoform Divergence**

* **Function:** This script annotates a phylogenetic tree with taxonomic information, identifies isoform divergence events, and provides a detailed analysis of highly significant evolutionary splits. It also visualizes the tree with color-coded isoform distributions and taxonomy. The very important thing it relies on is the prior annotated header by isoform
* **Inputs:**
  + Pruned Newick Tree File: ITPK\_cleaned\_tree.treefile containing the phylogenetic tree after pruning.
  + Hardcoded Parameters:
    - Minimum Proportion for Significant Divergence: 0.01 (proportion of total tree to consider a split significant).
    - Minimum Isoform Count: 20 (minimum number of isoforms to consider in a split).
    - Minimum Separation Ratio: 0.7 (ratio to determine significant divergence in isoform counts).
* **Outputs:** 
  + Annotated Phylogenetic Tree: Displays isoform counts, taxonomy, and significant divergence events.
  + Summary of Isoform Divergence: Prints detailed information about highly significant isoform divergence events, including evolutionary interpretations.

**Separate Isoform Clade in ete manually: Advantageous, as it helps recognizing non or false annotated sequences.**

**Code 010: Transforming Sequence Data into a Taxonomic Tree**

* **Function:** This script transforms sequence data into a taxonomic tree by extracting taxonomic IDs from sequence headers, retrieving their taxonomic lineages using the local NCBI taxonomy database, and constructing a tree based on these lineages.
* **Inputs:**
  + FASTA File: ITPK\_Isoform\_aligned\_protein\_sequences.fasta containing sequences with taxonomic information embedded in the headers.
* Outputs: Taxonomic Tree: A tree structure saved in Newick format (ITPK\_tax\_tree.newick) with nodes representing taxonomic hierarchy and leaves representing individual sequences.

**Code 011: Plotting Taxonomic Tree with Color-Coded Isoforms**

* **Function:** This script annotates a taxonomic tree with isoform information, color-codes the isoforms in the tree plot, and saves the modified tree.
* **Inputs:**
  + Newick Tree File: ITPK\_tax\_tree.newick containing the taxonomic tree.
  + FASTA Files:
    - ITPKA.fasta for ITPKA isoform.
    - ITPKB.fasta for ITPKB isoform.
    - ITPKC.fasta for ITPKC isoform.
  + Hardcoded Parameters:
    - Isoform Color Mapping:
      * ITPKA: red
      * ITPKB: green
      * ITPKC: blue
* **Outputs:** 
  + Annotated Taxonomic Tree: A visual tree with isoforms color-coded and annotated based on their sequence identifiers.
  + Modified Tree File: ITPK\_TaxAligned\_tree\_plot\_with\_isoforms.treefile containing the tree with updated leaf names.

**Code 012: Generating Custom Annotations from Treefile**

* **Function:** This script parses identifiers from a phylogenetic treefile and writes them into a custom FASTA annotation file. It allows for optional transformation of identifiers before generating the output.
* **Inputs:** Treefile: ITPKA.treefile containing the tree with sequence identifiers.
* **Outputs:** Custom FASTA Annotation File: ITPKA\_ownAnnotation.fasta containing the extracted and optionally transformed identifiers.

**Code 013: Motif Extraction and Filtering with RegEx and IUPRED**

* **Function:** This script identifies and extracts F-actin binding motifs from isoform sequences using a regular expression pattern. It then filters the extracted motifs based on IUPRED Disorder, ANCHOR and SinglePsipred random Coil scores to retain only those with scores above a defined threshold. If there are several Motifs for each Isoform, just run them separately and put them together in the end. Remove Duplicates in the end.
* **Inputs:**
  + FASTA File: ITPK\_OrthoDB.fasta containing all isoform sequences.
  + Hardcoded Parameters:
    - IUPRED and ANCHOR cutoff score: 0.4 (filter for flanking regions)
    - IUPRED flanking size: 60 amino acids (regions around the motif).
    - Motif Regular Expression: V[^P][^P][RKLIMV][LIMVKRC][^P][^P][FY]E (the pattern used to identify the motif).
* **Outputs:** Filtered Motif FASTA File: ITPK\_RegEx.fasta containing sequences that match the motif and meet the IUPRED score criteria.

**Code 014: PSSM Generation from Filtered Motif Sequences**

* **Function:** This script generates a Position-Specific Scoring Matrix (PSSM) from a set of filtered motif sequences, using the PAM30 substitution matrix to score the amino acid positions.
* **Inputs:** 
  + FASTA File: : ITPK\_RegEx.fasta containing the filtered motif sequences.
* **Outputs:** PSSM CSV File: pssm\_matrix1\_ITPK.csv containing the calculated PSSM values.

