Title: Phylogenomic prediction of interaction networks in the presence of gene duplication

Authors:
Tony C Gatts[1], Chris deRoux[1], Linnea E Lane[2], Monica Berggren[1], Elizabeth A Rehmann[2], Emily N Zak[2], Trinity Bartel[2], Luna L'Argent[2], Daniel B Sloan[1], and Evan S Forsythe[2,3*]

Affiliations:
[1]Colorado State University, Department of Biology
[2]Oregon State University-Cascades, Biology Program
[3]Oregon State University, Department of Integrative Biology
[*]Correspondence

Abstract:
Assigning gene function from genome sequences is a rate-limiting step in molecular biology research. A protein's position within an interaction network can potentially provide insights into its molecular mechanisms. Phylogenetic analyses of evolutionary rate covariation (ERC) have been shown to be effective for large-scale prediction of functional interactions from protein sequence data. However, gene duplication, gene loss, and other sources of phylogenetic incongruence are barriers for analyzing ERC on a genome-wide basis. Here, we developed *ERCnet*, a bioinformatic program designed to overcome these challenges, facilitating efficient all-vs-all ERC analyses for large protein sequence datasets. We compiled a sample set of 35 angiosperm genomes to test the performance of *ERCnet,* including its sensitivity to user-defined analysis parameters such as input dataset size, branch-length measurement strategy, and significance threshold for defining ERC hits. We find that our novel 'branch-by-branch' length measurements outperforms 'root-to-tip' approaches in most cases, offering a valuable new strategy for performing ERC even in the presence of extensive gene duplication. Further, we demonstrate that the number of genomes and the species composition both have profound effects on the genes that are predicted to interact. Our systematic exploration of the performance of *ERCnet* provides a roadmap for design of future ERC analyses to predict functional interactions in a wide array of genomic datasets. *ERCnet* code is freely available at *https://github.com/EvanForsythe/ERCnet*.

Introduction:

Proteins that interact with each other often exhibit correlated rates of sequence evolution across a phylogeny, which have been attributed to shared selective pressures or reciprocal coevolution (Clark et al. 2012). The statistical signature of these interactions is known as evolutionary rate covariation (ERC). ERC analyses have been applied in a wide variety of organisms (Steenwyk et al.; Goh et al. 2000; Ramani and Marcotte 2003; Sato et al. 2005; Clark and Aquadro 2010; De Juan et al. 2013; Wolfe and Clark 2015; Yan et al. 2019; Forsythe et al. 2021; Rei Liao et al. 2022; Tao et al. 2024), and bioinformatic tools have been developed to use ERC to identify novel interaction partners (Wolfe and Clark 2015; Steenwyk et al. 2021). Moreover, ERC analyses are beginning to be applied in a genome-wide manner to yield large sets of predicted protein-protein interactions (i.e, interactome networks) (Steenwyk et al. 2021; Steenwyk et al. 2022). A network view of genetic interactions has proven valuable in revealing interconnected nature of biological systems (Wright et al.; Mao et al. 2009; Rao and Dixon 2019), and ERC has the potential to be a powerful and efficient method to infer such interaction networks.

As we scale ERC analyses to more taxonomic groups and to larger numbers of genes, phylogenetic incongruence arising from gene duplication/loss, reticulate evolution, and incomplete lineage sorting becomes increasingly problematic (Degnan and Rosenberg 2013; Hahn and Nakhleh 2015). These processes can all lead to trees inferred from different loci (i.e. gene trees) having different branching patterns (i.e., topologies) from each other. Discordant gene trees make it difficult to directly compare evolutionary rates across gene trees because it is unclear which branches correspond to each other. Prior studies have overcome this challenge through combinations of (1) focusing on groups with low levels of phylogenetic incongruence and gene duplication/loss, (2) filtering gene families to retain only gene families with one-to-one orthology, (3) constraining gene tree topologies to the accepted species tree, and (4) measuring branch lengths in a topology-agnostic root-to-tip manner. These strategies have worked well when the underlying level of phylogenetic incongruence is minimal; however, they are insufficient in taxonomic groups such as plants that tend to experience more complex genome evolution with extensive gene and whole-genome duplication (Wendel 2015; Panchy et al. 2016; Forsythe et al. 2020). Thus, additional methods are needed to perform ERC analyses in such taxa. A common approach to performing analyses in the face phylogenetic incongruence is to incorporate gene-tree/species-tree (GT/ST) reconciliation (Vernot et al. 2008; Stolzer et al. 2012; Wu et al. 2014; Comte et al. 2020). GT/ST reconciliation regards phylogenetic incongruence not as noise, but as a signal with which to infer evolutionary events such as gene duplication. Importantly, GT/ST reconciliation also provides a framework for interpreting multiple gene trees in the shared context of a prevailing species tree, which can provide a valuable basis for making direct comparisons between discordant gene trees.

We previously developed a GT/ST reconciliation-based approach to performing genome-wide ERC in plants (Forsythe et al. 2021) that operated at the level of a 'one-vs-all' comparison strategy, in which a single locus was tested against a genome-wide set of genes. The biological findings from this approach suggested that large groups of functionally interacting proteins exhibit correlated evolution. However, the limited lens of one-vs-all comparisons only allowed for indirect tests of this hypothesis. A network view of the interactome is needed to assess whether large sets of proteins with correlated rates of evolution share functional relationships. Such a network view requires ERC to be applied in an 'all-vs-all' manner that tests all pairwise

combinations of proteins. This added dimensionality introduces and exacerbates scaling challenges to ERC analyses.

Here, we present *ERCnet*, a program for performing all-vs-all ERC analyses in the presence of phylogenetic incongruence. We describe our phylogenomic workflow, which automates and parallelizes the computationally intensive analytical steps, and our novel approach to performing branch-by-branch ERC analysis even in trees with gene duplicates. We compile a test dataset of 35 plant genomes and use a randomized subsampling approach to test *ERCnet* performance in response to user-defined analysis parameters and experimental design choices, thus providing an evidence-based roadmap for future ERC analyses.

Results:

Overcoming gene duplication/loss/incongruence with branch length reconciliation:

ERCnet (*https://github.com/EvanForsythe/ERCnet*) is a computational pipeline designed to predict functional interactions among proteins based on phylogenetic signatures of correlated evolution and overcome key scaling challenges associated running all-vs-all analyses on a genome-wide level (Fig. 1). *ERCnet* uses the output of the gene family clustering program *Orthofinder* (Emms and Kelly 2015, 2019), thus making it easy to plug *ERCnet* into existing genomic workflows. Our pipeline employs phylogenetic bootstrap resampling and GT/ST reconciliation strategies to account for phylogenetic incongruence and uncertainty, thus accommodating gene families that have undergone duplication and loss. *ERCnet* automates and parallelizes the computational steps needed to achieve genome-wide network analyses of ERC and consists of four major analytical steps, (1) phylogenomic analyses, (2) GT/ST reconciliation, (3) ERC analyses, and (4) network analyses (See Methods; Fig 1A). In a separate study, we applied *ERCnet* to an angiosperm dataset, revealing a highly connected network with a module that is enriched for proteins involved in plastid proteostasis (Gatts et al., *In preparation*), thus providing a proof of principle for the utility of *ERCnet* in predicting functional interactions. Below, we systematically explore the performance of *ERCnet* on a variety of plant genomic datasets to identify and optimize the parameters that influence computational prediction of interaction networks.

In addition to automating standard phylogenetic analyses, *ERCnet* provides a novel procedure for comparing rates in a branch-by-branch (BXB) manner. Some previous implementations of ERC have employed a root-to-tip (R2T) methodology for measuring branch lengths. The R2T approach simplifies the task of parsing gene trees (especially when incongruence is present); however, it results in statistical non-independence among datapoints used in correlation analyses because different species can share internal branches within a tree. This statistical shortcoming has been previously recognized (Yan et al. 2019; Forsythe et al. 2021; Smith et al. 2024) but was largely viewed as a 'necessary evil' due to lack of a method for traversing discordant gene trees to extract relevant branches for direct comparisons.

Our novel method, branch-length reconciliation (BLR), provides a strategy for avoiding the statistical shortcomings of the R2T method. BLR extracts individual internal and external branch length measurements in a species-tree-aware manner, which allows for direct comparisons across gene trees. The major barrier to performing BXB-based ERC between incongruent phylogenetic trees has been that it is difficult to systematically define which branches (if any) are shared

between two trees with potentially different taxon-composition and topology. The premise of GT/ST reconciliation methods is that all gene trees evolve within the context of an overarching species tree. This means that the species tree is the common denominator that connects all gene trees in a phylogenomic dataset. BLR uses the species tree as the common comparison point by tallying branch length information from individual gene trees in context of the species tree. BLR accomplishes this task by incorporating the labeled coalescent tree reconciliation structure employed in *DLCpar* (Wu et al. 2014), which provides information that maps gene tree nodes to species tree nodes. Using this information BLR iteratively extracts the branch(es) on each gene tree that correspond to each species tree branch. During this process BLR also tracks whether a duplication event occurred in the gene tree and, if so, takes the average branch length from the resulting paralogs. When a gene tree lacks information for a given species tree branch, BLR stores an "NA" for that branch. The output of BLR is a table of branch length measurements for each gene tree that are standardized by the species tree branches, allowing comparison across gene trees. This approach provides the ability to extract branch-specific evolutionary rate information from gene trees with incongruent topologies and histories of gene duplication and loss, enabling ERC analyses to include larger gene sets and be applied to a wider range of taxa.

