Developing a Comprehensive Python Library for Mendelian Randomization Analysis

Introduction

Mendelian Randomization (MR) is an increasingly popular epidemiological technique that leverages genetic variants as instrumental variables to infer causal relationships between modifiable exposures and health outcomes. This method has gained traction due to its ability to mitigate confounding and reverse causation, common pitfalls in traditional observational studies. The surge in genome-wide association studies (GWAS) has provided a wealth of genetic data, making MR a powerful tool for causal inference in epidemiology.

Several Python libraries and tools have been developed to facilitate MR analysis, such as MeRP. These tools automate key steps in developing and analyzing genetic instruments, but they also have limitations. For instance, MeRP focuses on high-throughput analysis but may not adequately address issues like pleiotropy and heterogeneity in genetic instruments.

Current Python MR Libraries and Their Limitations

1. **MeRP**:
    - **Methodology**: MeRP automates the creation of instrumental variable trait files from publicly available data, filters potential confounding associations, and performs MR-score analysis using summary data.
    - **Limitations**: MeRP's score-based approach may not fully account for heterogeneity in genetic mechanisms. Additionally, it may lack comprehensive methods for sensitivity analysis and pleiotropy detection.

2. **TwoSampleMR**:
    - **Methodology**: This R package, often used in Python via RPy2, provides tools for two-sample MR analysis, including various sensitivity analyses and pleiotropy tests.
    - **Limitations**: The reliance on R can be a barrier for Python-centric workflows. Moreover, the package may not be optimized for high-throughput analysis or integration with other Python-based bioinformatics tools.

Proposed Plan for a New MR Library

To address these limitations, we propose developing a new Python library for Mendelian Randomization analysis. This library will integrate robust statistical methods, user-friendly interfaces, and comprehensive documentation to facilitate MR studies.

Mathematical Foundation

The library will implement the following core mathematical concepts:

- **Instrumental Variable (IV) Analysis**: Using genetic variants as instruments to estimate causal effects.
- **Two-Sample MR**: Leveraging summary statistics from separate GWAS for exposure and outcome.
- **Pleiotropy Detection**: Implementing methods like MR-Egger regression and MR-PRESSO to identify and correct for pleiotropic effects.
- **Sensitivity Analyses**: Including weighted median, weighted mode, and leave-one-out analyses to assess the robustness of causal estimates.

Key Features

1. **Data Integration**:
    - Seamless integration with popular bioinformatics databases like PhenoScanner and the GWAS Catalog.
    - Support for importing and exporting data in various formats (e.g., CSV, JSON, HDF5).

2. **Instrument Selection and Validation**:
    - Automated selection of genetic instruments based on user-defined criteria.
    - Validation of instruments using metrics like F-statistics and linkage disequilibrium (LD) pruning.
3. **Causal Inference Methods**:
    - Implementation of standard MR methods (e.g., IVW, MR-Egger, weighted median).
    - Advanced methods for handling pleiotropy and heterogeneity (e.g., contamination mixture method).
4. **Visualization and Reporting**:
    - Comprehensive visualization tools for MR results, including forest plots and scatter plots.
    - Automated generation of detailed reports summarizing the analysis and findings.
5. **User-Friendly Interface**:
    - Intuitive API design for ease of use.
    - Extensive documentation and tutorials to guide users through the MR analysis process.

By incorporating these features, the proposed library aims to provide a comprehensive, user-friendly, and robust tool for conducting Mendelian Randomization studies in Python. This will not only enhance the reproducibility and reliability of MR analyses but also broaden the accessibility of these methods to the wider scientific community.

Table of Contents

Overview of Mendelian Randomization and Current Python Libraries

Mendelian Randomization: Concept and Methodology

Mendelian Randomization (MR) is an epidemiological technique that leverages genetic variants as instrumental variables to infer causal relationships between modifiable exposures and health outcomes. This method is particularly valuable in overcoming confounding and reverse causation issues inherent in traditional observational studies. The core principle of MR is based on Mendel's laws of inheritance, which ensure that genetic variants are randomly assorted during meiosis, thus mimicking the randomization process of a controlled trial.

