Unfolding the “black-box” associations between genotype and phenotype is essential for understanding the molecular mechanisms of complex human diseases. Here, we describe the use of GRPath to uncover putative causal paths (pcPaths) from genetic variants to disease phenotypes. GRPath takes multiple omics data and summary statistics as input and identifies pcPaths that link the putative causal region (pcRegion), putative causal variant (pcVariant), putative causal gene (pcGene), noteworthy cell type, and disease phenotype.

Publisher’s note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.
Protocol

Protocol for using GRPath to identify putative gene regulation paths in complex human diseases

Xi Xi, 1,3,* Haochen Li, 2 Lei Wei, 1 and Xuegong Zhang1,2,4,*

1MOE Key Laboratory of Bioinformatics and Bioinformatics Division, BNRIST/Department of Automation, Tsinghua University, Beijing 100084, China
2School of Medicine, Tsinghua University, Beijing 100084, China
3Technical contact
4Lead contact
*Correspondence: xix19@mails.tsinghua.edu.cn (X.X.), zhangxg@tsinghua.edu.cn (X.Z.)
https://doi.org/10.1016/j.xpro.2022.101831

SUMMARY
Unfolding the “black-box” associations between genotype and phenotype is essential for understanding the molecular mechanisms of complex human diseases. Here, we describe the use of GRPath to uncover putative causal paths (pcPaths) from genetic variants to disease phenotypes. GRPath takes multiple omics data and summary statistics as input and identifies pcPaths that link the putative causal region (pcRegion), putative causal variant (pcVariant), putative causal gene (pcGene), noteworthy cell type, and disease phenotype. For complete details on the use and execution of this protocol, please refer to Xi et al. (2022). 1

BEFORE YOU BEGIN

© Timing: <1 day

This section includes the minimal hardware requirements, the installation procedures, as well as the data requirements to be processed by GRPath.

Hardware preparation
Prepare a single- or multi-core computer with a Linux operation system and network connection. A minimum of 8 GB RAM is required.

Environment preparation
We recommend using GRPath in Anaconda environment, where compatible Python and R packages should be installed. Users should follow the step-by-step instructions below to set up their environment. Conda users may skip step 1.

1. Download Anaconda from https://www.anaconda.com/products/distribution according to individual computer specifications.
   
   Note: We installed conda version 4.9.2.

2. Create a new conda environment for GRPath in case of possible conflicts with the already installed packages. Run the following commands in the command line:
Note: In our case, Python 3.10.4 was automatically installed.

3. Install the following Python packages:

```bash
> conda create -n GRPath pip
> conda activate GRPath
> pip install pandas
> pip install numpy
> pip install sklearn
> pip install scipy

Optional: Install Python packages for better display of the final outcomes. These packages will only be used in the last step. Users may skip this step now and decide whether to install them then.

```bash
> pip install graphviz
> pip install networkx
> conda install -c conda-forge pygraphviz
> conda install -c conda-forge matplotlib
```

4. Install R and R packages as follows:

```bash
> conda install r-base
> conda install -c conda-forge r-dplyr
> conda install -c r r-tidy
```

Note: Here, we do not require the latest version of R. Both R 3.0 and R 4.0 should work.

**GRPath toolkit preparation**

5. Download the GRPath toolkit from https://github.com/xixi-cathy/GRPath.
   a. Click the green “Code” button and download the zip folder.

   Note: The folder contains the toy dataset and output files used as example in this protocol, the source code and a ReadMe file.

   b. Extract all files in the zip folder.

**Data collection**

This section describes the required data types, and the origins of our toy data. The databases and download links we mentioned below are also potential data sources for people’s own usage.
Note: Users do not need to download the raw data we described below if directly using our toy data.

6. Decide the disease of interest. Prepare the following multiple omics data relevant to the disease:
   a. Whole-genome sequencing (WGS), RNA-seq data of disease-relevant tissues and disease phenotype (disease/control) from the same individual.

   **Note:** In our example, we focused on cardiovascular disease, and analyzed protected data phs000424.v7.p2 (dbGaP) which can be accessed from GTEx portal (https://gtexportal.org/home/protectedDataAccess). Users may also use their own triple omics data.

   b. Significant eQTLs of disease-relevant tissues.
      i. Download significant cis-eQTLs of different tissues from GTEx data portal (https://storage.googleapis.com/gtex_analysis_v7/single_tissue_eqtl_data/GTEx_Analysis_v7_eQTL.tar.gz).

      **Note:** The recommended q-value threshold is 0.05. We used cis-eQTLs from GTEx V7.

      ii. Extract all files in the tar.gz folder. Select relevant tissues.

   **Note:** In our example, we selected eQTLs from heart left ventricle and left atrial appendage.

   c. GWAS summary statistics of the disease of interest.
      i. Download the full GWAS summary statistics from GWAS Catalog (https://www.ebi.ac.uk/gwas/api/search/downloads/full).
      ii. Select the significant disease-relevant loci according to “DISEASE/TRAIT”, with the standard genome-wide significance p-value threshold of $5 \times 10^{-8}$.

   **Note:** In our example, we selected loci under “cardiovascular disease” and “cardiovascular disease risk factors”.

   d. scRNA-seq data of disease and control samples.
      i. Download the heart failure scRNA-seq raw dataset from GEO: GSE121893 (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE121893), and the control dataset from GEO: GSE109816 (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE109816).
      ii. Process the scRNA-seq data according to “step-by-step method details” step 5.

   **Note:** The scRNA-seq data are not necessarily from the same samples that provide WGS, RNA-seq data and disease phenotype, but should be from the same tissues as the RNA-seq data.

---

**KEY RESOURCES TABLE**

| REAGENT or RESOURCE | SOURCE | IDENTIFIER |
|---------------------|--------|------------|
| Deposited data      | Xi et al.¹ | https://github.com/xixi-cathy/GRPath/tree/main/demo_data |
| Toy data            | Xi et al.¹ | https://github.com/xixi-cathy/GRPath/tree/main/code |
| Software and algorithms | OpenCausal | Li et al.² | https://github.com/liwenran/OpenCausal |
|                     | Seurat  | Stuart et al.³ | https://satijalab.org/seurat/ |
STEP-BY-STEP METHOD DETAILS

Here we describe the step-by-step methods for using GRPath, from calculating chromatin openness scores, to predicting putative causal regions (pcRegions), putative causal variants (pcVariants) within, corresponding putative causal genes (pcGenes) and noteworthy cell types, and finally revealing the putative causal paths (pcPaths) that link pcRegions, pcVariants, pcGenes and noteworthy cell types.

Calculating chromatin openness scores

Timing: <18 h

This section calculates disease-relevant- and tissue-specific openness scores of pre-defined regulatory elements (REs) corresponding to variants in the donor population from WGS, RNA-seq and disease phenotype data. We used the method proposed by Li et al.2

Note: Users may skip this section if directly using our toy data, since we already provided the output files in the data folder.

1. Refer to https://github.com/liwenran/OpenCausal to find out the usage of OpenCausal2 on GTEx data. After this step, the following files should be ready:

   a. variant_openness.txt: A variant-by-donor matrix that denotes the openness scores of each variant predicted from personal genomes (Figure 1A).

Note: The file should be explicitly named as such. The first 8 columns should be the RE and variant information, which are RE-located chromosome, RE starting site, RE ending site, variant chromosome, variant starting site, variant ending site, and the openness score.

Figure 1. Demonstration of the output files after “calculating chromatin openness scores” step

(A) Demonstration of “variant_openness.txt”.

(B) Demonstration of “variant_allele.txt”.

(C) Demonstration of “RE_openness.txt”.

Protocol
variant-located chromosome, variant starting site, variant ending site, variant reference allele and variant alternate allele, respectively. The rest columns are the predicted openness scores of each variant in each donor.

b. **variant_allele.txt**: A binary variant-by-donor matrix that records the genetic variations of each donor (0: reference allele / 1: alternate allele).

*Note*: The variants are the same as in "variant_openness.txt" (Figure 1B). The first 8 columns should be the same as in "variant_openness.txt". The rest columns are the alleles of each variant in each donor.

c. **RE_openness.txt**: An RE-by-sample matrix that denotes the openness scores predicted from reference genomes of all REs that contain the variants in "variant_openness.txt" (Figure 1C).

*Note*: The first 3 columns should be the RE information, which are RE chromosome, RE starting site and RE ending site, respectively. The rest columns are the predicted openness scores of each RE in each sample.

### Predicting pcRegions, pcVariants, and corresponding pcGenes

© **Timing**: <10 min

In this section, we will obtain some preliminary results from GWAS summary statistics, eQTLs and outputs in the previous step. We will identify the pcRegions, pcVariants and pcGenes which composed part of the pcPaths.

2. Prepare the following input files:
   a. **variant_openness.txt**, **variant_allele.txt** and **RE_openness.txt** from the previous step.
   b. **gwas_snps_pos.txt**: location of the GWAS SNPs.

*Note*: The 3 columns should be variant name, chromosome name, and chromosome position with heading (Figure 2A).

c. **phenotype.txt**: donor disease phenotype information with heading.

*Note*: The 4 columns should be donor id, the sample index (starting from zero) in "RE_openness.txt", the donor index (starting from zero) in *"variant_openness.txt"*, and disease label (0: control / 1: disease) (Figure 2B).

d. **variant_pos.txt**: location of variants from GTEx donors in the form of “Chromosome_Position_ReferenceAllele_AltimateAllele_GenomeVersion” with heading. For example, “chr2_136422784_C_T_b37” for “rs3863014” (Figure 2C).

e. **eqtls_200kb.txt**: eQTLs of disease-relevant tissues within the 200 kb region around each GWAS SNP in “gwas_snps_pos.txt”.

*Note*: The 3 columns should be region central SNP (GWAS SNP), variant name and corresponding eGene (Figure 2D).

3. After preparing the data, run the following script to identify the pcRegions, pcVariants within and corresponding pcGenes:
CRITICAL: GRPath also allows users to adjust parameters according to their own needs. The following parameters can be adjusted (As for more detailed meaning of the parameters, users may refer to the STAR Methods section in Xi’s work): 

- **f**: Name of the data directory. Users should store all the input data in a directory under the same parent folder as “code” (default: demo_data).

- **k**: Number of working kernels in multiprocessing. Although more kernels can accelerate the computing process, the value should be decided according to the number of cores of users’ devices (default: 8).

- **i**: Number of rounds of random sampling. The more rounds, the more robust the results users may get. If **i** changes, parameter **p** should also be adjusted accordingly due to effect size changes when calculating the original p-value for each region (default: 10).

- **r**: Number of repeats in all rounds. Users should decide this parameter carefully. Theoretically, a higher value may eliminate randomness, but may also be too strict and lead to too many false negatives (default: 2).

- **p**: Adjusted p-value threshold. This parameter decides whether the investigated region is a pcRegion. It should change along with parameter **i** (default: 1e-3).

For example,