Proteome coverage and statistical power for ERCnet analyses of sample datasets

ERC has been applied in a variety of organisms with considerable success. However, a systematic evidence-based exploration of the power, accuracy, and efficiency of the ERC method has never been performed. A consequential decision in ERC analyses is the number of species to include. Currently, researchers have little guidance in making taxon-sampling choices for ERC analyses, resulting in somewhat arbitrary choices that can likely have a substantial impact on ERC performance. To address this limitation, we compiled a set of 35 plant proteomes (i.e., protein sequences from all annotated genes in the genome), spanning angiosperm diversity (Table S1). We randomly subsampled sets of these proteomes to create test datasets of varying sizes. The resulting datasets contained n=6, n=11, n=18, and n=23 in-group taxa (see Methods). Three random replicates were performed for each n value, resulting in a total of 12 datasets, each of which was subjected to full *ERCnet* analyses (Fig. 2).

The first step of *ERCnet* is to filter *Orthofinder* data to identify the gene families ("orthogroups") that are suitable for phylogenomic analyses by applying filtering cutoffs based on minimum number of species represented and maximum number of gene copies per species. *ERCnet* provides a parameter scan option for users to make an informed choice on filtering settings (see Methods). To understand how filtering parameters affect dataset size, we applied a filtering formula across all datasets (see Methods) and found that datasets with a larger number of species yield a smaller number of genes retained after filtering (Fig. 3A). While datasets with a larger number of species tend to have more total gene families, these gene families tend to be filtered out at a disproportionally high rate based on both minimum number of species represented and maximum number of gene copies per species (Table S2), leading to a net loss of proteome coverage at larger dataset sizes.

The gene families that pass initial filtering are next run through the phylogenomic pipeline and ultimately subjected to all-vs-all ERC correlation analyses. Each pairwise ERC comparison between two gene trees constitutes a correlation analysis. The number of points for each correlation corresponds to the number of branches shared between the two gene trees being compared, which is variable because each gene may have experienced different degrees of gene duplication and loss. The number of overlapping branches is also dependent on the method used

to measure branch lengths (*i.e.* R2T vs BXB). We found that datasets with larger number of species yield a higher average number of overlapping branches (Fig. 3B and C). This result is expected because larger datasets yield larger gene trees, which inherently have more branches. Comparing R2T results (Fig. 3B) to BXB results (Fig. 3C), we see that BXB tends to yield a higher number of overlapping branches, which stems from including internal branches under BXB, while the R2T methodology is limited to the number of species in the in-group. For example, a gene tree with four ingroup species will only have four R2T measurements but will have six branches when internal branches are included. In general, the number of overlapping branches will directly impact the sample size of correlation analyses, suggesting that BXB has the potential to add statistical power to ERC analyses. Thus, there is a clear tradeoff between the proportion of the genome that can be probed with *ERCnet* (Fig. 3A) and the potential statistical power of ERC analyses (Fig. 3B and C).

Size and functional clustering of networks across datasets:

All-vs-all ERC analyses yield large tables of ERC results, including p-values calculated from both Pearson and Spearman correlation analyses. These ERC results are filtered according to user-defined cutoffs to retain the pairwise combinations displaying significant p-values, which we refer to as "ERC hits". We filtered results by either raw p-value < 0.0001 or FDR-corrected p-value < 0.01 (Fig. 4). In both cases, we required that both Pearson and Spearman p-values pass the threshold in order to designate an ERC hit. In general, R2T analyses returned a much higher number of ERC hits than BXB analyses (Fig. 4A and C vs B and D). The large number of hits observed for R2T analyses could partially result from pseudoreplication due to statistical non-independence of internal branch lengths (described above). Moreover, as expected, we found that the FDR filtering requirement is more stringent than raw p-values in most cases (Fig. 4A and B vs C and D). Interestingly, BXB ERC analyses display a weakly inverse relationship between number of species and number of ERC hits, while R2T analyses display the opposition pattern. This result suggests that there is a 'sweet-spot' in the number of overlapping branches underlying ERC analyses (Fig. 3) and that this sweet-spot may differ for BXB versus R2T, indicating that the optimal dataset size depends on the branch-length measuring method.

The number of ERC hits yielded by *ERCnet* analyses defines the total number of nodes in the interaction network output with each run of *ERCnet*. The size of the network is an important component of *ERCnet* performance; however, it is also important to understand contribution of biological signal versus noise in ERC networks. We sought a summary statistic that can serve as a proxy for biological signal in ERC networks. To this end, we incorporated subcellular localization annotations from *Arabidopsis thaliana* and used the graph-theory assortativity coefficient to quantify the degree to which co-localized proteins are clustered in interaction networks (Fig. 5). For BXB analyses, we found that most *ERCnet* runs yielded networks with significantly positive assortativity coefficient, suggesting biological signal outweighs noise (Fig. 5A and C). However, R2T *ERCnet* runs show much less consistent assortativity, suggesting the noise outweighs signal in most cases (Fig. 5B and D). In all cases, there does not appear to be a clear pattern connecting number of species to the strength or significance of clustering based on the functional metric of subcellular localization. These results highlight that internal biological validation will play an important role in interpreting ERC-based interaction networks.

Overlap in interactions predicted across multiple datasets:

This study was primarily focused on understanding how different datasets and parameters affect the number of ERC hits produced by *ERCnet*. While it is not the focus of this study to predict novel functional protein-protein interactions, it is important to understand the consistency with which individual interactions are predicted across different datasets. To explore consistency of *ERCnet* predictions, we compared the overlap of specific *ERCnet* hits (i.e. pairs of proteins predicted to coevolve) across different taxon sampling datasets. We found that there were no ERC hits that persisted across all 12 *ERCnet* runs. The majority of ERC hits for given *ERCnet* runs were non-overlapping 'singletons' (i.e. found in a single *ERCnet* run). The highest degree of overlap for an ERC hit was three-way overlap (an ERC hit found in three *ERCnet* runs) for BXB runs and six-way overlap (an ERC hit found in six *ERCnet* runs) for R2T runs (Fig. 6A-D, blue bars). We might expect background level of overlap, even if ERC hits were driven entirely by noise, so we next asked if our observed level of overlap exceeds a randomized background. We, indeed, found that the randomized datasets showed a low level of overlap; however, importantly the observed data exceeded the randomized data in all cases, especially at the rightward tail of the distribution (high-degree of overlap) (Fig. 6E-H). We performed permutation tests and found that this trend is highly significant ($p << 0.01$). Taken together, these results show that ERC signal rises above background noise and persists across *ERCnet* runs that use different input data. However, this analysis also demonstrates the importance of input dataset choice on *ERCnet* results.

Parallel processing and computational runtime

Scaling ERC analyses to whole proteomes involves computationally intensive statistical steps that are best suited for high-performance computing (HPC). *ERCnet* enables parallelization at multiple steps, including customizable front-end vs. back-end parallelization strategies at the most computationally intensive tree-inference steps in the pipeline. These built-in parallelization features allow users to tailor *ERCnet* to perform optimally in a variety of computing environments. We ran *ERCnet* on our test datasets on a server, utilizing 24 parallel threads for the time-consuming phylogenomics steps. Through our early testing of *ERCnet*, we found that the pairwise ERC analysis steps (implemented with ERC_analyses.py) suffered a performance deficit when run on more than eight parallel threads, perhaps due to a parallel computing phenomenon known as 'thrashing', in which increased resource allocation comes at an overhead cost for the operating system to schedule, resulting in reduced efficiency. Therefore, we run this step using only eight threads. Total runtimes for full *ERCnet* analyses ranged from ~10-30 hours for most *ERCnet* runs, with the notable exception on one run (n=18, replicate 1) taking more than 130 hours (Fig. 7A). This abnormally high runtime was predominantly driven by the pairwise ERC step. This step is expected to scale geometrically with number of genes in the dataset, which could explain some of the increased runtime because that replicate had much higher gene retention than other replicates with the same number of species (Fig. 3). However, the extent of this runtime increase cannot be explained by dataset size alone, highlighting that performance may vary in unexpected ways (see Discussion). We tested the performance of the phylogenomics steps in a parallel versus non-parallel architecture and found that 24-thread parallelization decreased runtime 9.49-fold on average compared to a single thread (Fig. 7B). While the maximum number of species included in our test datasets (23 ingroup species) is modest in comparison to some implementations in non-plant taxa (Steenwyk et al. 2022; Tao et al. 2024), the built-in parallelization features of *ERCnet* show potential to help enable high-throughput analyses in reasonable timeframes with modest computational resources.

Discussion:

Optimized design for ERC analyses

The choice of how many taxa to include is typically made in the early stages of planning an ERC study, yet the field lacks clear guidance in this area. A wide range of taxon sample sizes have been used in past studies, ranging from 18 budding yeast species (Clark et al. 2012) to 472 insect species (Tao et al. 2024). To our knowledge, past choices are mainly driven by the availability of publicly available genome/proteome sequences and are likely aimed at including the maximum possible number of taxa under the rationale that this is favorable because it increases statistical power. The datasets tested here represent only a modest range in size and extend to smaller numbers of species than have typically been used in ERC, yet we were able to detect patterns across this range that draw the 'more is more' rationale into question. Most notably, we demonstrate a tradeoff between the proportion of the proteome that passes filters as suitable to study via ERC and the potential statistical power of those ERC analyses gained in large datasets (Fig. 3). Our gene family filtering criteria are more lenient than the criteria typically applied in ERC studies, but even so, our largest datasets (24 ingroup species), restricted the number of gene families passing filters to ~5000 proteins (Fig. 3A), which makes up only 18% of the Arabidopsis proteome. This suggests that there is at least some benefit to limiting sampling sets to a modest number of species. Consistent with this idea, we did not observe a clear impact of dataset size on the functional clustering of ERC networks, with datasets as small as n=6 showing significant clustering (although with a large degree of variance across replicate sample datasets) (Fig. 5). Indeed, branch-length measurement strategy (R2T vs BXB) has a larger influence on the ability of *ERCnet* to produce functionally clustered networks, with 22/24 BXB networks versus 9/24 R2T networks achieving significant clustering, respectively. By this metric, our BXB-based ERC appears to outperform the commonly used R2T method. This boost in performance may be owed to the additional internal branch length datapoints made available in the BXB method or by the ability to reduce the statistical non-independence/pseudo-replication problem inherent to the R2T method (see Results). In any case, our results suggest that combining a moderately sized dataset with the BXB branch length method, could present an optimal balance of proteome coverage and statistical power.

