Find relevant genes from a gene list

Alice Hermann

August 26, 2025

Contents

1 Introduction

The objective of this project is to develop a reproducible and parameter-optimized pipeline for building high-quality gene networks from an initial list. The approach combines two complementary interaction databases (StringDB and FunCoup) and leverages the Optuna optimization framework to automatically tune parameters for network expansion, filtering, and robustness assessment. The ultimate goal is to retain only added genes that are biologically plausible and consistently recovered across multiple perturbations of the input data.

2 Databases Used for Network Construction

This study integrates two complementary interaction databases: StringDB and FunCoup. By combining these resources, the pipeline benefits from the broad coverage of StringDB and the functional specificity of FunCoup, increasing the likelihood of capturing both direct and indirect gene relationships relevant to the context of interest.

2.1 StringDB

StringDB[1] [3] is a large-scale resource of known and predicted protein–protein interactions. It integrates multiple types of evidence, including curated experiments, databases, co-expression, and text mining. Its strengths include wide species coverage and a programmable REST API.

Key parameters for querying the API include:

[1] https://string-db.org/

- **add_nodes**: Number of additional nodes to be included in the extended network.
- **required_score**: Minimum confidence score required for an interaction to be included, ranging from 0 to 1000.
- **network_type**: Type of network; either *functional* (default) or *physical*.

2.2 FunCoup

FunCoup[2] [2] is a framework designed to infer genome-wide functional couplings across 22 model organisms. Functional coupling refers to general associations between genes, including direct physical interactions, shared regulatory relationships, and co-participation in pathways or biological processes.

FunCoup distinguishes six types of functional coupling: complex co-membership, metabolic pathway co-membership, shared operon, protein–protein interaction, gene regulation, and signaling pathway co-membership. Each evidence type yields a separate network, and a composite view summarizes the strongest interaction per gene pair.

Query expansion starts from the seed genes and adds the most strongly connected genes based on confidence thresholds. Four parameters can be adjusted: link confidence threshold, direction confidence threshold, number of top interactors, and number of expansion steps.

Expansion algorithms include:

- **Global**: retrieves the N strongest interactors among all query genes. An optional setting prioritizes genes connected to multiple queries.
- **Local**: retrieves N interactors independently for each query.
- **MaxLink**: identifies genes statistically more connected than expected by chance, designed for large query lists.
- **TOPAS search**: detects biologically relevant modules within networks.

3 First trials

In the first trials, parameters were set manually. We attempted to keep only genes that appeared in both databases. However, the overlap was very limited (see Figure 1). We also attempted to rank genes based on their order of appearance when expanding the networks (Figure 2).

[2] https://funcoup.org/

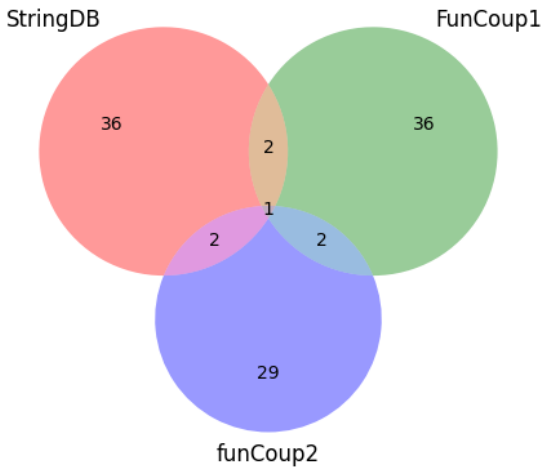

Figure 1: Venn Diagram from leucemia genes list.