Key Assumptions in MR

1. **Relevance**: The genetic variants (instruments) must be strongly associated with the exposure.
2. **Independence**: The instruments should not be associated with any confounders of the exposure-outcome relationship.
3. **Exclusion Restriction**: The instruments affect the outcome only through the exposure, not via any other pathway.

Current Python Libraries for Mendelian Randomization

Several Python libraries have been developed to facilitate MR analysis. Below, we review some of the prominent ones, their methodologies, and limitations.

MeRP: Mendelian Randomization Pipeline

Methodology:

- **Data Acquisition**: MeRP utilizes publicly available Genome-Wide Association Study (GWAS) data to generate instrumental variable trait files.
- **Filtering**: The library applies algorithms to filter out potential confounding associations and linkage disequilibrium (LD) among SNPs.
- **Causal Inference**: MeRP implements a score-based analytical approach to estimate causal effects using summary data.

Limitations:

- **Heterogeneity**: The score-based approach may not adequately address heterogeneity in genetic mechanisms affecting the trait and outcome.
- **Complexity**: The pipeline can be complex to set up and requires significant computational resources.
- **Flexibility**: Limited flexibility in incorporating new methods or custom analyses.

For more details, visit the MeRP documentation.

MR-Base

Methodology:

- **Data Integration**: MR-Base integrates data from multiple GWAS consortia, providing a comprehensive platform for MR analysis.
- **Statistical Methods**: It supports various MR methods, including inverse-variance weighted (IVW), MR-Egger, and weighted median approaches.
- **Visualization**: The platform offers tools for visualizing MR results, such as forest plots and scatter plots.

Limitations:

- **Data Dependency**: Relies heavily on the availability and quality of GWAS summary statistics.
- **User Interface**: The web-based interface may not be as flexible for advanced users who prefer command-line tools.

For more information, visit the MR-Base platform.

Proposed Plan for a New MR Python Library

To address the limitations of existing libraries and provide a more comprehensive tool for MR analysis, we propose the development of a new Python library with the following features:

Core Features

1. **Data Handling**:

    - **Integration with GWAS Databases**: Seamless integration with major GWAS databases like PhenoScanner and the GWAS Catalog.
    - **Data Harmonization**: Automated harmonization of exposure and outcome datasets to ensure consistency.

2. **Statistical Methods**:

    - **Basic MR Methods**: Implementation of standard MR methods such as IVW, MR-Egger, and weighted median.
    - **Advanced Methods**: Inclusion of robust methods like MR-PRESSO, contamination mixture method, and multivariable MR (MVMR).

3. **Sensitivity Analyses**:

    - **Heterogeneity Tests**: Cochran's Q test for detecting heterogeneity.
    - **Pleiotropy Tests**: MR-Egger intercept test and MR-PRESSO for identifying pleiotropic effects.
    - **Leave-One-Out Analysis**: Assessing the influence of individual SNPs on the overall causal estimate.

4. **Visualization Tools**:

    - **Forest Plots**: For visualizing causal estimates and confidence intervals.
    - **Scatter Plots**: For examining the relationship between genetic instruments and outcomes.
    - **Heatmaps**: For visualizing pleiotropy and heterogeneity across instruments.

5. **User Interface**:

    - **Command-Line Interface (CLI)**: For advanced users who prefer scripting and automation.
    - **Graphical User Interface (GUI)**: For users who prefer a more interactive approach.

6. **Documentation and Tutorials**:

    - **Comprehensive Documentation**: Detailed API documentation with examples.
    - **Tutorials and Vignettes**: Step-by-step guides for common MR analyses.

Mathematical Foundation

The mathematical foundation of the proposed library will be based on the following principles:

Instrumental Variable (IV) Estimation

For a single genetic variant (Z) used as an instrument for exposure (X) and outcome (Y):

[\text{IV Estimate} = \frac{\text{Cov}(Z, Y)}{\text{Cov}(Z, X)}]

Inverse-Variance Weighted (IVW) Method

For multiple genetic variants, the IVW method combines the individual estimates:

[\hat{\beta}{IVW} = \frac{\sum{i} wi \hat{\beta}i}{\sum{i} wi}]

where (wi = \frac{1}{\text{Var}(\hat{\beta}i)}) is the inverse variance of the individual estimates.

