Genetic Analysis and Investigating Pleiotropic Architecture with 'GGPA' Package

Dongjun Chung [1], Hang J. Kim [2], and Hongyu Zhao [3,4,5,6]
[1]Department of Public Health Sciences, Medical University of South Carolina, Charleston, SC, USA.
[2] Department of Mathematical Sciences, University of Cincinnati, Cincinnati, OH, USA.
[3] Department of Biostatistics, Yale School of Public Health, New Haven, CT, USA.
[4] Program in Computational Biology and Bioinformatics, Yale University, New Haven, CT, USA.
[5] Department of Genetics, Yale School of Medicine, New Haven, CT, USA.
[6] VA Cooperative Studies Program Coordinating Center, West Haven, CT, USA.

November 22, 2024

Contents

1 Overview

This vignette provides an introduction to the genetic analysis using the 'GGPA' package. R package 'GGPA' implements graph-GPA, a flexible statistical framework for the joint analysis of multiple genome-wide association studies (GWAS) using a hidden Markov random field architecture. We encourage questions or requests regarding 'GGPA' package to be posted on our Google group for the GPA Suite https://groups.google.com/d/forum/gpa-user-group. Users can find the most up-to-date versions of 'GGPA' package in our GitHub webpage (http://dongjunchung.github.io/GGPA/).

The package can be loaded with the command:

```
R> library("GGPA")
```

This vignette is organized as follows. Section 2 describes the overall graph-GPA analysis workflow [1] including model fitting (Section 2.1), association mapping (Section 2.2), and visualization of an estimate phenotype graph (Section 3). Section 4 illustrates how a prior disease graph can be queried and downloaded from the graph-GPA companion webiste and incorporated to the graph-GPA analysis workflow [2].

2 Workflow

[Note]

All the results below are based on the 200 burn-in and 200 main MCMC iterations for quick testing and building of the R package. These results are provided here only for the illustration purpose and should not be considered as real results. We recommend users to use sufficient number of burn-in and main MCMC iterations, as we use 10,000 burn-in and 40,000 main MCMC iterations for all the results in our manuscript [1].

In this vignette, we use the simulated GWAS data for $20,000$ SNPs and seven phenotypes for the illustration purpose. Users can find a p-value matrix of size $20,000 \times 7$ in the '`simulation$pmat`' object.

```
R> data(simulation)
R> dim(simulation$pmat)

[1] 20000     7

R> head(simulation$pmat)

           [,1]       [,2]       [,3]      [,4]      [,5]       [,6]       [,7]
[1,] 0.05201560 0.69394601 0.80095254 0.8935946 0.4062871 0.71502089 0.85759656
[2,] 0.42270504 0.86932265 0.83619632 0.5979045 0.7480898 0.04931404 0.62476746
[3,] 0.28201444 0.26960700 0.04520746 0.5359965 0.9764476 0.33313946 0.09141433
[4,] 0.87792229 0.75501240 0.95592348 0.1425499 0.7948491 0.36315009 0.67920778
[5,] 0.59408852 0.36952615 0.73312469 0.7972099 0.3852618 0.47646133 0.47012336
[6,] 0.01012334 0.06122195 0.87669912 0.9065982 0.5867958 0.96146317 0.65355255
```

In this simulation studies, we assume the three strongly correlated phenotypes (n1, n2, n3), two weakly correlated phenotypes (n4, n5), and two independent phenotypes (n6, n7), as illustrated in Figure 1. Parameters used to generate simulation data can be found in the list object '`simulation`'. More details about simulation data generation procedure can be found in our manuscript [1].

```
R> adjmat <- simulation$true_G
R> diag(adjmat) <- 0
R> ggnet2( adjmat, label=TRUE, size=15 )
```

2.1 Fitting the graph-GPA Model

We are now ready to fit a graph-GPA model using the GWAS p-value data described above (`simulation$pmat`). R package `GGPA` provides flexible analysis framework and automatically adjusts its model structure based on the provided data. Users can fit the graph-GPA model with the command:

```
R> set.seed(12345)
R> fit <- GGPA( simulation$pmat )
```

The following command prints out a summary of graph-GPA model fit, including data summary, proportion of SNPs associated with each phenotype, parameter estimates, and their standard errors.