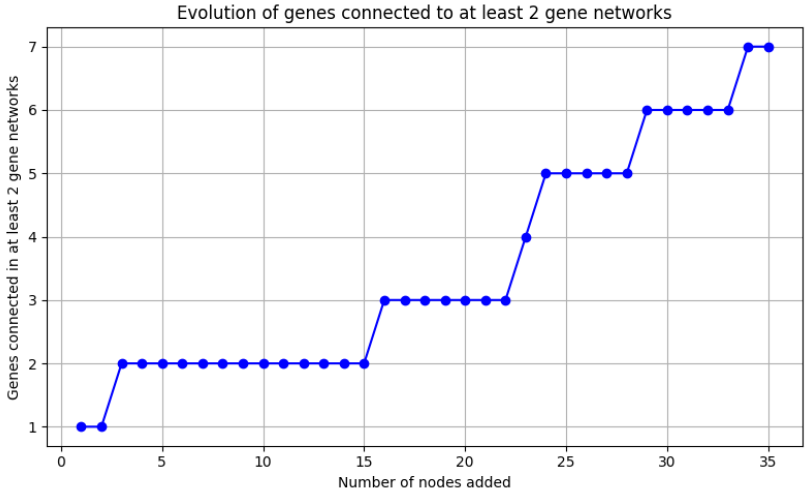

Figure 2: Evolution of genes number according number of added genes

3.1 Optuna

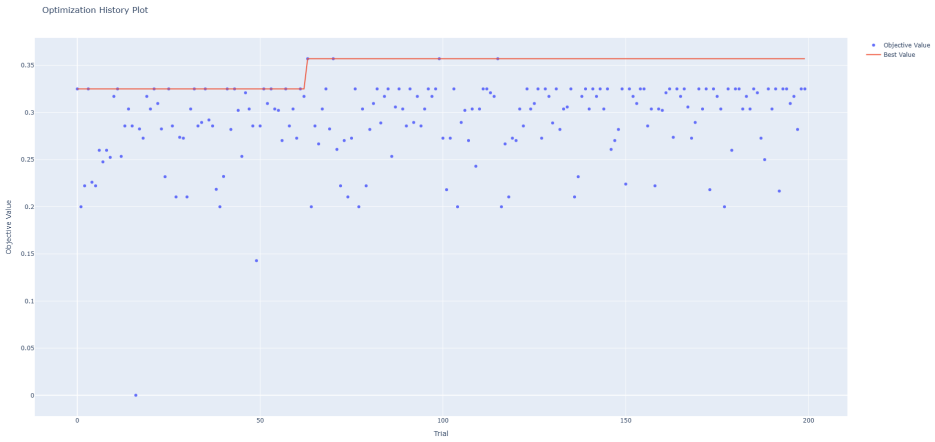

Figure 3: History of values in Optuna trials with F1-score

Initially, network construction parameters were defined in a configuration file and applied manually. To improve reproducibility and avoid arbitrary choices, we integrated **Optuna**[3], an open-source hyperparameter optimization framework [1]. It is particularly well suited for cases where parameters interact in complex ways and manual tuning is inefficient or biased. Optuna uses a define-by-run interface, allowing flexible definition of search spaces, and applies advanced search strategies such as Tree-structured Parzen Estimators (TPE) or random sampling. It also incorporates pruning, a mechanism that stops unpromising trials early to save computation time. In this project, Optuna was employed to optimize key parameters of network construction—such as confidence thresholds and number of added nodes, maximizing the F1-score as an objective. This approach replaces manual trial-and-error with a systematic and reproducible optimization process.

For the first runs, the parameters optimized were:

- confidence threshold (0.70–0.90),
- number of added nodes (5–70),

with the objective of maximizing the F1-score. However, the resulting networks showed only minor differences, and Optuna could not clearly distinguish the best parameter sets (Figure 3). Using the F2-score led to similar results.

3.2 Cross-validation

To better assess robustness, we introduced perturbations of the input gene list. At each run, 20% of the genes were randomly removed, and a new network was reconstructed. Parameters were fixed to a confidence threshold of 0.70 and 50 added nodes. For each gene–gene link, we computed the percentage of times it appeared across runs, defining a robustness score.

The conservation threshold for robustness was then optimized using Optuna. This approach improved the ability to discriminate between parameter settings (Figure 4). The objective function tested included both network density and F1-score. Although effective, it limited Optuna to optimizing only a single parameter, which is a drawback of this early version.

[3] <https://optuna.org/>

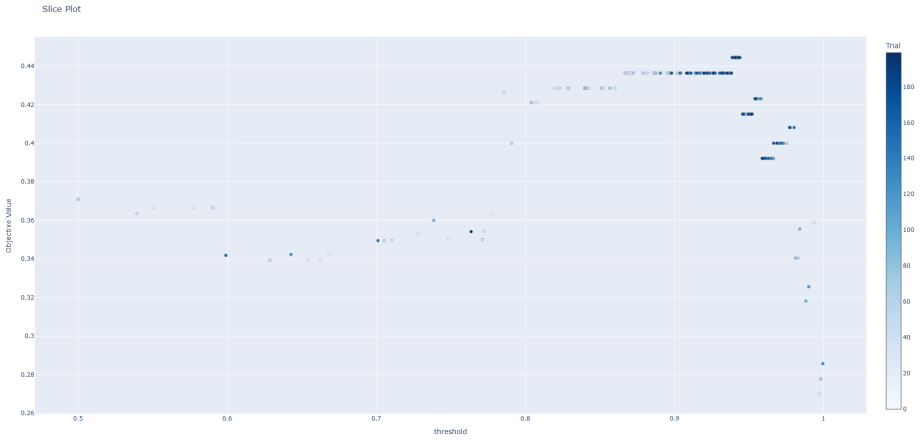

Figure 4: Evolution of density according to the threshold of robustness.