MR-Egger Regression

MR-Egger regression accounts for pleiotropy by including an intercept term:

[Y = \alpha + \beta X + \epsilon]

where (\alpha) represents the pleiotropic effect.

Implementation Plan

1. **Project Setup**:
    - **Repository Initialization**: Set up a GitHub repository with a clear structure.
    - **Package Configuration**: Use tools like `setuptools` and `poetry` for package management.
2. **Core Modules**:
    - **Data Module**: Functions for data acquisition, harmonization, and preprocessing.
    - **Analysis Module**: Implementation of MR methods and sensitivity analyses.
    - **Visualization Module**: Tools for generating plots and visualizations.
3. **Testing and Validation**:
    - **Unit Tests**: Comprehensive unit tests for all functions.

    - **Integration Tests**: Tests to ensure modules work together seamlessly.
    - **Benchmarking**: Performance benchmarking against existing libraries.

4. **Documentation and Release**:

    - **API Documentation**: Using tools like `Sphinx` for generating documentation.
    - **User Guides**: Detailed guides and tutorials.
    - **Release Management**: Versioning and release notes.

Example Code Snippet

Below is an example of how the new library might be used to perform an IVW analysis:

```python import mrpy as mr

# Load exposure and outcome data

exposuredata = mr.loadgwasdata('exposuregwas.csv') outcomedata = mr.loadgwasdata('outcomegwas.csv')

# Harmonize data

harmonizeddata = mr.harmonizedata(exposuredata, outcomedata)

# Perform IVW analysis

ivwresult = mr.ivw(harmonizeddata)

# Visualize results

mr.plotforest(ivwresult) ```
```

Conclusion

The proposed Python library aims to provide a comprehensive, flexible, and user-friendly tool for conducting Mendelian Randomization analyses. By addressing the limitations of existing libraries and incorporating advanced methods and visualization tools, this library will facilitate robust causal inference in epidemiological research. For more information on MR methodologies and tools, visit the MR-Base platform and the MeRP documentation.

Mathematical Foundations and Methodologies in Mendelian Randomization

Instrumental Variable (IV) Estimation

Instrumental Variable (IV) estimation is a cornerstone of Mendelian Randomization (MR) analysis. The IV approach leverages genetic variants as instruments to infer causal relationships between exposures and outcomes. The key assumptions for a valid IV are:

1. **Relevance**: The genetic variant must be associated with the exposure.
2. **Independence**: The genetic variant must be independent of any confounders.
3. **Exclusion Restriction**: The genetic variant affects the outcome only through the exposure.

Mathematically, the causal effect ((\beta)) can be estimated using the Wald ratio estimator for a single IV:

[\beta = \frac{\text{Cov}(G, Y)}{\text{Cov}(G, X)}]

where (G) is the genetic variant, (Y) is the outcome, and (X) is the exposure.

Inverse-Variance Weighted (IVW) Method

The IVW method extends the Wald ratio to multiple genetic variants. It assumes that each genetic variant provides an independent estimate of the causal effect. The IVW estimator is given by:

[\hat{\beta}{IVW} = \frac{\sum{i} wi \hat{\beta}i}{\sum{i} wi}]

where (\hat{\beta}i) is the causal estimate from the (i)-th variant, and (wi) is the inverse of the variance of (\hat{\beta}_i). This method is efficient and widely used due to its simplicity and robustness under the assumption of no pleiotropy.

MR-Egger Regression

MR-Egger regression is designed to account for pleiotropy, where genetic variants affect the outcome through pathways other than the exposure. The MR-Egger model is:

[Y = \alpha + \beta X + \epsilon]

where (\alpha) represents the intercept, which can be interpreted as an indicator of directional pleiotropy. The slope ((\beta)) provides an unbiased estimate of the causal effect even in the presence of pleiotropy.

Current Python Libraries for Mendelian Randomization

MeRP: Mendelian Randomization Pipeline

MeRP is a high-throughput pipeline for MR analysis. It automates key steps such as generating instrumental variable trait files, filtering for confounding associations, and performing causal inference using summary data. However, MeRP has limitations, including:

- **Heterogeneity**: The score-based approach may not account for heterogeneity in genetic mechanisms.
- **Limited Methods**: It primarily uses a single method for causal inference, which may not be robust across all scenarios.

