**Prioritizing Cancer Therapeutic Genes Using BioRank: A Biologically-Informed PageRank Framework**

Duc-Tinh Pham1\*, Huu-Tam Nguyen2, Van-Hai Pham2

1School of Information and Communications Technology, Hanoi University of Industry, 298 Cau Dien Street, Bac Tu Liem District, Hanoi, Vietnam.

2School of Information and Communications Technology, Hanoi University of Science and Technology, Hanoi, Vietnam.

\*Corresponding: Duc-Tinh Pham (tinhpd@haui.edu.vn)

**Table of Contents**

[1 Overview 1](#_Toc202003395)

[2 Installation 1](#_Toc202003396)

[3 Dataset information 1](#_Toc202003397)

[4 Data Preprocessing Functions 2](#_Toc202003398)

[4.1 Compute Ontology Graph 2](#_Toc202003399)

[4.2 Disease-Specific Ontologies 3](#_Toc202003400)

[4.3 Tumor-Control Table Generation (TCGA) 3](#_Toc202003401)

[4.4 DE Gene and Co-expression Network 4](#_Toc202003402)

[5 Running PageRank Algorithms 4](#_Toc202003403)

[5.1 BioRank Algorithm 5](#_Toc202003404)

[6 Output and Export 5](#_Toc202003405)

# Overview

BioRank GUI is a Python-based standalone desktop tool for prioritizing cancer-related genes using biological network analysis. It supports both standard and enhanced PageRank algorithms and provides full preprocessing functionalities.

# Installation

BioRank can be installed and run in two ways:

* **Option 1**

Clone Repository:

|  |
| --- |
| git clone https://github.com/tinhpd/BioRank.git  cd BioRank |

Install Dependencies:

|  |
| --- |
| pip install -r requirements.txt |

Launch GUI:

|  |
| --- |
| python main.py |

* **Option 2 (Runs without installing Python or dependencies)**

Navigate to the /App folder and run: Users can download the full dataset [**here**](App).

|  |
| --- |
| main.exe |

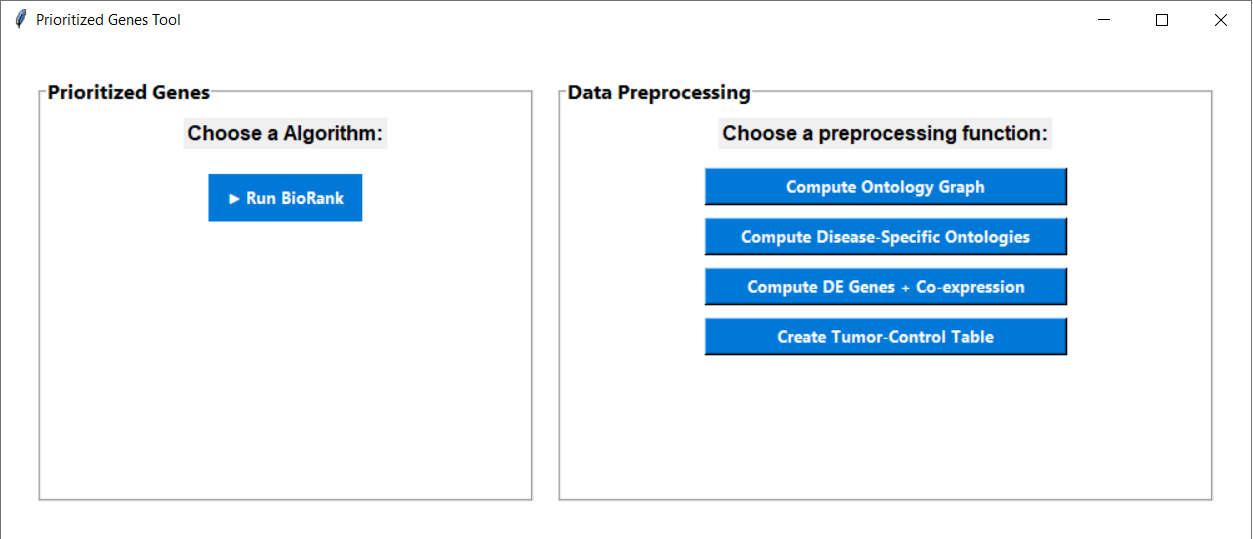
# Dataset information

BioRank requires several biological datasets to perform gene prioritization:

* **Protein-Protein Interaction (PPI) Network**: Used to build the base network structure for PageRank. Source: HIPPIE v2.2.
* **Gene Expression Data**: Tumor vs. control expression profiles from TCGA or RNA-seq/microarray studies.
* **Gene Ontology Annotations**: Including GO (.gaf), KEGG pathways, and Reactome mappings for functional context.
* **UniProt ↔ Ensembl Mapping**: Required for consistent ID resolution across different databases.
* **OncoKB**: Used as ground truth for validation of prioritized genes.

**Note:** Sample data and preprocessing scripts are included in the /dataset/ folder, but it is not enough. Users can download the full dataset [**here**](data_set).

Interface



**Fig. S1**. BioRank GUI

The GUI is divided into two panels:

* **Left Panel:** Run BioRank algorithm.
* **Right Panel:** Data preprocessing tools.