4 Methodology

The final pipeline integrates Optuna-based optimization with robustness evaluation. The steps are as follows:

Step 1 – Input gene list preparation The input is a set of genes of interest (e.g., patient-specific, disease panels). Configuration is provided in a JSON file (Listing 1). In the JSON file, we have a name of input file, the trials number of Optuna, run_number is the number of runs, the name of the output file and the output network. Each run divides the input into 10 folds, removing 20% of genes at a time for cross-validation.

```
{
    "trials_number": 100,
    "run_number": 10,
    "input_file": "./data/gene_list_test.txt",
    "output_file": "my_output.xlsx",
    "output_graph": "network.png"

}
```

Listing 1: JSON example of input file

Step 2 – Network construction For each database (StringDB and FunCoup), interaction partners are retrieved through API queries. Parameters such as confidence threshold, number of added nodes, and expansion strategy are initialized over broad ranges to allow optimization.

Step 3 – Filtering for trusted interactions Only experimentally validated or curated interactions are retained. Predicted or text-mined edges are excluded to minimize noise.

Step 4 – Parameter optimization with Optuna Optuna is searched for the optimal values of:

- Confidence threshold (0.70–0.90)
- Number of added nodes (5–70)
- Robustness threshold (0.30–0.95)

The F1-score is used as the main objective, balancing precision and recall. Trials are pruned if early results are poor, which reduces computational cost.

Step 5 – Robustness assessment To test stability, 10 perturbations are performed per fold, each removing 20% of the input genes. Perturbations are consistent across trials. A gene is considered robust if it appears in a high percentage of resulting networks This percentage is a parameter of Optuna.

Step 6 - Network evaluation To evaluate the quality of the inferred networks, we used a fold-specific F1 score that measures the ability of the method to recover hidden genes. For each fold i, let H_i denote the set of hidden genes, P_i the set of predicted (added) genes, and $TP_i = |H_i \cap P_i|$ the number of hidden genes recovered. Precision and recall are defined as:

$$\text{Precision}_i = \frac{TP_i}{|P_i|}, \qquad \text{Recall}_i = \frac{TP_i}{|H_i|}$$

The F1 score for fold i is then:

$$F1_i = \frac{2 \times \text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}$$

Finally, we aggregate across k folds using the macro-averaged F1:

$$\text{macro-average } F1\text{-score} = \frac{1}{k} \sum_{i=1}^{k} F1_i$$

Step 7 - Gene annotation Added genes are annotated using the UniProt API to provide biological context for interpretation.

5 First results

5.1 Genes list from a patient

For the patient-specific gene list (16 genes), Optuna identified the following optimal parameters:

- Confidence threshold: 0.75
- Number of added nodes: 5
- Robustness threshold: 0.41