MR-Base

MR-Base is a comprehensive platform for systematic causal inference using genetic data. It supports various MR methods, including IVW, MR-Egger, and weighted median. Despite its robustness, MR-Base has some limitations:

- **Complexity**: The platform can be complex for new users.
- **Data Integration**: Integrating new datasets can be cumbersome.

Proposed Plan for a New MR Python Library

To address the limitations of existing libraries, the new MR Python library should include the following features:

Core Features

1. **Multiple MR Methods**: Implement IVW, MR-Egger, weighted median, and contamination mixture methods.
2. **Data Harmonization**: Functions to harmonize exposure and outcome data, ensuring consistency in allele coding.
3. **Sensitivity Analyses**: Tools for heterogeneity tests, pleiotropy detection, and leave-one-out analyses.
4. **Visualization**: Plotting functions for causal estimates, funnel plots, and forest plots.
5. **User-Friendly Interface**: Simplified API for easy integration and use.

Mathematical Foundation

Weighted Median Estimator

The weighted median estimator provides a robust causal estimate by taking the median of the weighted distribution of individual variant estimates. It is less sensitive to outliers and pleiotropy. The weighted median ((\hat{\beta}_{WM})) is calculated as:

[\hat{\beta}{WM} = \text{median}(\hat{\beta}i, w_i)]

where (\hat{\beta}i) are the individual causal estimates, and (wi) are the weights.

Contamination Mixture Method

The contamination mixture method identifies groups of genetic variants with similar causal estimates, representing distinct mechanisms. It performs MR robustly in the presence of invalid IVs. The method involves:

1. **Clustering**: Grouping genetic variants based on their causal estimates.
2. **Estimation**: Performing MR within each cluster and combining results.

Implementation Plan

The implementation plan for the new MR library involves the following steps:

1. **Data Handling**: Functions to load and preprocess GWAS summary statistics.
2. **IV Selection**: Tools to select and validate genetic instruments.
3. **Causal Estimation**: Implementing IVW, MR-Egger, weighted median, and contamination mixture methods.
4. **Sensitivity Analyses**: Functions for heterogeneity tests, MR-PRESSO, and leave-one-out analyses.
5. **Visualization**: Plotting functions for results interpretation.

Example Code Snippet

Below is an example code snippet for performing IVW analysis using the proposed library:

```python import mrpy as mr

# Load exposure and outcome data

exposuredata = mr.loaddata('exposure.csv') outcomedata = mr.loaddata('outcome.csv')

# Harmonize data

harmonizeddata = mr.harmonizedata(exposuredata, outcomedata)

# Perform IVW analysis

ivwresult = mr.ivw(harmonizeddata)
```

Visualize results

mr.plotforest(ivwresult) ```

This snippet demonstrates the simplicity and functionality of the proposed library, making it accessible for researchers.

Conclusion

The proposed MR Python library aims to provide a comprehensive, user-friendly tool for Mendelian Randomization analysis. By incorporating multiple MR methods, robust sensitivity analyses, and intuitive visualization tools, it addresses the limitations of existing libraries and facilitates high-quality causal inference in genetic epidemiology.

Proposed Features and Implementation Plan for the New Python MR Library

Current Python MR Libraries: Methodology and Limitations

MeRP: Mendelian Randomization Pipeline

Methodology: MeRP (Mendelian Randomization Pipeline) is designed to facilitate rapid causal inference analysis by automating key steps in developing and analyzing genetic instruments from publicly available data. The pipeline consists of three main components:

1. **Data Acquisition:** MeRP uses the National Human Genome Research Institute (NHGRI) catalog to generate instrumental variable trait files (IVFs).
2. **Filtering:** The pipeline applies a filtering algorithm to minimize associations with potential confounding factors and linkage disequilibrium (LD).
3. **Causal Inference:** MeRP performs causal effect estimation using a score-based approach, which aggregates multiple SNPs into a genetic score to estimate the causal effect on the outcome.

Limitations:

- **Heterogeneity:** The score-based approach may not account for heterogeneity in genetic mechanisms or pathways affecting the trait and outcome differently (Yin & Voight, 2015).
- **Limited Robustness:** The pipeline may not be robust against pleiotropy, where genetic variants influence multiple traits.
- **Complexity:** The tool requires significant computational resources and expertise to handle large datasets and perform comprehensive analyses.