It should be noted that the test set of species included here were selected based on model organism status and phylogenetic representation, but not based on organismal biology or ecology features. This is a major difference from some prior studies (Williams et al. 2019; Forsythe et al. 2021) that have incorporated *a priori* knowledge of important transitions in evolutionary rates into taxon sampling decisions. The use of ERC to explore functional modules that coevolve during specific evolutionary transitions (Gatts et al., *In preparation*) or to explore how rate variation interacts with complex categorical (Redlich et al. 2023) or quantitative trait evolution (Kowalczyk et al. 2019) is an exciting new direction (Hu et al. 2019; Treaster et al. 2023; Yan et al. 2023). In such cases, it may make sense to prioritize sampling species that represent important evolutionary transitions or trait values. Although we have not explored this dimension of taxon sampling here, it is possible that *a priori* information of phenotypic and evolutionary rate variation could alter the dynamics of how dataset size influences ERC analyses.

Dataset dependence of ERC analyses

A surprising outcome of our analysis is that the ERC hits produced by different runs of *ERCnet* show only a modest degree of overlap (Fig 7A). While the level of overlap we observed clearly exceeds a random background (Fig. 7B), the lack of any ERC hit that was observed across more than half of *ERCnet* runs raises concerns about the reproducibility of ERC results. On the other hand, the lack of overlapping hits may suggest that ERC signatures reflect the specific biology of the taxa included in an *ERCnet* analysis. This demonstrates that applying ERC in unique sets of taxa has the potential to identify novel interactions. Our results also highlight that ERC does not return all protein-protein interactions in the cell. Instead, ERC is equipped to return the interactions that have been influenced by variable selection pressures across the chosen taxa. Taken together, the specific set of species included, a decision often made somewhat arbitrarily or based purely on data availability, has a heavy influence on which proteins are predicted to interact.

It is unclear whether disparities in ERC results are driven by differences in biology, genome evolution, or data quality among sample species or by a separate, unknown factor. In any case, our results suggest that it may be of value to incorporate a taxon subsampling approach that performs genome-wide ERC analyses on several randomly selected subsets of the larger pool of taxa into ERC workflows, especially if the goal is to identify a shortlist of the most high-confidence, reproducible ERC hits. Adding such a layer of complexity to analyses would introduce increased computational requirements, but the efficiency improvements demonstrated in this study lay the groundwork for developing such a feature in a future version of *ERCnet*.

Materials and methods

Obtaining and processing randomized plant proteome datasets

We created a pool of 35 plant proteome sequence dataset (Table S1), which included a mixture of crop species and model organisms, chosen for their general high-quality genomic resources. Sequences were obtained from the *Phytozome* plant comparative genomics portal (release 13) (Goodstein et al. 2012). We downloaded the 'primary transcripts only' version fasta files for each species to avoid splice isoforms being mistaken for paralogs. Fasta files were reformatted to standardize sequence ID formatting (e.g. "Atha__AT1G06950" for an *Arabidopsis thaliana* protein sequence), using *SeqKit* Unix tools (Shen et al. 2016).

From our pool of 35 plant genomes, we randomly selected species for datasets of size n=6, n=11, n=18, and n=23, where n refers to the number of ingroup species included. We performed three replicates for each n value, each with its own randomly selected set of taxa. All datasets included *Amborella trichopoda* as an outgroup and *Arabidopsis thaliana* as a common focal species for cross-dataset comparisons. Three replicates of randomly selected taxa were performed for each n value, resulting in a total of 12 datasets, each of which was subjected to full *ERCnet* analyses.

ERCnet analyses require clustered gene families as input, which we obtained by running *Orthofinder* (v2.2.5) (Emms and Kelly 2015, 2019). We ran *Orthofinder* separately on each test dataset using the following command.

```
orthofinder -f <path/to/dir/containing/proteomes/> -y -X -M msa -t <number of
threads available on computing system>
```

Orthofinder species tree outputs were manually inspected to verify that *Amborella trichopoda*

was the outgroup. When this was not the case, we re-ran *Orthofinder* with a constrained species tree consistent with established species relationships (Jansen et al. 2007). *Orthofinder* "hierarchical orthogroups' (HOGs) for each dataset were used as input for each *ERCnet* run.

Setting up computing environment and installing ERCnet dependencies:

ERCnet has been tested extensively on multiple MacOS and Linux-based computers. For this study, we ran all ERCnet analyses on an HPC Linux server, which has AMD series 7002/7003 CPUs with a core clock speed of 2.6ghz and 128 total threads (64 physical threads and 64 virtual threads), 1TB (16x64 GB) DDR4 - 3200 RAM, two Micron 7300 PRO 480GB NVMe M.2 solid state drives for the operating system, and 256 TB hard disk for storage. The server's operating system is Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-84-generic x86_64). We created two *Conda* virtual environments to accomplish *ERCnet* steps that require *Python 3* and *Python 2*. Our environments utilized *Python* 3.9.7 and *Python* 2.7.18, respectively. The *Conda* environments, including *ERCnet* dependencies, can be recreated using the .yml files provided with the *ERCnet* package. All *ERCnet* runs presented in this study were performed using *ERCnet* (v1.0.0) (see *ERCnet* GitHub page for release information).

Phylogenomic analyses

The first steps of *ERCnet* are implemented in the *Phylongenomics.py* script, which filters gene families by user defined parameters (see -r and -p flag below). Next, multiple sequence alignments are generated using the L-INS-I algorithm in *MAFFT* (v7.515) (Katoh and Standley 2013). Next, *TAPER* (Zhang et al. 2021) is (optionally) used to identify outlier sequences and filter out noise from heterogeneity. *GBLOCKS* (*v0.91b)* (Castresana 2000) is then used to trim poorly aligned regions from alignments. Alignments that are shorter than 100 amino acids (set with the '-l 100' flag) following trimming are discarded. Below is an example command used to call Phylogenomics.py, including descriptions of all required arguments. For detailed descriptions of optional arguments, see the *ERCnet* GitHub page.

```
./Phylogenomics.py -j <Job Name> -o <OrthoFinder Results Files> -x <Python
Environment Path> -m <Threads> -s -p 4 -r 11 -T <Julia Environment Path> -n 1
```

The -j flag indicates the desired name for the output folder for all *ERCnet* outputs. This 'jobname' is used to help *ERCnet* find needed input/output files throughout the *ERCnet* workflow. The -o flag indicates the path to *Orthofinder* results used in the analyses. The -x flag is the path to the user's *Python* environment where all prerequisite libraries are installed (see *ERCnet* GitHub Readme for details). The -m flag allows for parallel multithreading to speed up analysis time by using *Joblib* (see below). The -s flag allows the user to provide a custom species mapping file to help *ERCnet* assign gene names to species. The -p and -r flags are used during an initial filtering step to remove gene families with too many paralogs or too little species representation, which are unlikely to be useful in *ERCnet* analyses (e.g. gene families consisting of only 100 paralogs from one species). The -p flag indicates the maximum number of paralogs per species allowed in each gene family. For example, "-p 3" would cause *ERCnet* to exclude a gene family that contains four or more paralogs for any species. Note that *Orthofinder's* HOGs (Emms and Kelly 2019) split deep paralogs into separate gene families, so the paralogs detected during this filtering step represent only relatively recent gene duplicates. We set -p to 3 for all *ERCnet* runs used in this study. The -r flag controls the minimum number of species represented in each gene family. For example, "-r 6" indicates that at least six unique ingroup species must

be present in a gene family in order for *ERCnet* to retain the gene family. For the randomized datasets of varying size used in this study, we used set -r to the ceiling of (n/2)+1. For example, for n=18 dataset, r was set to 10. The -T flag is (optionally) used to tell *ERCnet* where to find the *Julia* installation used to run *TAPER.* The -n flag allows the user to specify the species tree node from which to obtain *Orthofinder* HOG subtrees (Emms and Kelly 2019).

To achieve parallelization for the steps that represent potential computational bottlenecks (Fig. 1), we use *Joblib*, an open-source *Python* library that uses *C* and *loky* to circumvent *Python's* Global Interpreter Lock and spawn parallel worker processes. When parallelizing *RAxML* calls within Phylogenomics.py*,* we were faced with the choice of relying on *RAxML's* built-in parallelization (back-end parallelization) versus our *Python*-based parallelization (front-end parallelization) method. To balance the two levels of parallelization, we devised three 'core distribution' strategies and tested their performance to define the strategy that *ERCnet* uses by default (see -c option described in the *ERCnet* GitHub page). A technical hurdle that we encountered during development was system crashes during runtime as a result of parallelization. After *Joblib* was fully integrated into the *ERCnet* pipeline, we began to experience inconsistent errors, resulting in script termination. Further investigation into this issue yielded a potential negative interaction between the *Numpy* library and *Joblib. Joblib* utilizes *Numpy memmap* arrays by default in order to allow *loky* workers to share large amounts of data. However, *Numpy memmap* has known issues with its API for closing out and releasing resources back to the operating system (https://numpy.org/doc/stable/reference/generated/numpy.memmap.html). We suspect that, when combined with parallel processes that iterate sequentially through large array files (like *ERCnet*), this exacerbates the issue and can result in memory leaks. Disabling *Joblib's* native use of *Numpy* memmap arrays appears to have alleviated script crashes related to excessive memory usage in *ERCnet*.