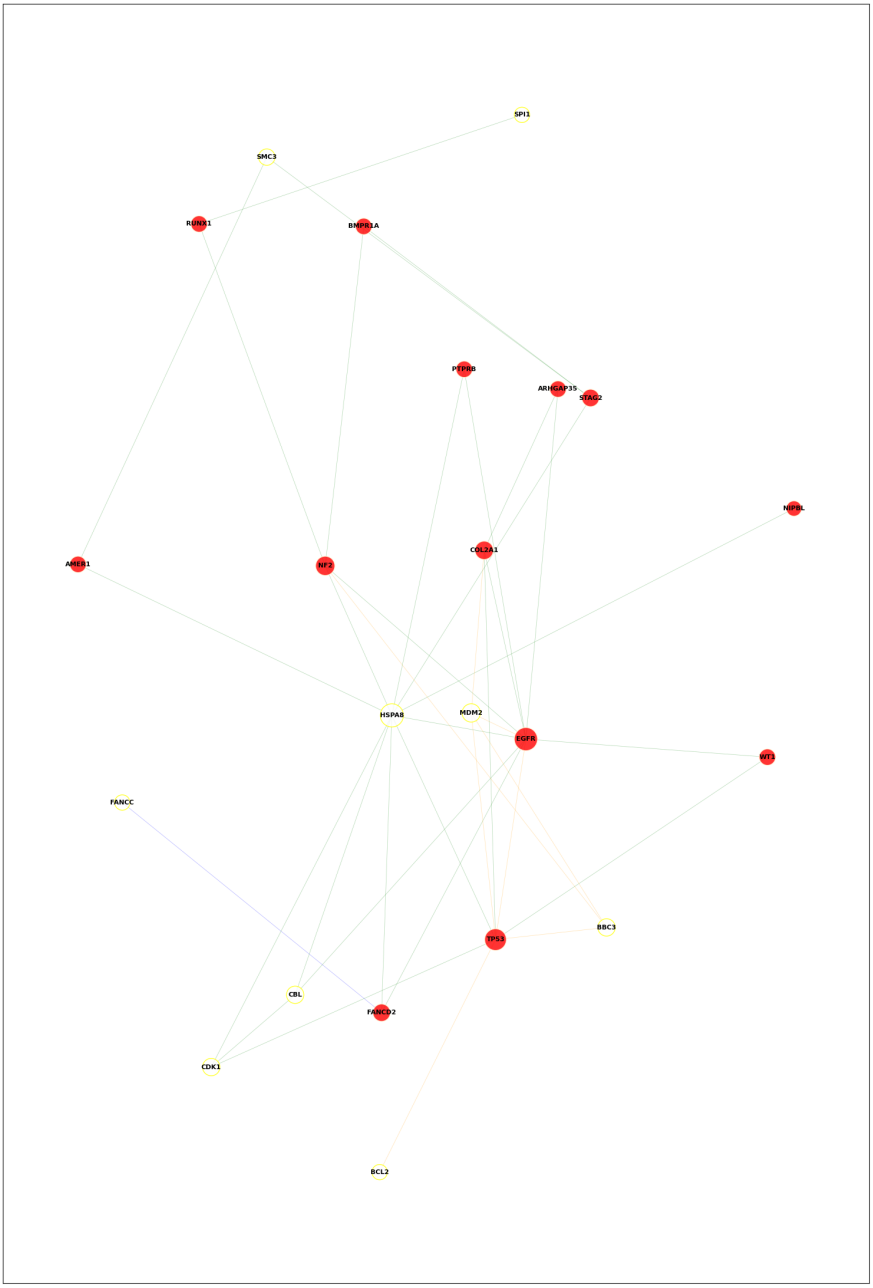

Figure 5: Network after optimization of parameters.

81% of the initial list are in the final network. 9 genes are added: BBC3, BCL2, CBL, CDK1, FANCC, HSPA8, MDM2, SMC3, SPI1. Several of these genes are known to play roles in cancer pathways, suggesting that the pipeline can recover biologically relevant candidates. Several genes are regulators of apoptosis (e.g., BCL2, BBC3). Some are involved in cell cycle regulation and DNA repair (e.g., CDK1, FANCC, MDM2). Others correspond to signaling or transcriptional regulators (e.g., SRC, CBL, SPI1). One gene encodes a molecular chaperone (HSPA8).

The network is presented in the figure 5. The red nodes are the genes from the initial list. The white nodes are the added genes. The color of the links differ depending on their the source: blue if the link is from StringDB, green if the link is from FunCoup group and orange if the link is from FunCoup maxlink.

Figures 6, 7 and 8 represent the results from Optuna. Figure 6 presents the optimization history plot. It displays the progression of the objective values across trials. Each point represents the performance of one trial, while the line tracks the best objective value found up to that trial. Figure 7

shows the importance of different parameters. The parameter importance plot ranks the parameters according to their impact on the objective value. Optuna estimates this importance using statistical methods such as functional ANOVA. Here, we can see that the added nodes number is the parameter with the most sensitive with a score of 0.68. Then, the parameter "robustness threshold" has a importance of 0.26. The confidence score has only a importance of 0.05. Figure 8 visualizes the relationship between a single parameter and the objective value.

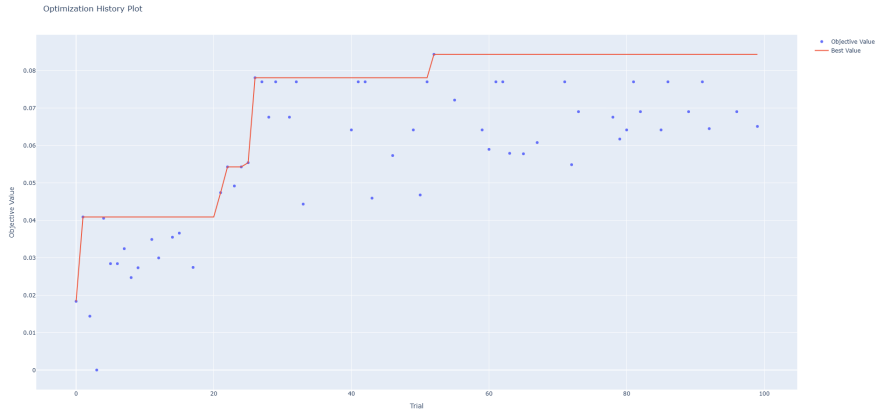

Figure 6: Optimization history plot for a gene list from a patient.

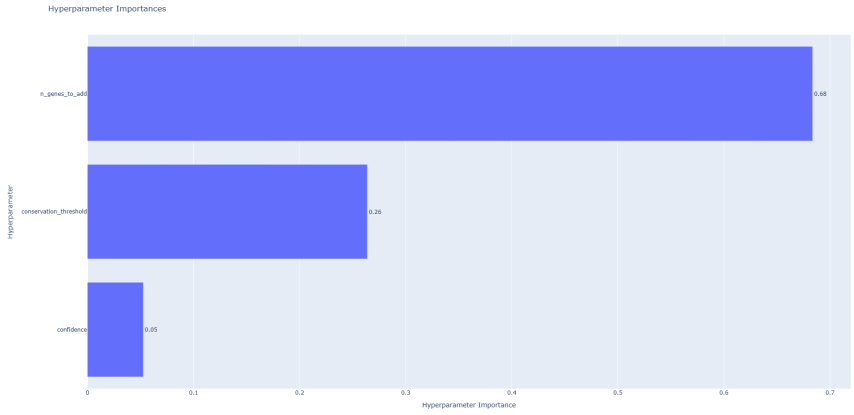

Figure 7: Hyperparamater importance for a gene list from a patient.