MR-Base

Methodology: MR-Base is a platform that supports systematic causal inference across the human phenome using billions of genetic associations. It provides tools for:

1. **Data Integration:** Aggregating GWAS summary statistics from various sources.
2. **Harmonization:** Aligning exposure and outcome data to ensure consistency.
3. **Analysis:** Performing various MR methods, including Inverse-Variance Weighted (IVW), MR-Egger, and Weighted Median.

Limitations:

- **Data Quality:** The accuracy of MR-Base analyses depends on the quality and completeness of the input GWAS data (Hemani et al., 2018).
- **Pleiotropy Detection:** While MR-Base includes methods to detect pleiotropy, it may not fully account for all pleiotropic effects, potentially biasing causal estimates.
- **User Interface:** The platform's complexity can be a barrier for users without advanced bioinformatics skills.

Proposed Plan for a New MR Python Library

Core Features

1. **Data Acquisition and Integration:**

    - **GWAS Data Import:** Functions to import GWAS summary statistics from various sources, including the NHGRI catalog and PhenoScanner.
    - **Data Harmonization:** Tools to align exposure and outcome datasets, ensuring consistency in allele coding and effect direction.

2. **Instrument Selection and Filtering:**

    - **LD Clumping:** Algorithms to filter SNPs based on linkage disequilibrium, ensuring independence of genetic instruments.
    - **Confounder Filtering:** Methods to exclude SNPs associated with potential confounding traits.

3. **Causal Inference Methods:**

    - **Inverse-Variance Weighted (IVW):** Standard method for estimating causal effects using multiple genetic instruments.
    - **MR-Egger Regression:** Method to detect and adjust for pleiotropy.
    - **Weighted Median Estimator:** Robust method that provides consistent estimates even when up to 50% of the instruments are invalid.
    - **Contamination Mixture Method:** Advanced technique to handle invalid instruments and pleiotropy.

4. **Sensitivity Analyses:**

    - **Leave-One-Out Analysis:** Assessing the influence of individual SNPs on the overall causal estimate.
    - **MR-PRESSO:** Detecting and correcting for horizontal pleiotropy.
    - **MR-Steiger Directionality Test:** Inferring the direction of causality between exposure and outcome.

5. **Visualization Tools:**

    - **Forest Plots:** Visualizing causal estimates and confidence intervals for individual SNPs.
    - **Scatter Plots:** Displaying the relationship between genetic associations with exposure and outcome.
    - **Funnel Plots:** Assessing the presence of bias in MR estimates.

6. **User Interface:**

    - **Command-Line Interface (CLI):** Simplified interface for running MR analyses from the terminal.
    - **Graphical User Interface (GUI):** User-friendly interface for non-expert users to perform MR analyses and visualize results.

Mathematical Foundation

Instrumental Variable (IV) Estimation

The core principle of Mendelian Randomization (MR) is the use of genetic variants as instrumental variables (IVs) to estimate the causal effect of an exposure on an outcome. The IV estimation relies on three key assumptions:

1. **Relevance:** The genetic variant is associated with the exposure.
2. **Independence:** The genetic variant is independent of confounders.
3. **Exclusion Restriction:** The genetic variant affects the outcome only through the exposure.

Inverse-Variance Weighted (IVW) Method

The IVW method combines the effect estimates from multiple genetic instruments into a single causal estimate. The formula for the IVW estimator is:

[\hat{\beta}{IVW} = \frac{\sum{i=1}^{K} \frac{\beta{Yi} \beta{Xi}}{\sigma{Yi}^2}}{\sum{i=1}^{K} \frac{\beta{Xi}^2}{\sigma{Yi}^2}}]

where (\beta{Yi}) and (\beta{Xi}) are the effect estimates of the genetic variant on the outcome and exposure, respectively, and (\sigma{Yi}) is the standard error of (\beta{Yi}).

MR-Egger Regression

MR-Egger regression extends the IVW method by allowing for an intercept term, which can account for pleiotropy. The regression model is:

[\beta{Yi} = \alpha + \beta{MR} \beta{Xi} + \epsiloni]

where (\alpha) represents the pleiotropic effect, and (\beta_{MR}) is the causal estimate.