Gene-tree/species-tree reconciliation

Once gene trees are generated for all retained gene families, *ERCnet* performs a GT/ST reconciliation step via the *GTST_reconciliation.py* script. GTST_reconciliation.py employs *DLCpar* (v2.0.1 Kellis) to infer gene duplication, coalescence, and loss, which allows *ERCnet* to reconcile conflicting gene trees and species and perform BLR (See Results and Fig 1B). While *ERCnet* generally runs inside a *Python 3* environment, *DLCpar* requires *Python 2*. We use *Conda* environments to easily toggle between *Python 2* and *Python 3*. Once a *Python* 2 environment is activated, *GTST_reconciliation.py* can be called with the following command:

```
./GTST_reconciliation.py -j <Job Name>
```

The -j flag requires the same job name from the previous step to continue the analysis.

ERC analyses with branch length reconciliation

After GT/ST reconciliation, *ERCnet* performs ERC analyses in an all-vs-all fashion by comparing the branch lengths for every possible pairwise combination of gene trees. Below is an example script used to call ERC_analysis.py.

```
./ERC_analyses.py -j <Job Name> -m <Threads> -s <Focal Species> -b <Branch
Reconciliation Method>
```

The -j flag continues to use the same job name from previous steps. The -m flag determines how many threads used for parallelizing the analysis. The -s flag indicates which species should be

the focal species for the analysis. The -b flag controls which branch length correlation method (R2T or BXB) is used. For this study, we ran *ERCnet* with both methods in order to compare results.

Network analyses

The final step of *ERCnet* filters ERC results to retain ERC hits according to user-defined cutoffs and filtering criteria. *Network_analysis.py* includes options to filter results by R-squared values and p-values obtained from Pearson and/or Spearman correlation analyses. Users choose to filter by raw p-value or p-values that are multiple-test corrected with the Benjamini-Hochberg false discovery rate (FDR) method (Benjamini and Hochberg 1995). The script also includes options for building network graphs to visualize interaction networks. Nodes indicate gene families and edges between nodes indicate significant ERC signal. Many options for network visualization, clustering, and export are available (see *ERCnet* GitHub page). Additionally, *Network_analysis.py* provides the option to evaluate clustering of user-defined functional gene annotations by calculating the assortativity coefficient (Newman 2003) using the *igraph* package (Csardi and Nepusz 2006). *Network_analyses.py* was run with the following command:

```
./Network_analyses.py -j <Job Name> -m <Branch method used>  -y <Clustering
algorithm> -s <Focal Species> -f <ERC results filename>
```

The -j flag continues the same job name from previous steps. The -m flag informs the script which branch length method was used from the previous step. The -f flag indicates which ERC results file should be analyzed in the event that prior step was run multiple times. The -y flag allows the user to specify which clustering method they would like for generating network plots. The -s flag specifics which focal species the user is providing. The -f flag provides the script with the ERC results tsv file the user wishes to use for this portion of the analysis. Multiple ERC results files can be generated by *ERC_analysis.py*.

During development we tested parallelization of ERC_analysis.py with varying levels of resource allocation ranging from single threaded (non-parallelized) to maximum usage of our test system at 128 threads. We discovered that thread allocation beyond 8 resulted in significant degradation of performance, so we ran the *Network_analyes.py* with -m set to 8 for the analyses described in this study.

Analysis of ERC hit overlap

To ask if ERC hits from multiple *ERCnet* runs overlap more than would be expected by chance, we performed a permutation test in which we randomized the ERC results table. The ERC results table includes a "Gene A" and "Gene B" corresponding to the two gene families being compared for ERC signature in a given pairwise combination. First, to gauge overall overlap between different *ERCnet* runs, we simply counted the number of *ERCnet* runs in which a given Gene A and Gene B combination are present among the ERC hits. To achieve a 'null distribution', we randomized the Gene B column of the tables and again calculated ERC hit overlap across *ERCnet* runs. We replicated this procedure ten times to get an average degree of overlap for *ERCnet* runs with different parameters and we used a z-test to calculate the significance of the difference between the observed overlap and the null distribution.

Works Cited

Benjamini Y and Hochberg Y. Controlling the False Discovery Rate : A Practical and Powerful Approach to Multiple Testing. Royal Statistical Society. 1995:**57**(1):289–300.

Castresana J. Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. Mol Biol Evol. 2000:**17**(4):540–552. https://doi.org/10.1093/oxfordjournals.molbev.a026334

Clark NL, Alani E, and Aquadro CF. Evolutionary rate covariation reveals shared functionality and coexpression of genes. Genome Res. 2012:**22**(607):714–720. https://doi.org/10.1101/gr.132647.111.Guerois

Clark NL and Aquadro CF. A novel method to detect proteins evolving at correlated rates: Identifying new functional relationships between coevolving proteins. Mol Biol Evol. 2010:**27**(5):1152–1161. https://doi.org/10.1093/molbev/msp324

Comte N, Morel B, Hasic D, Gueguen L, Boussau B, Daubin V, Penel S, Scornavacca C, Gouy M, Stamatakis A, et al. Treerecs: An integrated phylogenetic tool, from sequences to reconciliations. Bioinformatics. 2020:**36**(18):4822–4824. https://doi.org/10.1093/bioinformatics/btaa615

Csardi G and Nepusz T. The igraph software package for complex network research. InterJournal Complex Systems. 2006:**Complex Sy**(1695):1695.

Degnan JH and Rosenberg NA. Discordance of species trees with their most likely gene trees: A unifying principle. Mol Biol Evol. 2013:**30**(12):2709–2713. https://doi.org/10.1093/molbev/mst160

Emms DM and Kelly S. OrthoFinder: solving fundamental biases in whole genome comparisons dramatically improves orthogroup inference accuracy. Genome Biol. 2015:**16**(1):157. https://doi.org/10.1186/s13059-015-0721-2

Emms DM and Kelly S. OrthoFinder: Phylogenetic orthology inference for comparative genomics. Genome Biol. 2019:**20**(1):1–14. https://doi.org/10.1186/s13059-019-1832-y

Forsythe ES, Nelson ADL, and Beilstein MA. Biased gene retention in the face of introgression obscures species relationships. Genome Biol Evol. 2020:**12**(9):1646–1663. https://doi.org/10.1093/GBE/EVAA149

Forsythe ES, Williams AM, and Sloan DB. Genome-wide signatures of plastid-nuclear coevolution point to repeated perturbations of plastid proteostasis systems across angiosperms. Plant Cell. 2021:1–18. https://doi.org/10.1093/plcell/koab021

Gatts TC, Rehmann EA, DB Sloan DB, & Forsythe ES. A functional module of coevolving proteins reveals novel components and cross-compartment connectivity in plant proteostasis systems. *In Preparation.*

Goh CS, Bogan AA, Joachimiak M, Walther D, and Cohen FE. Co-evolution of proteins with their interaction partners. J Mol Biol. 2000:**299**(2):283–293. https://doi.org/10.1006/jmbi.2000.3732

Goodstein DM, Shu S, Howson R, Neupane R, Hayes RD, Fazo J, Mitros T, Dirks W, Hellsten U, Putnam N, et al. Phytozome: A comparative platform for green plant genomics. Nucleic Acids Res. 2012:**40**(D1):1178–1186. https://doi.org/10.1093/nar/gkr944

Hahn MW and Nakhleh L. Irrational exuberance for resolved species trees. The Society for the Study of Evolution. 2015:**70**(1):7–17. https://doi.org/10.1111/evo.12832

Hu Z, Sackton TB, Edwards S V., and Liu JS. Bayesian Detection of Convergent Rate Changes of Conserved Noncoding Elements on Phylogenetic Trees. Mol Biol Evol. 2019:**36**(5):1086–1100. https://doi.org/10.1093/molbev/msz049

Jacob L. Steenwyk, Megan A. Phillips, Feng Yang, Swapneeta S. Date, Todd R. Graham, Judith Berman, Chris Todd Hittinger, and Antonis Rokas. An orthologous gene coevolution network provides insight into eukaryotic cellular and genomic structure and function. Sci Adv. 2022.