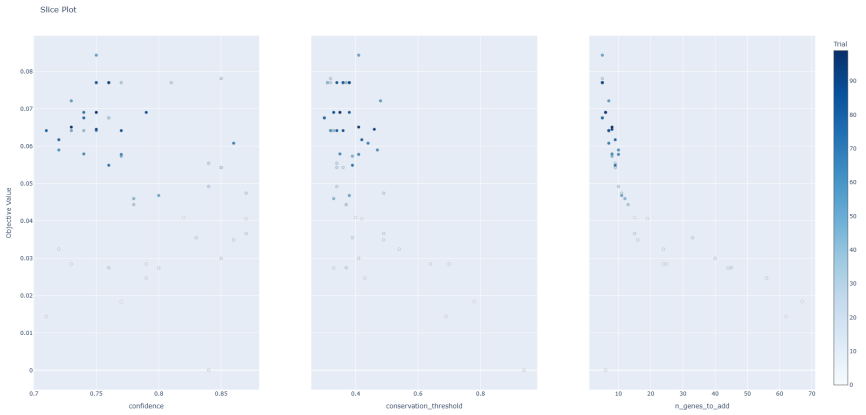

Figure 8: Slice plot for a gene list from a patient.

5.2 Leukemia genes list

For the leukemia gene list (31 genes)[4], Optuna identified the following optimal parameters (10 runs, 100 trials):

- Confidence threshold: 0.73
- Number of added nodes: 7
- Robustness threshold: 0.3

17 genes (54%) from the original list are present in the network. 15 genes are added: AKT1, CBLB, EGFR, HNRNPA0, HRAS, MAP2K2, MAPK3, MYC, PIK3CA, RALGDS, RGL1, SHOC2, SNRPB, SRC, U2AF2. Several belong to kinase and signaling pathways (e.g., AKT1, EGFR, HRAS, MAP2K2, MAPK3, SRC, PIK3CA). One is a transcription factor (MYC).Others are RNA-binding or splicing-related proteins (HNRNPA0, SNRPB, U2AF2). Some are signaling adaptors or regulators (RALGDS, RGL1, SHOC2, CBLB).

Figure 9 shows the network done from the leukemia list.

5.3 Anaemia in silico gene panel

For the anaemia in silico gene panel (210 genes), Optuna identified the following optimal parameters (5 runs, 100 trials) :

- Confidence threshold: 0.89
- Number of added nodes: 46
- Robustness threshold: 0.8

54 genes (26%) from the original list are present in the network. 38 genes are added : CCR5, NOP56, PRC1, RPL12, RPL13, RPL14, RPL18A, RPL22, RPL22L1, RPL23, RPL23A, RPL24, RPL26L1, RPL30, RPL32, RPL34, RPL36, RPL36A, RPL37A, RPL38, RPL39, RPL7, RPL7A, RPL8, RPLP0, RPLP1, RPLP2, RPS12, RPS13, RPS18, RPS2, RPS20, RPS21, RPS23, RPS25, RPS4X, RPS9, VAMP8. The majority of added genes correspond to ribosomal protein genes (RPL and RPS families). One gene encodes a vesicle-associated membrane protein (VAMP8). Others include nucleolar proteins (NOP56) and proteins involved in cytokinesis (PRC1).

Figure 10 shows the network done from the anaemia list.

[4]Source: GSEA-MSIGDB leukemia gene set https://www.gsea-msigdb.org/gsea/msigdb/human/geneset/HP_MYELOID_LEUKEMIA.html