**Code 015: Motif Filtering with Pre-calculated PSSM and IUPRED**

* **Function:** This script filters isoform sequences by scoring them against a pre-calculated PSSM and IUPRED disorder scores. Sequences that pass the defined thresholds are retained, and their motifs are extended with additional flanking amino acids.
* **Inputs:**
  + FASTA File: ITPK\_OrthoDB.fasta containing isoform sequences to be filtered.
  + PSSM Matrix CSV: pssm\_matrix1\_ITPK.csv used to score the sequences.
  + Hardcoded Parameters:
    - PSSM Cutoff Score: 17 (threshold for retaining sequences).
    - IUPRED Cutoff Score: 0.5 (threshold for disorder prediction).
    - IUPRED Flanking Size: 30 amino acids.
    - N-terminal Additional Amino Acids: 0
    - C-terminal Additional Amino Acids: 0
* **Outputs:** Filtered Motif FASTA File: ITPK\_meanPSSM\_cut17\_001.fasta

**Again Code 014: PSSM Generation from Filtered Motif Sequences**

* **Function:** This script generates a Position-Specific Scoring Matrix (PSSM) from a set of filtered motif sequences, using the BLOSUM62 substitution matrix to score the amino acid positions.
* **Inputs:** 
  + FASTA File: ITPK\_meanPSSM\_cut17\_001.fasta containing the filtered motif sequences.
* **Outputs:** PSSM CSV File: pssm\_matrix2\_ITPK.csv containing the calculated PSSM values.

**Again Code 015: Motif Filtering with Pre-calculated PSSM and IUPRED**

* **Function:** This script filters isoform sequences by scoring them against a pre-calculated PSSM and IUPRED disorder, ANCHOR and SinglePsipred Random Coil scores. Sequences that pass the defined thresholds are retained, and their motifs are extended with additional flanking amino acids.
* **Inputs:**
  + FASTA File: ITPK\_OrthoDB.fasta containing isoform sequences to be filtered.
  + PSSM Matrix CSV: pssm\_matrix2\_ITPK.csv used to score the sequences.
  + Hardcoded Parameters:
    - CutoffScores: Preliminary Data .
    - N-terminal Additional Amino Acids: 7
    - C-terminal Additional Amino Acids: 7
* **Outputs:** Filtered Motif FASTA File: ITPK\_meanPSSM\_cut17\_002.fasta

**Code 016: Header Modification for Sequence Files**

* **Function:** This script modifies the headers of sequences by replacing spaces and special characters with underscores, ensuring compatibility with downstream tools.
* **Inputs:** FASTA File: ITPK\_meanPSSM\_cut17\_002.fasta containing sequences that need header modification.
* **Outputs:** Modified Sequence FASTA File: ITPK\_Motif.fasta containing sequences with modified headers.

**Code 017: Describing the Taxonomic Appearance of Motifs**

* Function: This script analyzes the taxonomic appearance of F-actin binding motifs within a phylogenetic tree, annotates the tree with isoform and motif information, and identifies the most recent common ancestor (MRCA) of motif-containing sequences for each isoform. The script also checks for unmapped motifs and provides detailed insights into their distribution.
* Inputs:
  + Newick Tree File: ITPK\_Isoform\_aligned\_protein\_sequences.fasta.treefile containing the phylogenetic tree.
  + Motif FASTA File: ITPK\_Motif.fasta containing the sequences of motifs to be mapped onto the tree.
  + Isoform-Specific FASTA Files:
    - ITPKA\_ownAnnotation.fasta for ITPKA isoform.
    - ITPKB\_ownAnnotation.fasta for ITPKB isoform.
    - ITPKC\_ownAnnotation.fasta for ITPKC isoform.
  + Hardcoded Parameters:
* Outputs:
  + Annotated Phylogenetic Tree: A visual and saved tree with motifs and isoforms annotated, showing their taxonomic distribution.
  + Summary of Common Ancestors: Prints the common ancestor for sequences containing motifs in each isoform.
  + Analysis of Unmapped Motifs: Provides insights into motifs that could not be mapped to the tree, along with their taxonomic information.