Weighted Median Estimator

The weighted median estimator provides a robust causal estimate by taking the median of the weighted distribution of individual SNP estimates. This method is less sensitive to outliers and invalid instruments.

Contamination Mixture Method

The contamination mixture method identifies groups of genetic variants with similar causal estimates, representing distinct mechanisms by which the exposure influences the outcome. It performs MR robustly in the presence of invalid IVs by minimizing the mean squared error across various scenarios.

Implementation Plan

1. **Library Structure:**

    - **Data Module:** Functions for data import, harmonization, and filtering.
    - **Analysis Module:** Implementation of IVW, MR-Egger, weighted median, and contamination mixture methods.
    - **Sensitivity Module:** Tools for leave-one-out analysis, MR-PRESSO, and MR-Steiger tests.
    - **Visualization Module:** Functions for generating forest plots, scatter plots, and funnel plots.
    - **Interface Module:** CLI and GUI for user interaction.

2. **Development Workflow:**

    - **Phase 1:** Develop core functions for data handling and basic MR methods (IVW, MR-Egger).

  - **Phase 2:** Implement advanced methods (weighted median, contamination mixture) and sensitivity analyses.
  - **Phase 3:** Create visualization tools and user interfaces (CLI, GUI).
  - **Phase 4:** Test and validate the library using real-world datasets.
  - **Phase 5:** Documentation and user guides.

3. **Example Code Snippet:**

```python import pandas as pd from mr_library import DataHandler, MRAnalysis, Visualization

# Load exposure and outcome data

exposuredata = DataHandler.loadgwasdata('exposuregwas.csv') outcomedata = DataHandler.loadgwasdata('outcomegwas.csv')

# Harmonize data

harmonizeddata = DataHandler.harmonizedata(exposuredata, outcomedata)

# Perform IVW analysis

ivwresult = MRAnalysis.ivw(harmonizeddata)

# Perform MR-Egger analysis

eggerresult = MRAnalysis.mregger(harmonized_data)

# Visualize results

Visualization.forestplot(ivwresult, title='IVW Analysis') Visualization.scatterplot(harmonizeddata, title='MR-Egger Analysis') ```
```

This implementation plan ensures a comprehensive and user-friendly MR library that addresses the limitations of existing tools and provides robust causal inference methods for researchers.

References

- https://augmentationlab.medium.com/writing-a-simulation-i-conways-game-of-life-86202a003040
- https://link.springer.com/chapter/10.1007/978-3-031-48465-0_5
- https://noweyr.github.io/
- https://www.biorxiv.org/content/10.1101/2022.08.30.505937v1
- https://www.cell.com/patterns/fulltext/S2666-3899(24)00096-5
- https://stackoverflow.com/questions/128478/should-import-statements-always-be-at-the-top-of-a-module
- https://conwaylife.com/forums/index.php
- https://conwaylife.com/?rle=9b2o$9bobo$4b2o6bo7b2o$2obo2bo2bo2bo7b2o$2o2b2o6bo$9bobo$9b2o!&name=(p30) queen bee shuttle
- https://www.sciencedirect.com/science/article/pii/S0927025620305401
- https://www.nature.com/articles/s41467-019-14156-4
- https://www.thelancet.com/journals/landia/article/PIIS2213-8587(23)00348-0/fulltext
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7614635/
- https://learnpython.com/blog/define-function-python/
- https://andrewslabucsf.github.io/MR-tutorial/scripts/intro.html
- https://www.medrxiv.org/content/10.1101/2024.05.23.24307776v1.full.pdf
- https://www.sciencedirect.com/science/article/pii/S135964622300430X
- https://amymariemason.github.io/MR/
- https://pubmed.ncbi.nlm.nih.gov/25388149/
- https://biofish.medium.com/pyconway-6dc2479de409
- https://github.com/anpu9/UCB-CSC1C-proj1-Game-of-Life-
- https://www.sciencedirect.com/science/article/abs/pii/S0925838809019021
- https://longnow.org/ideas/conways-game-of-life-and-three-millennia-of-human-history/
- https://pubmed.ncbi.nlm.nih.gov/38706291/