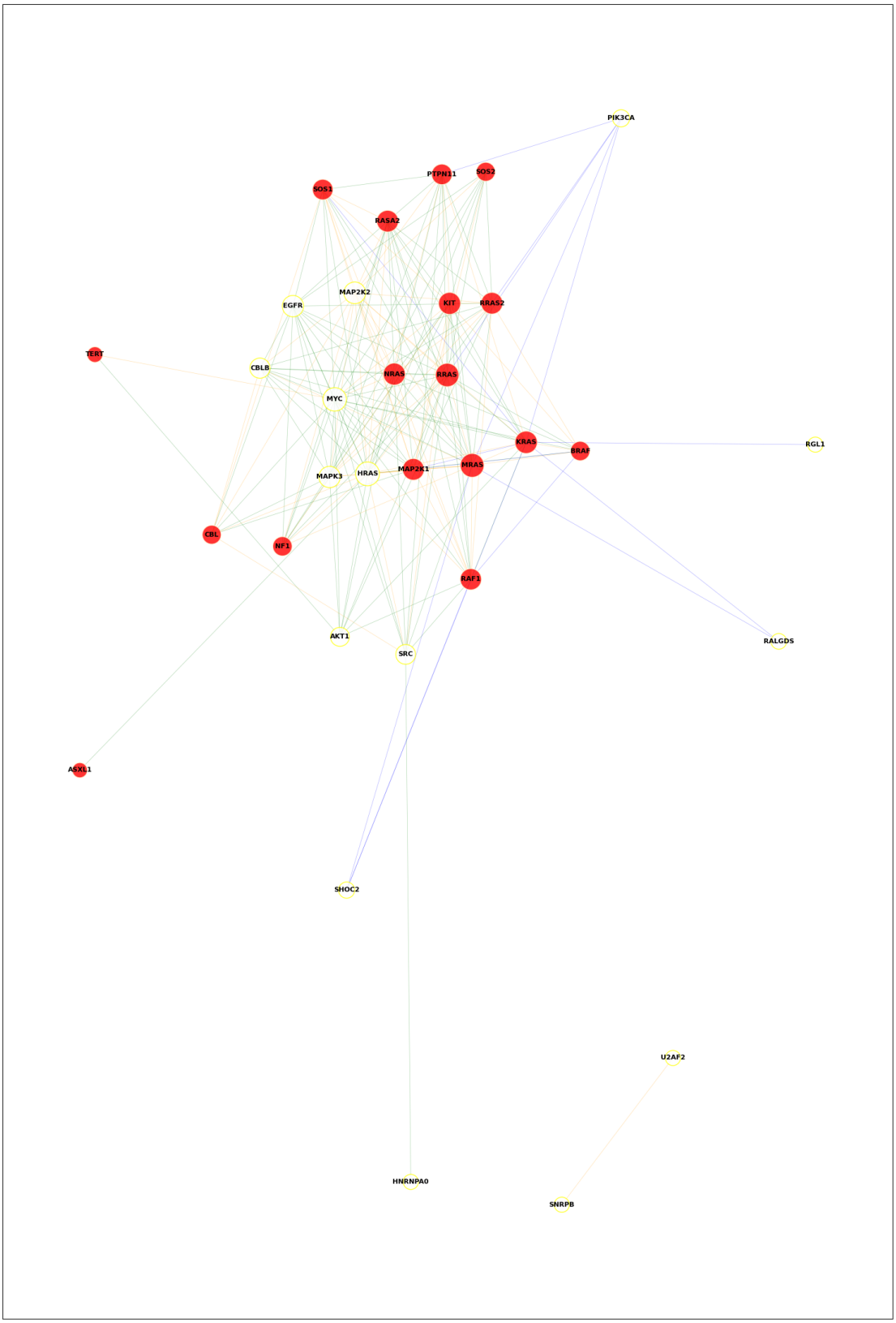

Figure 9: Leukemia Network.