Jansen RK, Cai Z, Raubeson LA, Daniell H, Depamphilis CW, Leebens-Mack J, Müller KF, Guisinger-Bellian M, Haberle RC, Hansen AK, et al. Analysis of 81 genes from 64 plastid genomes resolves relationships in angiosperms and identifies genome-scale evolutionary patterns. Proc Natl Acad Sci U S A. 2007:**104**(49):19369–19374. https://doi.org/10.1073/pnas.0709121104

De Juan D, Pazos F, and Valencia A. Emerging methods in protein co-evolution. Nat Rev Genet. 2013:**14**(4):249–261. https://doi.org/10.1038/nrg3414

Katoh K and Standley DM. MAFFT multiple sequence alignment software version 7: Improvements in performance and usability. Mol Biol Evol. 2013:**30**(4):772–780. https://doi.org/10.1093/molbev/mst010

Kowalczyk A, Meyer WK, Partha R, Mao W, Clark NL, and Chikina M. RERconverge: An R package for associating evolutionary rates with convergent traits. Bioinformatics. 2019:**35**(22):4815–4817. https://doi.org/10.1093/bioinformatics/btz468

Mao L, Van Hemert JL, Dash S, and Dickerson JA. Arabidopsis gene co-expression network and its functional modules. BMC Bioinformatics. 2009:**10**:1–24. https://doi.org/10.1186/1471-2105-10-346

Newman MEJ. Mixing patterns in networks. Phys Rev E. 2003:**67**(2):026126. https://doi.org/10.1103/PhysRevE.67.026126

Panchy N, Lehti-Shiu M, and Shiu SH. Evolution of gene duplication in plants. Plant Physiol. 2016:**171**(4):2294–2316. https://doi.org/10.1104/pp.16.00523

Priedigkeit N, Wolfe N, and Clark NL. Evolutionary Signatures amongst Disease Genes Permit Novel Methods for Gene Prioritization and Construction of Informative Gene-Based Networks. PLoS Genet. 2015:**11**(2):1–17. https://doi.org/10.1371/journal.pgen.1004967

Ramani AK and Marcotte EM. Exploiting the co-evolution of interacting proteins to discover interaction specificity. J Mol Biol. 2003:**327**(1):273–284. https://doi.org/10.1016/S0022-2836(03)00114-1

Rao X and Dixon RA. Co-expression networks for plant biology: Why and how. Acta Biochim Biophys Sin (Shanghai). 2019:**51**(10):981–988. https://doi.org/10.1093/abbs/gmz080

Redlich R, Kowalczyk A, Tene M, Sestili HH, Foley K, Saputra E, Clark N, Chikina M, Meyer WK, and Pfenning A. RERconverge Expansion: Using Relative Evolutionary Rates to Study Complex Categorical Trait Evolution. Biorxiv. 2023. https://doi.org/10.1101/2023.12.06.570425

Rei Liao J-Y, Friso G, Forsythe ES, Michel EJS, Williams AM, Boguraev SS, Ponnala L, Sloan DB, and van Wijk KJ. Proteomics, phylogenetics, and co-expression analyses indicate novel interactions in the plastid CLP chaperone-protease system. Journal of Biological Chemistry. 2022:101609. https://doi.org/10.1016/j.jbc.2022.101609

Sato T, Yamanishi Y, Kanehisa M, and Toh H. The inference of protein-protein interactions by co-evolutionary analysis is improved by excluding the information about the phylogenetic relationships. Bioinformatics. 2005:**21**(17):3482–3489. https://doi.org/10.1093/bioinformatics/bti564

Shen W, Le S, Li Y, and Hu F. SeqKit: A cross-platform and ultrafast toolkit for FASTA/Q file manipulation. PLoS One. 2016:**11**(10). https://doi.org/10.1371/journal.pone.0163962

Smith CH, Mejia-Trujillo R, and Havird JC. Mitonuclear compatibility is maintained despite relaxed selection on male mitochondrial DNA in bivalves with doubly uniparental inheritance. Evolution (N Y). 2024. https://doi.org/10.1093/evolut/qpae108

Steenwyk JL, Buida TJ, Labella AL, Li Y, Shen XX, and Rokas A. PhyKIT: A broadly applicable UNIX shell toolkit for processing and analyzing phylogenomic data. Bioinformatics. 2021:**37**(16):2325–2331. https://doi.org/10.1093/bioinformatics/btab096

Steenwyk JL, Phillips MA, Yang F, Date SS, Graham TR, Berman J, Hittinger CT, and Rokas A. An orthologous gene coevolution network provides insight into eukaryotic cellular and genomic structure and function. https://doi.org/10.1101/2021.07.09.451830

Stolzer M, Lai H, Xu M, Sathaye D, Vernot B, and Durand D. Inferring duplications, losses, transfers and incomplete lineage sorting with nonbinary species trees. Bioinformatics. 2012:**28**(18):i409–i415. https://doi.org/10.1093/bioinformatics/bts386

Tao M, Chen J, Cui C, Xu Y, Xu J, Shi Z, Yun J, Zhang J, Ou G-Z, Liu C, et al. Identification of a longevity gene through evolutionary rate covariation of insect mito-nuclear genomes. Nat Aging. 2024. https://doi.org/10.1038/s43587-024-00641-z

Treaster S, Deelen J, Daane JM, Murabito J, Karasik D, and Harris MP. Convergent genomics of longevity in rockfishes highlights the genetics of human life span variation.

Vernot B, Stolzer M, Goldman A, and Durand D. Reconciliation with non-binary species trees. J Comput Biol. 2008:**15**(8):981–1006. https://doi.org/10.1089/cmb.2008.0092

Wendel JF. The wondrous cycles of polyploidy in plants. Am J Bot. 2015:**102**(11):1753–1756. https://doi.org/10.3732/ajb.1500320

Williams AM, Friso G, Wijk KJ Van, and Sloan DB. Extreme variation in rates of evolution in the plastid Clp protease complex. The plan journal. 2019:1–17. https://doi.org/10.1111/tpj.14208

Wolfe NW and Clark NL. ERC analysis: Web-based inference of gene function via evolutionary rate covariation. Bioinformatics. 2015:**31**(23):3835–3837. https://doi.org/10.1093/bioinformatics/btv454

Wright SN, Colton S, Schaffer L V, Pillich RT, Churas C, Pratt D, and Ideker T. State of the Interactomes: an evaluation of molecular networks for generating biological insights. AUTHORS. https://doi.org/10.1101/2024.04.26.587073

Wu Y, Rasmussen MD, Bansal MS, and Kellis M. Most parsimonious reconciliation in the presence of gene duplication , loss , and deep coalescence using labeled coalescent trees. 2014:475–486. https://doi.org/10.1101/gr.161968.113.Freely

Yan H, Hu Z, Thomas GWC, Edwards S V., Sackton TB, and Liu JS. PhyloAcc-GT: A Bayesian Method for Inferring Patterns of Substitution Rate Shifts on Targeted Lineages Accounting for Gene Tree Discordance. Mol Biol Evol. 2023:**40**(9). https://doi.org/10.1093/molbev/msad195

Yan Z, Ye G, and Werren JH. Evolutionary Rate Correlation between Mitochondrial-Encoded and Mitochondria-Associated Nuclear-Encoded Proteins in Insects. Mol Biol Evol. 2019:**36**(5):1022–1036. https://doi.org/10.1093/molbev/msz036

Zhang C, Zhao Y, Braun EL, and Mirarab S. TAPER: Pinpointing errors in multiple sequence alignments despite varying rates of evolution. Methods Ecol Evol. 2021:**12**(11):2145–2158. https://doi.org/10.1111/2041-210X.13696

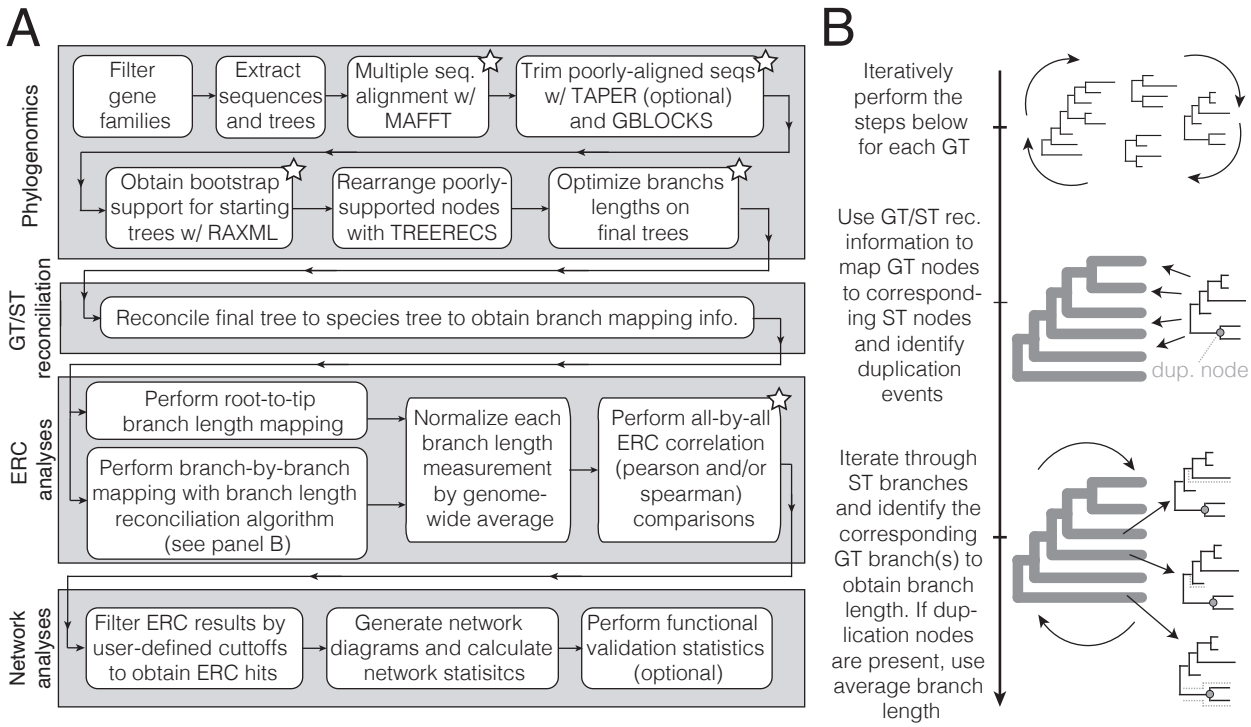

Fig. 1: *ERCnet* workflow and algorithm development. (A) Four-step analytical workflow used to analyze *Orthofinder* results (input) and generate an ERC-based interaction network and accompanying summary statistics (output). Stars indicate steps that employ parallel computing. (B) The major analytical steps of the novel 'branch length reconciliation' procedure used to calculate branch lengths on a branch-by-branch basis, using species tree (ST) and gene tree (GT) information. Rounded arrows indicate iterative processes.

