

# PEDIGREE ENGINE

17/November 2019

**Version** 1.0

**Date** 2019-11-17

**Title** Pedigree Engine

**Depends** R(>=3.4.0), kinship2,shiny

**License** GPL (>= 2)

**URL** <https://github.com/cloudyaaron/6112project>

## **Author**

Ziqi Yang,

Hujia Wang,

Jaiden Choy

## **Description**

Pedigree charts are used in Medicine to record a patient's family history. They are particularly useful to document genetic diseases that can be inherited from one generation to the next. The example below shows a pedigree chart of a (fictitious) family where multiple individuals are affected by breast cancer:

In most clinical settings, pedigree charts are still drawn by hand on paper, which limits their utility. Pedigree drawing software tools exist but are complicated to use and/or record their data in proprietary formats that are unsuitable for medical research purposes.

In this project, we aim to develop a pedigree drawing engine that can create pedigree charts as the example above from simple textual descriptions. The concept is similar to Graphviz (<https://www.graphviz.org/>) and mermaid (<https://mermaidjs.github.io/>), two projects that allow creating graphs and diagrams using a simple text syntax.

## Table of Contents

Instruction and install.....	3
Access through shinnyapps.io .....	3
Clone from our GitHub repo.....	3
Form of Input.....	4
Attributes input: .....	4
Relation input: .....	4
Notes .....	5
Use Function ONLY .....	5
print_ped(df) .....	5
producePED(filepath) .....	6
Producegraph(df,d,position,arg,legendsize) .....	6

## Instruction and install

There are currently two ways to use the pedigree engine.

### Access through shinyapps.io

URL: [https://6112project.shinyapps.io/Pedigree\\_project/](https://6112project.shinyapps.io/Pedigree_project/)

Limits to this method:

Due to our free account limitations. This can only be access for 25 hours.

### Clone from our GitHub repo.

URL: <https://github.com/cloudyaaron/6112project>

This can be run locally or deploy in LAN.

#### **Run locally:**

1. Clone our repo from GitHub.
2. Install all the required packages.
3. Open shiny.R file by RStudio.
4. Click Run App on top right corner of the script window.
5. Open a web browser or built in browser and start using it.

#### **Deploy in LAN:**

Deploy in Lan is possible however it is not integrated in our code. Use provided link to modify the code and deploy.

URL: <https://shiny.rstudio.com/articles/deployment-local.html>

## Notes

After creating new file, new file needs to be loaded to start creating pedigree graph.

If the web page accidentally closed and Pedigree Engine was run locally.

The pedigree graph and ped file may be able to find in ./out/ folder.

Python version was no longer in maintenance or develop. However, python code was in ice bucket folder for who really need them.

## Form of Input

Sequence of input is important.

Different sequence of input may result in different graph.

Attributes input:

1. A gender\_is male
2. A,B,C gender\_is female
3. A name\_is "John Doe"
4. A DOB\_is "13 Oct 1970"
5. A deceased\_is true
6. A,B,C deceased\_is true
7. A affected\_is diseaseA
8. A affected\_is diseaseA,B,C
9. A,B,C affected\_is diseaseX,Y,Z
10. A ad\_is "addition text "

Relation input:

1. A father\_of B
2. A father\_of B,C,D
3. A mother\_of B
4. A mother\_of B,C,D
5. A partner\_of B (A will be B's children parents)
6. A brother\_of B (Only can be use if B has parents)
7. A sister\_of B (Only can be use if B has parents)
8. A m\_twin B (monozygotic twin)
9. A d\_twin B (dizygotic twin)
10. A unkonwn\_twin B

## Notes

**Affected attribute must be assign at last or after creating every node otherwise an error will rise.**

All relation input will create new nodes if node is not in the graph.

For example, A father\_of B,C,D will create 4 nodes named A,B,C,D.

But the graph won't show until B,C,D has a mother.

Assign B,C,D a mother Z can be done in:

Z mother\_of B,C,D

Or Z partner\_of A

**More sample inputs can be found in sample inputs folder.**

## Use Function ONLY

PedigreeEngine.R file can be found in ./src/ folder. This file can be copy and individual used if needed. Here are functions can be called.

### print\_ped(df)

this function takes a data frame as input and print the data frame into .ped format at ./output/ouput.ped

**df -> An data frame(a data frame generates by *producePED* function)**

This function **return nothing**.

## producePED(filepath)

This function takes the text input file's path and returns the information into a data frame.

**Filepath** -> Text input file's path. (cannot be relative path)

This function **returns df**.

**df** -> The data frame that contains relationship and attributes. Initially 14 columns max 18 columns.

**14 columns are:** "ped", "id", "father", "mother", "sex", "affected", "deceased", "twin", "node", "name", "dob", "partner", "sg", "ad"

**4 columns additions are for disease.**

## Producegraph(df,d,position,arg,legendsize)

This function takes data frame and a set of arguments as input and returns the pedigree graph.

**df** -> the data frame that is produced by producePED function.

**d** -> distance between nodes, prefer from 0.01 to 2.

**position** -> where the legend locates on the graph. Only accept from "topright", "topleft", "bottomright", "bottomleft"

**arg** -> a list may contain (name, id, dob, affect, ad) or empty. If list contains any of it the graph will show extra content.

**legendsize** -> The legends' size on the graph. Prefer from 0.001 to 0.05.