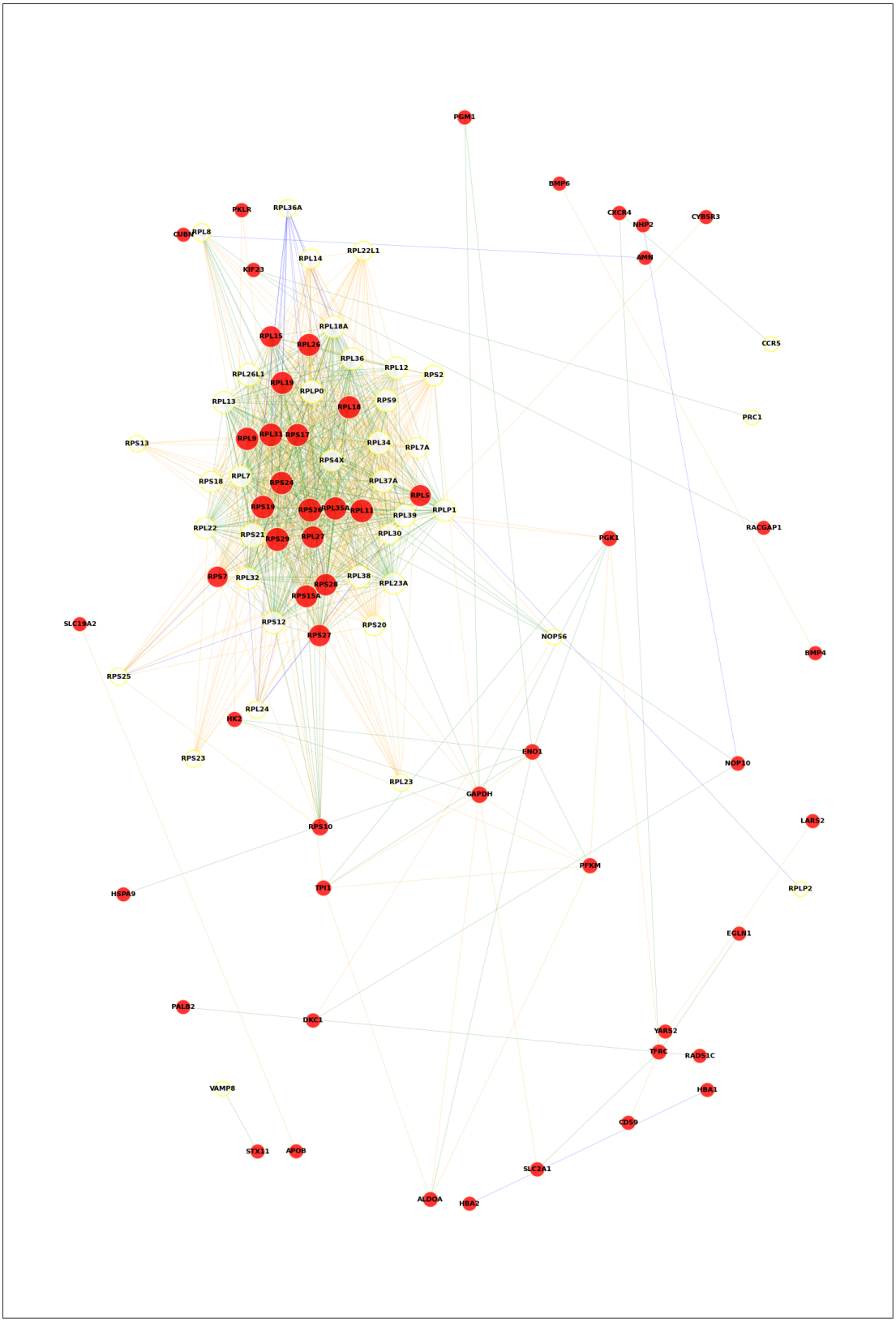

Figure 10: Network from Anaemia in silico gene panel.