```bash
> # Set working kernels to 4
> bash predict_region_variant_egene.sh -k 4
> # Set the number of rounds to 5, and corresponding adjusted
> # p-value threshold to 0.02
> bash predict_region_variant_egene.sh -i 5 -p 0.02
> # Change data directory to 'data'
> bash predict_region_variant_egene.sh -f data
```

4. Check out the output files (Figure 3), which are:
   a. alter_pathogenic_region_variant_egene.csv: alternate-allele-pathogenic regions, pcVariants within, and their corresponding pcGenes.
   b. ref_pathogenic_region_variant_egene.csv: reference-allele-pathogenic regions, pcVariants within, and their corresponding pcGenes.
   c. A “processing” folder that contains some intermediate results.

Now, some interesting biological discoveries may start to appear.

**Predicting noteworthy cell types and obtaining complete pcPaths**

Timing: <5 min

In this section, users may move one step further, to explore disease mechanisms on a finer cell type level by incorporating aforementioned scRNA-seq data of disease and control samples, and obtain
the complete “pcRegion-pcVariant-pcGene-noteworthy cell type-disease phenotype” form of pcPaths.

5. Process the downloaded scRNA-seq datasets.

   **Note:** The goal of this step is to minimize the influences of sequencing technologies and data noises, as well as annotate the cells. Standard practices include quality control, log-normalization, batch effect correction, data integration, dimensionality reduction and clustering etc. depending on the properties (e.g., sequencing technology, sources) of the data. We recommend using R and Seurat package\(^3\) in this step. Users may refer to online tutorials https://satijalab.org/seurat/articles/pbmc3k_tutorial.html and https://satijalab.org/seurat/articles/integration_introduction.html for code and more detailed information.

   **Note:** Users may skip this step if directly running our toy data, since we already provided the processed data in the folder.

6. Prepare two files after processing the downloaded scRNA-seq datasets:
   a. **single_cell_data.txt**: a gene-by-cell gene expression matrix.

      **Note:** Values in the matrix are normalized gene expression levels ready for machine learning models (Figure 4A).

   b. **single_cell_label.txt**: disease state (0: control / 1: disease) and cell type of each cell in the gene-by-cell matrix (Figure 4B).
7. After preparing the data, run the following script to link the previously identified pcGenes with noteworthy cell types, and obtain the complete pcPaths:

```bash
> bash predict_regulation_path.sh
```

⚠ CRITICAL: Since we used down-sampling technique in this step to avoid the bias caused by unbalanced sample sizes in different classes, we need to repeat the classification process for multiple rounds and take the average. Here, GRPath also allows users to decide the number of rounds they prefer by parameter `r`:

- `-f`: Name of the data directory. Same as the parameter in step 3 (default: demo_data).
- `-r`: Number of rounds of down-sampling. Results will better converge with a higher value, but will be more time-consuming (default: 3).

For example, users may run the following command:

```bash
> # Set 10 rounds of down-sampling
> bash predict_regulation_path.sh -r 10
> # Change data directory to 'data'
> bash predict_regulation_path.sh -f data
```

8. Check out the output files (Figure 3), which are:
   a. `regulation_path.csv`: pcPaths.
   b. `cell_type_auc_mean.csv` and `cell_type_auc_std.csv` in the “processing” folder as intermediate results.

Now, users may freely explore the pcPaths and see if there are any interesting discoveries.

**Visualizing results**

⏰ Timing: <3 min

To better visualize the results, users may further draw the paths in hierarchical graphs.
**Optional:** Make sure that the packages described in the “Optional” part in “environment preparation” were all installed properly. Then, run the following command:

```python
> python plot_regulation_paths.py
```

The graphs will be saved in “regulation_paths-alternate_pathogenic.pdf” for pcPaths originating from alternate-allele-pathogenic regions (Figure 5A) and “regulation_paths-reference_pathogenic.pdf” for pcPaths originating from reference-allele-pathogenic regions (Figure 5B), respectively. When zooming in, users can see what each node stands for exactly (Figure 5C).

**EXPECTED OUTCOMES**

GRPath links genetic variants, gene expression, cell type and individual-level disease state together, and can uncover putative gene regulation paths of a specific disease. It bridges the gap between statistical associations and biological mechanisms, and can be used to study heredity and micro-to-macro mechanisms of complex diseases with strong genetic effects.

As outcomes, GRPath reveals pcRegions, pcVariants within and corresponding pcGenes from matched triple omics (WGS, RNA-seq and phenotype) data after the second step, pcPaths in the form of “pcRegion-pcVariant-pcGene-noteworthy cell type-disease phenotype” with cell type information added after the third major step, and more intuitive visualization of the pcPaths in the last step.

The computational outcomes may serve as a reference to guide scientists to further explore and validate the molecular mechanisms of their interested diseases.

**LIMITATIONS**

Currently, the biggest limitation rises from data requirements. On the one hand, GRPath must take WGS, RNA-seq and disease phenotype data from the same individual as input, which is strict, especially when WGS data are usually protected. On the other hand, finding appropriate GWAS summary statistics and scRNA-seq data that match with the triple omics data (in terms of disease and tissues) is difficult. We may improve our method in the future to make the data requirements less strict. Moreover, incorporating large-scale scRNA-seq atlas should also be helpful for uncovering putative gene regulation paths in a broader scope, which we may explore in future works as well.

**TROUBLESHOOTING**

**Problem 1**

The reference genome versions are not matched among different sources of data in “data collection”, for example, GRCh37 for WGS and GRCh38 for GWAS summary statistics.

**Potential solution**

Users should guarantee that all the input files of GRPath are in a unified reference genome version. We recommend using the web-based software biomaRt to convert between different reference genome versions.

**Problem 2**

Unable to find perfectly matched scRNA-seq data with the triple omics data in terms of disease phenotype in “before you begin” step 7d.

**Potential solution**

Although we recommend using perfectly matched data, it is acceptable if the disease phenotype in scRNA-seq data belongs to a subtype of that in triple omics data. For example, the disease...
phenotype in triple omics data is “cardiovascular disease”, but “heart failure” in scRNA-seq data. In that case, users should manually check the results more carefully to avoid false positives.

Problem 3
GRPath fails to process the input files in “step-by-step method details” steps 3 and 7.

Potential solution
Check the formats of input files carefully, including the headings and indices. The formats should be exactly the same as toy data demonstrate.

Problem 4
The outcomes in “step-by-step method details” step 4 are different in each run.

Potential solution
This is normal, since we do random sampling in this step. Users may adjust the parameters i, r, and p according to their data and needs as suggested in “step-by-step method details” step 3. Users may also repeat this step for multiple times and take the intersection as the final result.

Problem 5
Unable to solve the bug report when running the scripts.

Potential solution
Users may create an issue in our GitHub repository (https://github.com/xixi-cathy/GRPath) with detailed descriptions, and we will try to help.

RESOURCE AVAILABILITY

Lead contact
Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Xuegong Zhang (zhangxg@tsinghua.edu.cn).

Materials availability
This study did not generate new unique reagents.
Data and code availability
The toy data and code generated during this study are available at: https://github.com/xixi-cathy/GRPPath or https://doi.org/10.5281/zenodo.7146018.

ACKNOWLEDGMENTS
The work was supported by National Key Research and Development Program of China (2021YFF1200901) and NSFC grants (61721003, 62250005).

AUTHOR CONTRIBUTIONS
X.Z. conceptualized and designed the project. X.X. designed the protocol and conducted the example experiments. H.L. examined the example experiments. X.X. and L.W. wrote the manuscript. All authors read and approved the manuscript.

DECLARATION OF INTERESTS
The authors declare no competing interests.

REFERENCES
1. Xi, X., Li, H., Chen, S., Lv, T., Ma, T., Jiang, R., Zhang, P., Wong, W.H., and Zhang, X. (2022). Unfolding the genotype-to-phenotype black box of cardiovascular diseases through cross-scale modeling. Science 25, 104-90. https://doi.org/10.1016/j.sci.2022.104790.

2. Li, W., Duren, Z., Jiang, R., and Wong, W.H. (2020). A method for scoring the cell type-specific impacts of noncoding variants in personal genomes. Proc. Natl. Acad. Sci. USA 117, 21364-21372. https://doi.org/10.1073/pnas.1922703117.

3. Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck, W.M., Hao, Y., Stoeckius, M., Smibert, P., and Satija, R. (2019). Comprehensive integration of single-cell data. Cell 177, 1888-1902. https://doi.org/10.1016/j.cell.2019.05.031.

4. Skinnider, M.A., Squair, J.W., Kathe, C., Anderson, M.A., Gautier, M., Matson, K.J.E., Milano, M., Hutson, T.H., Barraud, Q., Phillips, A.A., et al. (2021). Cell type prioritization in single-cell data. Nat. Biotechnol. 39, 30-34. https://doi.org/10.1038/s41587-020-0605-1.

5. Durinck, S., Spellman, P.T., Birney, E., and Huber, W. (2009). Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package biomaRt. Nat. Protoc. 4, 1184–1191. https://doi.org/10.1038/nprot.2009.97.