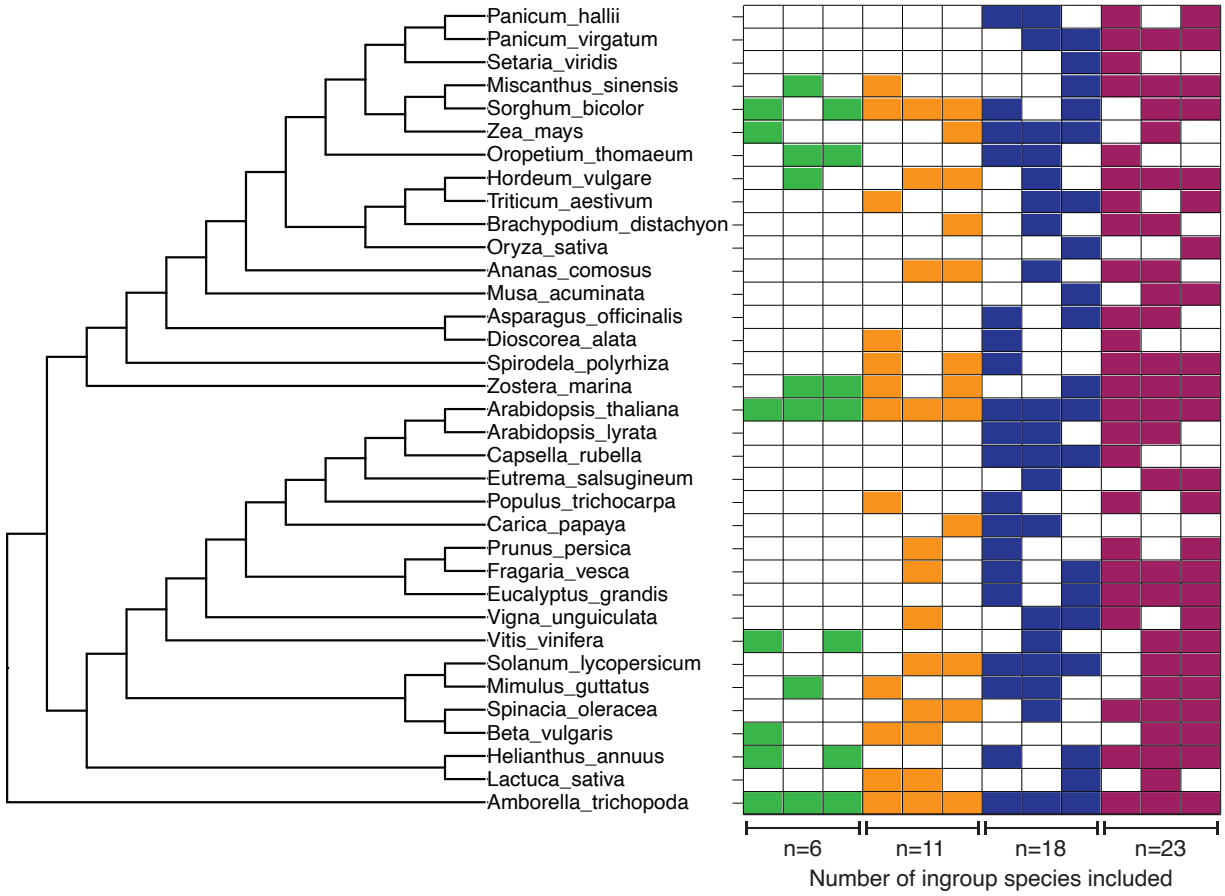

Fig. 2: Angiosperm taxon-sampling datasets used to assess *ERCnet* performance. (Left) Phylogenetic tree of the full pool of species included for random sub-sampling. (Right) Presence-absence plot indicating the species included in random datasets of each size. Three replicates were performed for each dataset size. *Amborella trichopoda* was included as an outgroup for all replicates and *Arabidopsis thaliana* was included as a common ingroup representative for all replicates.

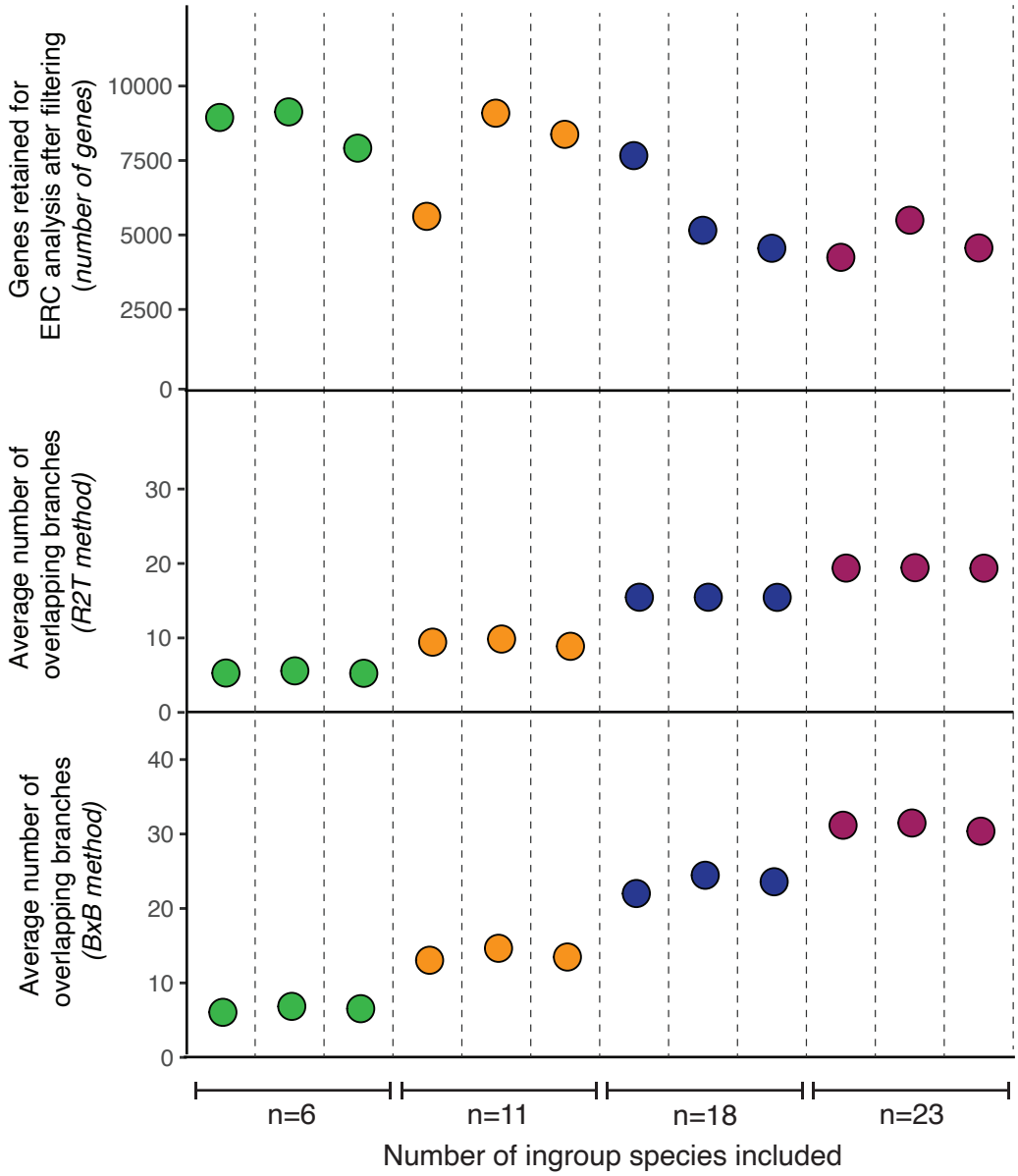

Fig. 3: Proteome coverage and overlapping branches after *ERCnet* filtering. (Top) The number of proteins retained for phylogenomic analysis after several phases of quality-control filtering that occur during the first steps of *ERCnet*. These number represent the number of proteins that are tested for interaction during 'all-vs-all' ERC analyses at later steps of *ERCnet.* (Middle and Bottom) The number of overlapping branches (i.e. points on correlation plots) among pairs of proteins during all-vs-all ERC analyses. For the root-to-tip method (middle), 'branches' refers to paths from root of tree to each tip. For the branch-by-branch method (bottom), 'branches' refers to the common branches determined by our 'branch length reconciliation' method.