**Code 018: Motif Analysis Across Taxonomic Classes**

* **Function:** This script analyzes the development of motifs across different taxonomic classes within a phylogenetic tree. It annotates the tree with both isoform and motif information, color-coding the motifs and isoforms in the tree plot, and saves the modified tree.
* **Inputs:**
  + Newick Tree File: ITPK\_tax\_tree.newick containing the taxonomic tree.
  + FASTA Files:
    - ITPKA\_ownAnnotation.fasta for ITPKA isoform.
    - ITPKB\_ownAnnotation.fasta for ITPKB isoform.
    - ITPKC\_ownAnnotation.fasta for ITPKC isoform.
    - ITPK\_Motif.fasta containing the sequences of motifs to be mapped onto the tree.
  + Hardcoded Parameters:
    - Isoform Color Mapping:
      * ITPKA: red
      * ITPKB: green
      * ITPKC: blue
      * Multiple Isoforms: purple
* **Outputs:**
  + Annotated Phylogenetic Tree: A visual tree with motifs and isoforms color-coded and annotated based on their sequence identifiers.
  + Modified Tree File: ITPKA\_TaxAligned\_tree\_plot\_with\_isoforms\_and\_motifs.treefile containing the tree with updated leaf names.

**Code 019: Motif Development Across Isoforms and Taxonomic Classes**

* **Function:** This script analyzes the development of motifs across different isoforms and taxonomic classes. It identifies which isoforms and classes contain specific motifs, generates frequency matrices for amino acid positions within motifs, and compares these matrices within and across isoforms.
* **Inputs:**
  + Motif FASTA File: ITPK\_Motif.fasta containing the sequences of motifs.
  + Tree Files:
    - ITPKA.treefile for ITPKA isoform.
    - ITPKB.treefile for ITPKB isoform.
    - ITPKC.treefile for ITPKC isoform.
    - ITPKRemains.treefile for the remaining isoforms.
  + Isoform-Specific FASTA Files:
    - ITPKA\_ownAnnotation.fasta for ITPKA isoform.
    - ITPKB\_ownAnnotation.fasta for ITPKB isoform.
    - ITPKC\_ownAnnotation.fasta for ITPKC isoform.
    - ITPK\_Motif.fasta for the remaining isoforms.
* **Outputs:**
  + Frequency Matrices CSV Files: CSV files containing amino acid frequency matrices for each class within each isoform.
  + Matrix Comparison Results: A text file (matrix\_comparisons.txt) detailing the similarity between matrices within and across isoforms.
  + Class Sequence Counts: A text file (class\_sequence\_counts.txt) listing the number of sequences for each class within each isoform.

**Code 020: Sequence Logo Generation with WebLogo**

* **Function:** This script generates sequence logos from PSSMs stored in CSV files. It creates sequences based on the probabilities in the PSSMs and then uses WebLogo to visualize these sequences as sequence logos.
* **Inputs:**
  + Directory Path: /home/pythagoras/Documents/PhD/Evolution/FL/ITPKA/Motif containing the PSSM CSV files.
  + PSSM CSV Files: Files in the directory containing position-specific scoring matrices.
* **Outputs:**
  + Sequence Logo Images: Sequence logo images saved as PNG files for each PSSM.

**Code 021: Construction of an Exemplary Phylogenetic Tree**

* Function: This script constructs an exemplary phylogenetic tree based on hardcoded taxonomic groups of interest. The tree is generated using data retrieved from the NCBI taxonomy database, and it is saved in Newick format with internal node names.
* Inputs:
  + Hardcoded Taxonomic Groups:
    - A list of specific taxonomic groups including 'Chondrichthyes', 'Cladistia', 'Coelacanthiformes', 'Ceratodontiformes', 'Amphibia', 'Actinopteri', 'Mammalia', 'Lepidosauria', 'Crocodylia', 'Testudines', and 'Aves'.
* **Outputs:** Exemplary Phylogenetic Tree: A tree structure saved in Newick format (phylogenetic\_tree\_with\_names.nw), including internal node names that represent the taxonomic hierarchy.

**Code 022: Extracting, Filtering, and Comparing Sequence Information for a Clade**

* **Function:** This script extracts sequences corresponding to specific identifiers from a FASTA file and filters them based on adjusted identifiers from an annotation file. The filtered sequences are saved into a new FASTA file. Additionally, the script can be used to convert clade-specific isoform trees into sequence data, allowing for a comparison between the sequences obtained and those annotated in public databases like NCBI.
* **Inputs:**
  + FASTA File: Contains the sequences to be filtered. Examples include:
    - Filtered\_FL\_ITPKA.fasta (for general use).
    - ITPK\_meanPSSM\_cut15\_002.fasta or the ITPKA-specific version (for clade-specific analysis).
  + Annotation File: Contains the identifiers to be used for filtering. Examples include:
    - ITPK\_meanPSSM\_cut15\_002.fasta (for general use).
    - ITPKA\_Annotation.fasta (for clade-specific analysis).
* **Outputs:**
  + Filtered FASTA File: Contains sequences that match the adjusted identifiers from the annotation file (e.g., Filtered\_FL2\_ITPKA.fasta).
  + Matching and Non-Matching Count: The script prints the number of sequences that were successfully matched and those that were not.
  + Comparison of Sequence Counts: By comparing the number of sequences obtained from the general PSSM (e.g., ITPK\_meanPSSM) with those from isoform-specific annotations (e.g., ITPKA), one can identify new sequences similar to the isoform that were not previously annotated.