# Data Preprocessing Functions

In the **right panel** of the GUI, users can perform preprocessing tasks using point-and-click dialogs. These scripts prepare input files step-by-step for PageRank execution.

## Compute Ontology Graph

This step builds a bipartite graph linking genes to ontology terms (GO, KEGG, Reactome).

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

**Fig. S2.** Compute Ontology Graph window

Steps:

* Click Compute Ontology Graph.
* Input files in the Data\_set\data\_reprocessing folder:
  + Goa\_human.gaf
  + KEGG\_pathway.txt
  + Reactome.txt
  + Uniprot\_to\_ensembl\_mapping.tsv
  + KEGG\_identifier\_to\_Uniprot.txt
* Click Run.
* Dist\Output\ontology\_output.tsv

## Disease-Specific Ontologies

This step enriches disease-specific annotations from a seed gene list using the ontology graph.

Ảnh có chứa văn bản, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

**Fig. S3.** Compute Disease-Specific Ontologies window

Steps:

* Click Compute Disease-Specific Ontologies.
* Input files:
  + ontology\_output.tsv from [step 4.1](#_Compute_Ontology_Graph)
  + Seed gene (For each disease, select one corresponding file from the seed\_set folder.)
* Click Run.
* Dist\Output\disease\_ontology\_output.txt

## Tumor-Control Table Generation (TCGA)

Creates gene expression tables from raw RNA-seq folders and GDC metadata.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Nội dung do AI tạo ra có thể không chính xác.

**Fig. S4.** Create Tumor-Control table

Steps:

* Click Create Tumor-Control Table.
* Input files (in TCGA folder):
  + GDC sample sheet
  + GDC manifest file
  + RNA-seq data folder
  + Output directory
* Click Run.

## DE Gene and Co-expression Network

Computes differentially expressed (DE) genes and co-expression network from tumor/control samples.

Ảnh có chứa văn bản, phần mềm, Phông chữ, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

**Fig. S5.** Compute DE Genes + Co-expression window

Steps:

* Click Compute DE Genes + Co-expression.
* Input files (in output directory from [step 4.3](#_Tumor-Control_Table_Generation) or out folder in TCGA folder, choose for each disease):
  + Tumor expression table
  + Control expression table
  + Identifier list (file HIPPIE\_node\_list.txt from ppi\_network folder)
* Click Run.
* Output:
  + de\_genes.tsv
  + co\_expression.tsv

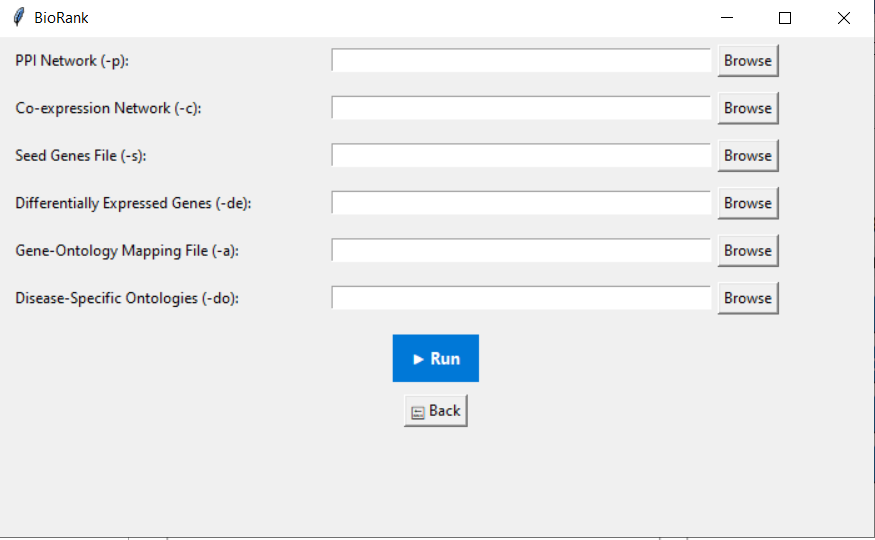
# Running PageRank Algorithms

In the **left panel** of the GUI, users can run PageRank algorithms for cancer gene prioritization. Each option opens a dialog to provide necessary input files and start the computation.

## BioRank Algorithm

Steps:

* Click **Run BioRank.**



**Fig. S6. BioRank** window

Provide the following files:

* + PPI Network (file HIPPIE.tsv in ppi\_network folder)
  + Co-expression network (For each disease, select one corresponding file from the co\_expression\_network folder) from [step 4.4](#_DE_Gene_and)
  + Seed genes list (For each disease, select one corresponding file from the seed\_set folder.)
  + Differentially expressed genes (For each disease, select one corresponding file from the differentially\_expressed\_genes folder) from [step 4.4](#_DE_Gene_and)
  + Gene-Ontology annotation file (file ontology\_network.tsv in ontology\_network folder) from [step 4.1](#_Compute_Ontology_Graph)
  + Disease-specific ontology file (For each disease, select one corresponding file from the disease\_specific\_ontologies folder) from [step 4.2](#_Disease-Specific_Ontologies)
* Click Run to execute.
* Results are saved as LATEST\_RESULT in the output folder.

# Output and Export

* Outputs are saved in /output/.
* View and export results in CSV/TSV format
* GUI provides visual result preview