- https://www.freecodecamp.org/news/functions-in-python-a-beginners-guide/
- https://www.sciencedirect.com/science/article/pii/S0925838822008374
- https://stackoverflow.com/questions/139180/how-to-list-all-functions-in-a-module
- https://pypi.org/project/genal-python/
- https://python.land/introduction-to-python/functions
- https://www.annualreviews.org/content/journals/10.1146/annurev-genom-090314-050016
- https://onlinelibrary.wiley.com/doi/10.1002/brb3.3551
- https://wellcomeopenresearch.org/articles/4-186
- https://www.nature.com/articles/srep13802
- https://towardsdev.com/python-functions-with-code-examples-91f3e73df655
- https://www.sciencegate.app/keyword/693600
- https://academic.oup.com/ije/article/51/6/2031/6671803
- https://en.wikipedia.org/wiki/Conway'sGameof_Life
- https://www.sciencedirect.com/science/article/pii/S0921509321010285
- https://www.mdpi.com/2075-4701/13/7/1174
- https://www.sciencedirect.com/science/article/pii/S1359645414008295
- https://www.nature.com/articles/s43856-024-00530-x
- https://bmjopen.bmj.com/content/bmjopen/13/9/e072087.full.pdf?with-ds=yes
- https://www.discovermagazine.com/the-sciences/mathematicians-prove-the-omniperiodicity-of-conways-game-of-life?ref=gorillasun.de
- https://github.com/matthijsknigge/mendelianrandomization
- https://conwaylife.com/wiki/Conway'sGameof_Life
- https://www.nature.com/articles/s41598-024-59326-7
- https://www.sciencedirect.com/science/article/pii/S0925838809019021
- https://www.youtube.com/watch?v=aabO1dwT2Fs
- https://github.com/peteryin21/py-merp
- https://kyushu-u.elsevierpure.com/en/publications/the-role-of-trace-element-segregation-in-the-eutectic-modificatio
- https://www.sciencedirect.com/science/article/pii/S1471531702000093
- https://www.publichealth.columbia.edu/academics/non-degree-special-programs/professional-non-degree-programs/skills-health-research-professionals-sharp-training/mendelian-randomization

- https://onlinelibrary.wiley.com/doi/pdf/10.1002/gepi.22544
- https://www.quantamagazine.org/maths-game-of-life-reveals-long-sought-repeating-patterns-20240118/
- https://www.nature.com/articles/s41598-024-62178-w
- https://www.nature.com/articles/s43586-021-00092-5
- https://medium.com/@alper.bulbul1/exploring-the-causal-pathways-with-python-implementing-mendelian-randomization-methods-in-research-adca766bcf2f
- https://www.nytimes.com/2020/12/28/science/math-conway-game-of-life.html
- https://www.cambridge.org/core/journals/psychological-medicine/article/mendelian-randomization-causal-inference-leveraging-genetic-data/333B418DF7B88B53D58032EA316D0834
- https://github.com/NCBI-Hackathons/MR_BACOn
- https://conwaylife.com/forums/viewtopic.php?t=6489
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8725623/
- https://www.nature.com/articles/s41598-019-41919-2
- https://www.frontiersin.org/articles/10.3389/fcimb.2016.00057/full
- https://www.sciencedirect.com/science/article/pii/S1359645416301963
- https://academic.oup.com/bioinformatics/article/31/6/957/215107
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7384151/
- https://github.com/OpenOmics/mr-seek
- https://github.com/topics/mendelianrandomization
- https://conwaylife.com/
- https://academic.oup.com/book/40819/chapter/348789403
- https://www.researchgate.net/publication/268231351MeRPahigh-throughputpipelineforMendelianRandomizationAnalysis
- https://link.springer.com/article/10.1007/s11665-024-09511-6
- http://py-merp.github.io/intro.html
- https://www.medrxiv.org/content/10.1101/2024.05.23.24307776v1
- https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1009238
- https://www.codewithc.com/an-in-depth-guide-to-python-functions-creation-usage-and-best-practices/
- https://math.stackexchange.com/questions/1536288/what-are-the-practical-uses-of-game-of-life-or-langtons-ant
- https://onlinelibrary.wiley.com/doi/full/10.1002/jbm4.10675

- https://www.nature.com/articles/s41398-024-02950-8
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11161388/
- https://pubmed.ncbi.nlm.nih.gov/38263619/