**Code 023C: Retrieving Accession Numbers and FLDNA Sequences from NCBI**

* **Function:** This script searches the NCBI database for genes in specific organisms, retrieves their accession numbers, and then fetches corresponding DNA sequences. It checks whether a given protein motif is present in the translated DNA sequence and extracts the nucleotide sequence corresponding to the motif.
* **Inputs:**
  + FASTA File: Filtered\_ITPKB.fasta containing sequences to be processed.
  + Hardcoded Parameters:
    - Gene Name: The gene name to search for in NCBI (e.g., itpkb).
* **Outputs:**
  + DNA Sequences FASTA File: A FASTA file containing the nucleotide sequences that correspond to the identified motifs in the gene of interest.
  + Search Results Summary: The script prints the number of successful and failed retrievals.

**Code 024: Filtering Ambiguous DNA Sequences and Removing Stop Codons**

* **Function:** This script filters DNA sequences by removing those with ambiguous nucleotides, trimming sequence IDs, removing stop codons, and ensuring that sequences are of a length divisible by three. The filtered sequences are saved in a new FASTA file.
* **Inputs:** FASTA File: ITPKA\_FL\_DNA\_Sequences.fasta containing DNA sequences to be filtered.
* **Outputs:** Filtered DNA Sequences FASTA File: A FASTA file containing sequences that passed all filters (ITPKA\_FL\_DNA\_Sequences\_filtered.fasta).

**Code 025: Comparing Coding Sequences of Full-Length and Flanking Motifs**

* **Function:** This script compares coding sequences (CDS) from two FASTA files: one containing full-length (FL) DNA sequences and the other containing flanking motif DNA sequences. It identifies sequences that are common between the two files based on their headers, and filters them into separate output FASTA files. The script also provides a summary of the number of sequences before and after filtering.
* **Inputs:** 
  + FL FASTA File: Contains the full-length DNA sequences to be compared (e.g., ITPKA\_FL\_DNA\_Sequences.filtered.fasta)
  + Flank FASTA File: Contains the flanking motif DNA sequences to be compared (e.g., ITPKA\_Flank\_DNA\_Sequences.filtered.fasta)
* **Outputs:**
  + Filtered FL FASTA File: A FASTA file containing the full-length sequences that are also present in the flank file (e.g., ITPKA\_FL\_DNA\_Sequences\_Common.fasta).
  + Filtered Flank FASTA File: A FASTA file containing the flanking sequences that are also present in the FL file (e.g., ITPKA\_Flank\_DNA\_Sequences\_Common.fasta).
  + Summary of Sequence Counts: The script prints the number of sequences in each input file before filtering, the number of sequences in common, and the number of sequences in both files after filtering.

**Codon Alignment with Prank and subsequent IQTree2**

A codon sequence alignment was performed with Prank using the following command:   
/home/pythagoras/Programs/PRANK/prank.linux64.170427/prank/bin/prank -d=Lifeact\_FL\_DNA\_Sequences\_Common.fasta -codon -F

Then IQTree2 was run on the full length aligned sequence:   
iqtree2 -s output.best.fasta -m GTR -bb 1000 -nt 11

Jalview was used to visualize the trimmed motif sequence and T-Coffee with a gap opening penalty of -20 and gap extension penalty of -5.

**dN/dS Analysis using Hyphy**

hyphy fel --alignment Flank.fasta --tree output.best.fasta.treefile

**Ancestor Sequence reconstruction using Hyphy**

hyphy /home/pythagoras/Downloads/FitMG94.bf --alignment /home/pythagoras/Documents/PhD/Evolution/FL/Shroom3/dNdS/Flankoutput1.best.fasta --tree /home/pythagoras/Documents/PhD/Evolution/FL/Shroom3/dNdS/FLoutput.best.fasta.treefile --save-fit /home/pythagoras/Documents/PhD/Evolution/FL/Shroom3/dNdS/Shroom3\_best1.fit

hyphy /home/pythagoras/Downloads/AncestralSequences.bf --fit /home/pythagoras/Documents/PhD/Evolution/FL/Shroom3/dNdS/Shroom3\_best1.fit --output /home/pythagoras/Documents/PhD/Evolution/FL/Shroom3/dNdS/Shroom3\_best1.json