5.4 Thrombocytopenia in silico gene panel

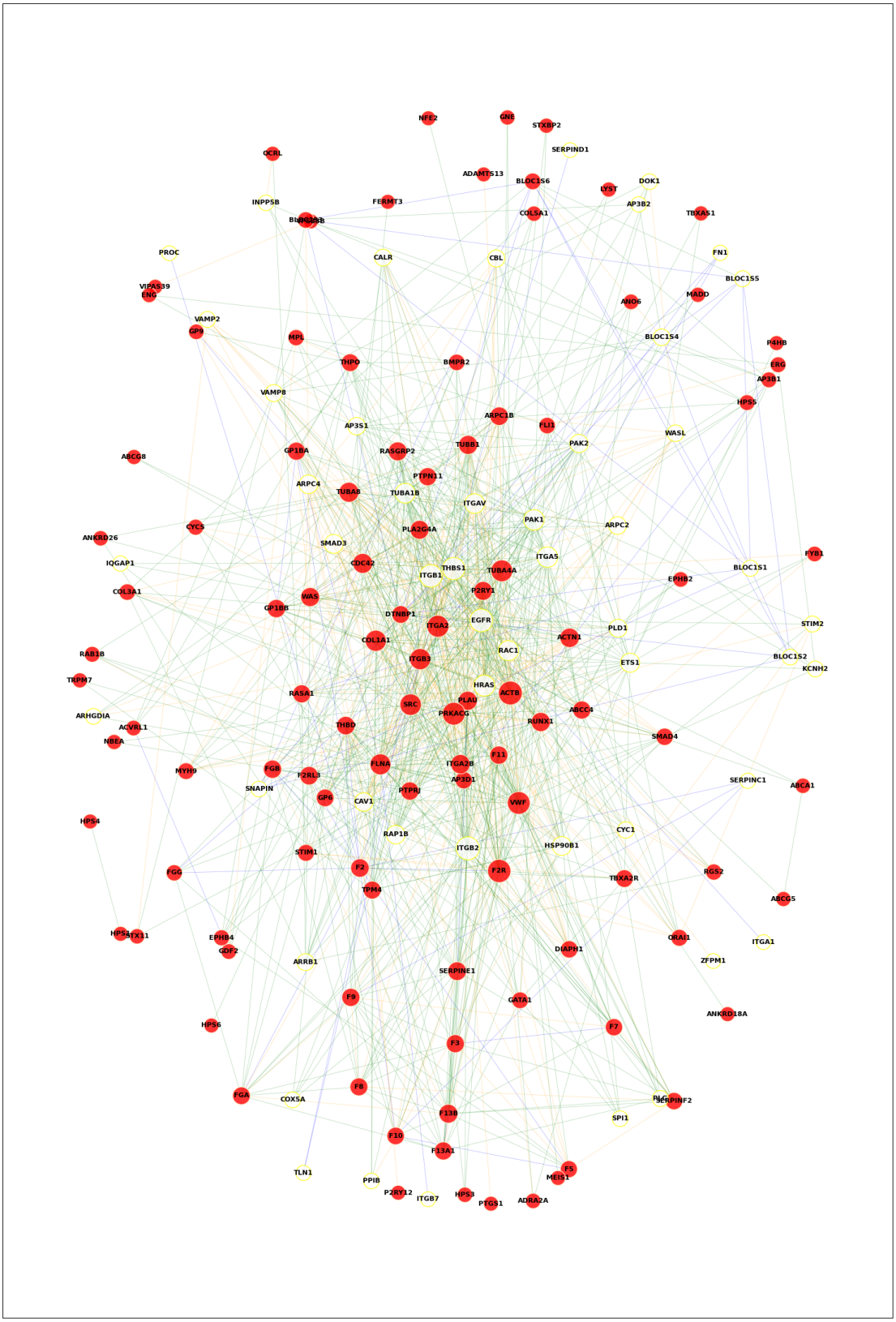

Figure 11: Genes network from thrombocytopenia in silico gene panel.

With 5 runs and 100 trials, Optuna identified the following optimal parameters:

- Confidence threshold: 0.7
- Number of added nodes: 29
- Robustness threshold: 0.47

Figure 11 shows the network done from the thrombocytopenia list.

106 genes (86%) from the original list are present in the network. 51 genes are added to the network: AP3B2, AP3S1, ARHGDIA, ARPC2, ARPC4, ARRB1, BLOC1S1, BLOC1S2, BLOC1S4, BLOC1S5, CALR, CAV1, CBL, COX5A, CYC1, DOK1, EGFR, ETS1, FN1, HRAS, HSP90B1, INPP5B, IQGAP1, ITGA1, ITGA5, ITGAV, ITGB1, ITGB2, ITGB7, KCNH2, PAK1, PAK2, PLD1, PLG, PPIB, PROC, RAC1, RAP1B, SERPINC1, SERPIND1, SMAD3, SNAPIN, SPI1, STIM2, THBS1, TLN1, TUBA1B, VAMP2, VAMP8, WASL, ZFPM1. Several belong to integrins and adhesion molecules (e.g., ITGA1, ITGA5, ITGAV, ITGB1, ITGB2, ITGB7, TLN1, FN1, THBS1). Some are part of vesicle trafficking and endocytosis complexes (e.g., AP3B2, AP3S1, BLOC1S1–5, SNAPIN, VAMP2, VAMP8). Added genes include signaling proteins and small GTPases (e.g., HRAS, RAC1, RAP1B, ARHGDIA, IQGAP1, ARPC2, ARPC4, PAK1, PAK2, WASL). Several encode receptors or kinases (e.g., EGFR, CBL, CAV1, PLD1). Some are involved in transcriptional regulation (e.g., ETS1, SMAD3, SPI1, ZFPM1). Others relate to coagulation and proteases (e.g., PROC, SERPINC1, SERPIND1, PLG). A few are chaperones or mitochondrial proteins (e.g., CALR, HSP90B1, COX5A, CYC1, PPIB, TUBA1B).

6 Conclusion and Future Directions

This work presents a reproducible, parameter-optimized pipeline for expanding disease-relevant gene lists using high-confidence biological interaction data. By combining StringDB and FunCoup, applying strict filtering, and leveraging Optuna for parameter optimization, the pipeline produces networks that still need to be validated.

Future improvements include:

- Validation with independent gene lists and experimental datasets,
- Integrating additional resources such as GeneMANIA [4] via Cytoscape,
- Assessing pipeline determinism and optimizing for computational efficiency,
- Adding others parameters for prioritize specific types of interactions, for instance.

References

[1] Takuya Akiba et al. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019.

[2] Davide Buzzao et al. "The FunCoup Cytoscape App: multi-species network analysis and visualization". In: *Bioinformatics* 41.1 (Dec. 2024), btae739. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btae739. eprint: https://academic.oup.com/bioinformatics/article-pdf/41/1/btae739/61238979/btae739.pdf. URL: https://doi.org/10.1093/bioinformatics/btae739.

[3] Damian Szklarczyk et al. "The STRING database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest". In: *Nucleic Acids Research* 51.D1 (Nov. 2022), pp. D638–D646. ISSN: 0305-1048. DOI: 10.1093/nar/gkac1000. eprint: https://academic.oup.com/nar/article-pdf/51/D1/D638/48440966/gkac1000.pdf. URL: https://doi.org/10.1093/nar/gkac1000.

[4] David Warde-Farley et al. "The GeneMANIA prediction server: biological network integration for gene prioritization and predicting gene function". In: *Nucleic Acids Research* 38.suppl$_2$ (June 2010), W214–W220. ISSN: 0305-1048. DOI: 10.1093/nar/gkq537. eprint: https://academic.oup.com/nar/article-pdf/38/suppl_2/W214/16775007/gkq537.pdf. URL: https://doi.org/10.1093/nar/gkq537.