Find relevant genes from a gene list

Alice Hermann

August 26, 2025

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 Methodology

Two complementary databases to extend the gene list were used:

- StringDB[1] [3]: broad coverage of protein–protein and functional interactions, but includes many predicted links.
- FunCoup[2] [2]: focuses on functional couplings and biological modules, based on diverse evidence.

The pipeline builds a network from the input list. To avoid arbitrary choices, the parameters (confidence thresholds, number of added genes, robustness cut-off) are optimized automatically using Optuna[3], an open-source hyperparameter optimization framework [1].

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

[1] https://string-db.org/
[2] https://funcoup.org/
[3] <https://optuna.org/

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.

3 Conclusion

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.

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.