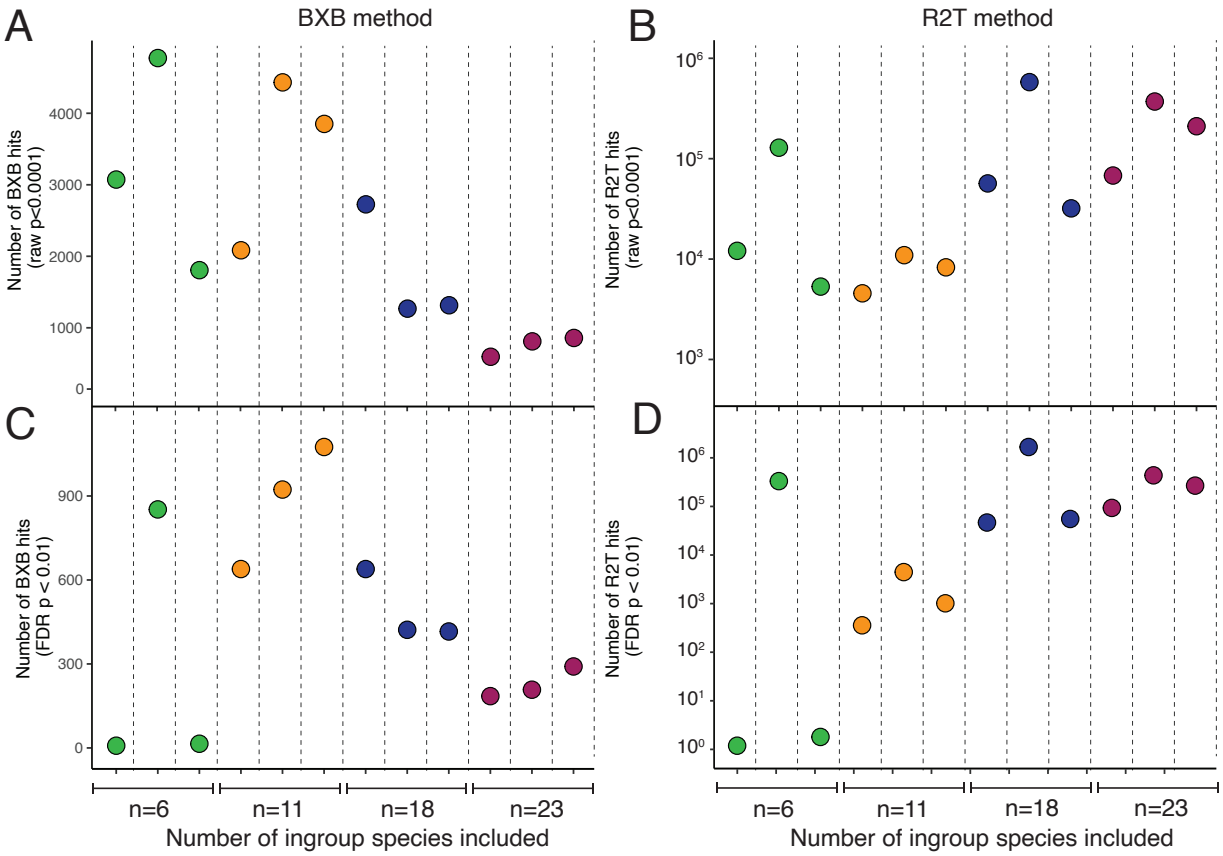

Fig. 4: Significant ERC hits from *ERCnet* analyses of test datasets. The number of pairwise ERC hits passing filters according to raw (A and B) or FDR-corrected (C and D) p-value cutoffs from p-values obtained from branch-by-branch (A and C) or root-to-tip (B and D) branch length methods. Note the log scale for panels B and D.

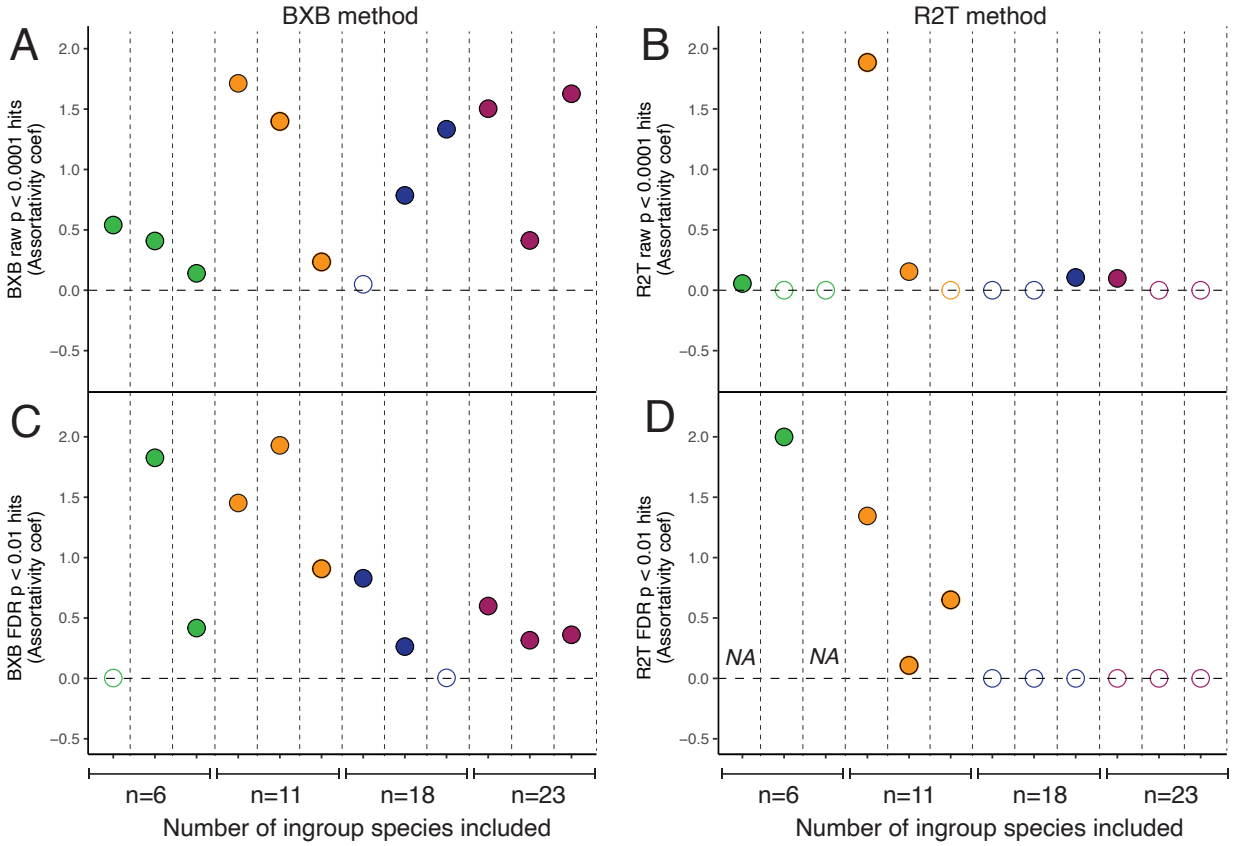

Fig. 5 Functional validation of *ERCnet* clustering. The assortativity coefficient estimated from networks of ERC hits passing filters to raw (A and B) or FDR-corrected (C and D) p-value cutoffs from p-values obtained from branch-by-branch (A and C) or root-to-tip (B and D) branch length methods. Filled points indicate the assortativity coefficient significantly differed from 0 (z-score from randomized permutation test). Significant positive assortativity indicates clustering of traits across a network. The trait measured here was the predicted targeting (plastid, mitochondrial, other) of the proteins in the network. "NA" in panel D indicate runs that yielded no significant ERC hits (networks of size 0), meaning Assortativity could not be calculated.

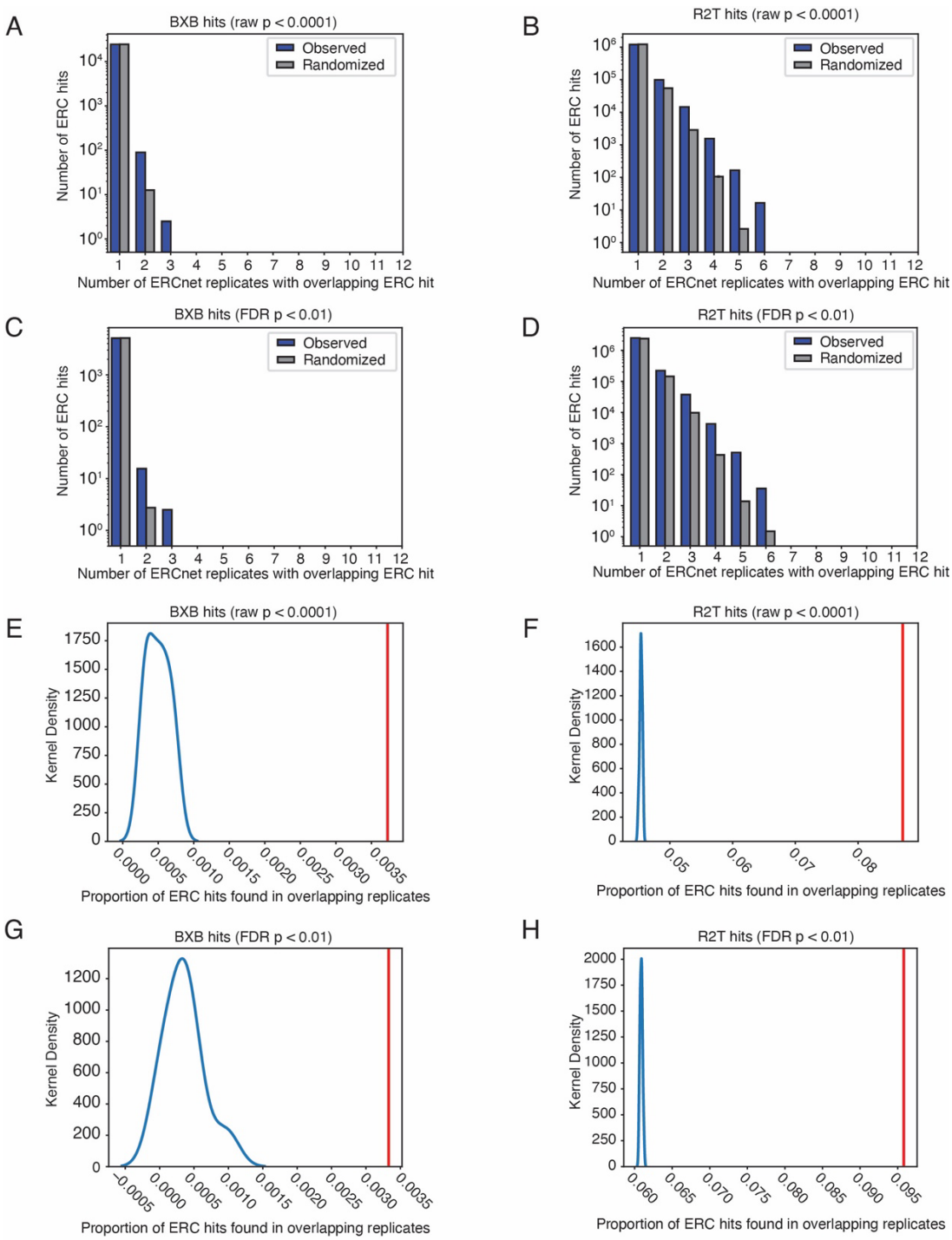

Fig. 6: Overlap in *ERCnet* hits between runs. (A-D) Bar plots describing the number of hits plotted by the number of *ERCnet* replicate runs they appeared in. Blue bars show observed values, and gray bars show averages from ten randomized replicates. The branch length method and filtering criteria used for each dataset are shown above each plot. (E-H) Kernel density plots showing the distribution of proportions of hits found in overlapping runs from randomized replicates (blue curve). Proportion for observed data from each run is shown in red.

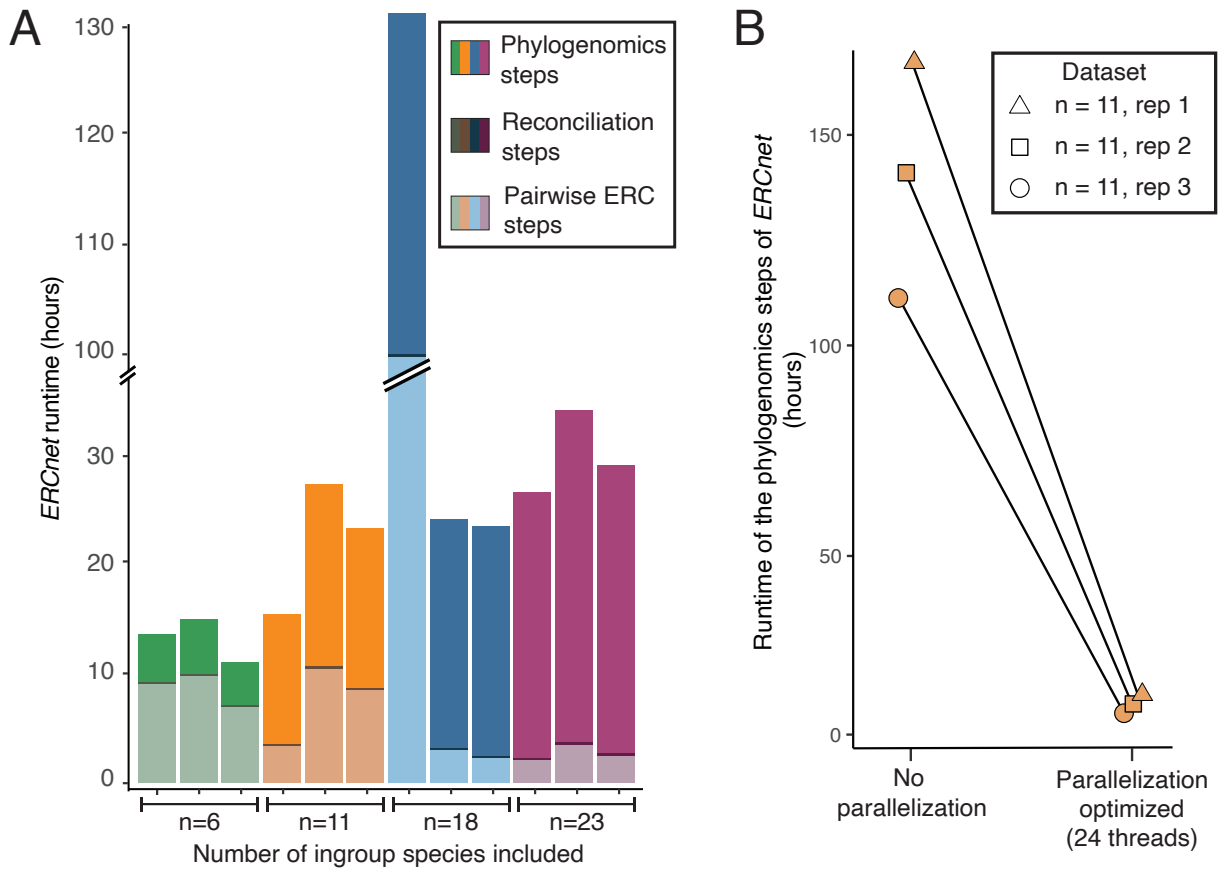

Fig. 7: Performance of *ERCnet* using parallel processing (A) Runtime of three analytical steps of *ERCnet* parallelized with 24 threads. The reconciliation steps (middle) are very fast relative to the phylogenomics steps (top) and pairwise ERC steps (bottom). (B) Runtime of the *Phylogenomics* steps of *ERCnet* on either one thread or 24 threads. Parallel runs were approximately 10-fold faster than unparallelized runs.

Table S1: Proteome data used for random sample datasets.

Species	File name	Common name	Version	Clade
Arabidopsis thaliana	Atha	Thale cress	Araport11	Eudicot
Amborella trichopoda	Atri	NA	v1.0	Outgroup
Ananas comosus	Acom	Pineapple	v3	Monocots
Asparagus officinalis	Aoff	Asparagus	V1.1	Monocots
Brachypodium distachyon	Bdis	Brachypodium	v3.1	Monocots
Dioscorea alata	Dala	Purple yam	v2.1	Monocots
Hordeum vulgare	Hvul	Barley	r1	Monocots
Musa acuminata	Macu	Banana	v1	Monocots
Miscanthus sinensis	Msin	Chinese silver grass	v7.1	Monocots
Oryza sativa	Osat	Rice	v7.0	Monocots
Oropetium thomaeum	Otho	Dessication tolerant grass	v1.0	Monocots
Panicum hallii	Phal	Hall's panic grass	v2.0	Monocots
Panicum virgatum	Pvir	Switchgrass	v5.1	Monocots
Sorghum bicolor	Sbic	Sorghum	v3.1.1	Monocots
Spirodela polyrhiza	Spol	Duckweed	v2	Monocots
Setaria viridis	Svir	Wild foxtail millet	v2.1	Monocots
Triticum aestivum	Taes	Triticum	v2.2	Monocots
Zostera marina	Zmar	Eelgrass	v2.2	Monocots
Zea mays	Zmay	Maize	RefGen_v4	Monocots
Arabidopsis lyrata	Alyr	Sand cress	v2.1	Eudicot
Beta vulgaris	Bvul	Sugar beet	EL10_1.0	Eudicot
Carica papaya	Cpap	Papaya	ASGPBv0.4	Eudicot
Capsella rubella	Crub	Pink sheapard's purse	v1.1	Eudicot
Eutrema salsugineum	Esal	Saltwater cress	v1.0	Eudicot
Populus trichocarpa	Ptri	California poplar	v4.1	Eudicot
Vitis vinifera	Vvin	Common grape vine	v2.1	Eudicot
Fragaria vesca	Fves	Strawberry	v4.0.a2	Eudicot
Eucalyptus grandis	Egra	Ecalyptus	v2.0	Eudicot
Helianthus annuus	Hann	Sunflower	r1.2	Eudicot
Mimulus guttatus	Mgut	Monkeyflower	v2.0	Eudicot
Prunus persica	Pper	Peach	v2.1	Eudicot
Solanum lycopersicum	Slyc	Tomato	TAG2.4	Eudicot
Spinacia oleracea	Sole	Spinach	Spov3	Eudicot
Vigna unguiculata	Vung	Black-eyed pea	v1.2	Eudicot
Lactuca sativa	Lsat	Lettuce	V8	Eudicot

Table S2: Gene families filtered from sample datasets.

				Number of gene families					Proportion of gene families			
Spec-ies in data-set (n)	**Rep-licate**	**Repres ent-ation filter cutoff (r)**	**Para-log filter cutoff (p)**	**Re-tained**	**Dis-qualified by r**	**Dis-qualified by p**	**Dis-qualified by p and r**	**Ortho-finder output**	**Re-tained**	**Dis-qualified by r**	**Dis-qualified by p**	**Dis-qualified by p and r**
6	1	4	3	9555	10233	2372	2922	25082	0.38	0.41	0.09	0.12
6	2	4	3	9675	12037	2764	3352	27828	0.35	0.43	0.10	0.12
6	3	4	3	8342	11732	3481	3412	26967	0.31	0.44	0.13	0.13
11	1	7	3	6085	20994	5772	6681	39532	0.15	0.53	0.15	0.17
11	2	7	3	9736	12986	2925	5588	31235	0.31	0.42	0.09	0.18
11	3	7	3	8965	13923	2881	5073	30842	0.29	0.45	0.09	0.16
18	1	10	3	8331	20881	4616	6159	39987	0.21	0.52	0.12	0.15
18	2	10	3	5628	25331	6906	8599	46464	0.12	0.55	0.15	0.19
18	3	10	3	4906	30723	7406	10596	53631	0.09	0.57	0.14	0.20
23	1	13	3	4627	30771	7391	10789	53578	0.09	0.57	0.14	0.20
23	2	13	3	5947	29620	6249	10730	52546	0.11	0.56	0.12	0.20
23	3	13	3	4988	31483	7445	11290	55206	0.09	0.57	0.13	0.20