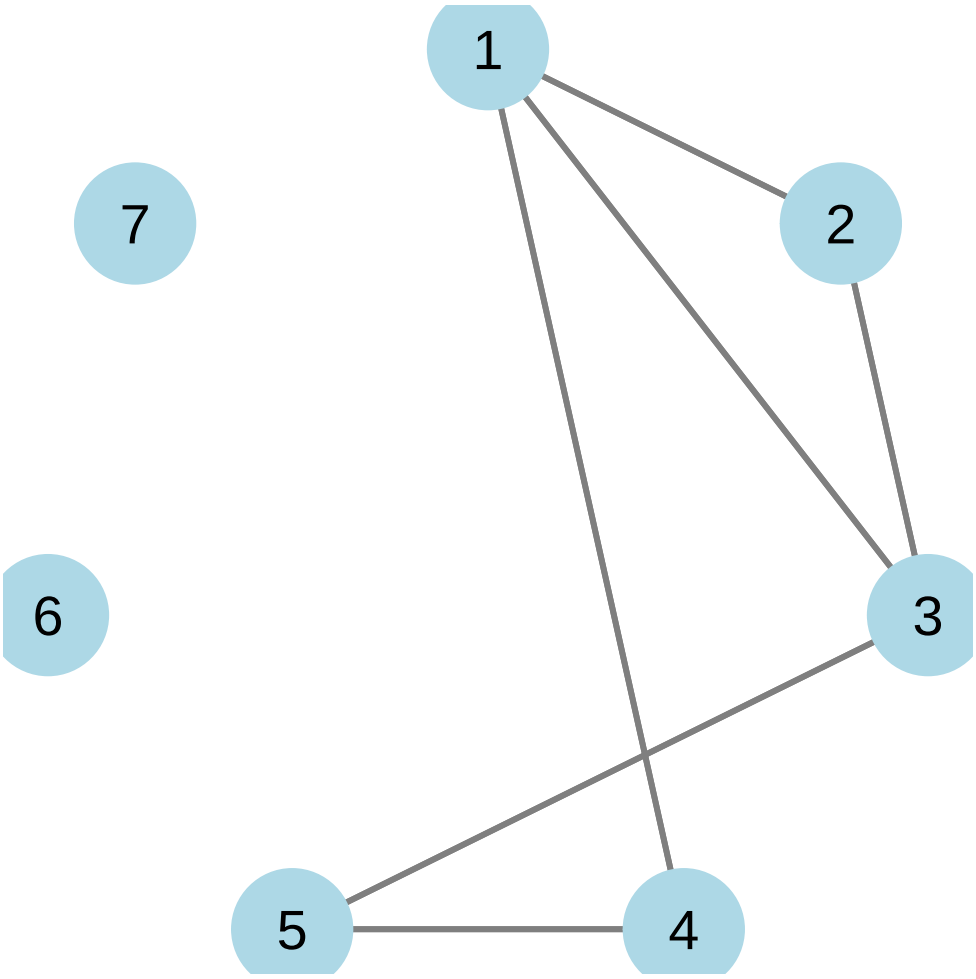

Figure 1: True phenotype graph for simulation studies.

```
R> fit

Summary: GGPA model fitting results (class: GGPA)
-------------------------------------------------
Data summary:
        Number of GWAS data: 7
        Number of SNPs: 20000
Use a prior phenotype graph? NO
mu
     estimate   SE
[1,]     1.11 0.02
[2,]     0.99 0.01
[3,]     1.18 0.02
[4,]     1.19 0.01
[5,]     1.32 0.01
[6,]     1.12 0.02
[7,]     1.30 0.02
```

```
sigma
     estimate   SE
[1,]     0.38 0.02
[2,]     0.29 0.01
[3,]     0.33 0.02
[4,]     0.29 0.01
[5,]     0.38 0.01
[6,]     0.38 0.01
[7,]     0.28 0.01
Proportion of associated SNPs
     estimate SE
[1,]     0.04  0
[2,]     0.08  0
[3,]     0.02  0
[4,]     0.04  0
[5,]     0.06  0
[6,]     0.07  0
[7,]     0.03  0
-------------------------------------------------
```

Parameter estimates and their standard errors can be extracted using methods '`estimate`'.

```
R> str(estimates(fit))

List of 10
 $ P_hat_ij      : num [1:7, 1:7] 0 1 1 0.985 0 0 0 1 0 1 ...
 $ Sum_E_ijt     : num [1:7, 1:7, 1:20000] 2 0 0 0 0 0 0 0 0 0 ...
 $ est_beta      : num [1:7, 1:7] -4.78 0 0 0 0 ...
 $ sd_beta       : num [1:7, 1:7] 0.0807 0 0 0 0 ...
 $ est_mu_vec    : num [1:7] 1.115 0.994 1.183 1.19 1.315 ...
 $ sd_mu_vec     : num [1:7] 0.0221 0.0104 0.0245 0.0133 0.0133 ...
 $ est_sigma1    : num [1:7] 0.379 0.294 0.333 0.294 0.382 ...
 $ sd_sigma1     : num [1:7] 0.01518 0.00754 0.01683 0.0095 0.01093 ...
 $ est_prob_e_ijt: num [1:7, 1:7] 0.03792 0.02993 0.00826 0.00422 0.00449 ...
 $ sd_prob_e_ijt : num [1:7, 1:7] 0.1695 0.1565 0.0837 0.0588 0.0603 ...
```

2.2 Association Mapping

Now, based on the fitted graph-GPA model, we implement association mapping with the command:

```
R> assoc.marg <- assoc( fit, FDR=0.10, fdrControl="global" )
R> dim(assoc.marg)

[1] 20000     7

R> apply( assoc.marg, 2, table )

   [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]
0 19308 18654 19715 19089 18803 18880 19444
1   692  1346   285   911  1197  1120   556
```

'assoc' method returns a binary matrix indicating association of each SNP, where one indicates that a SNP is associated with the phenotype and zero otherwise. Its rows and columns match those of input p-value matrix for 'GGPA' method. 'assoc' method allows both local ('fdrControl="local"') and global FDR controls ('fdrControl="global"'), and users can control nominal FDR level using the argument 'FDR'. The association mapping results above indicate that about 300 $\sim$ 1400 SNPs are estimated to be associated with these phenotypes under the global FDR control at 0.10 level.

'fdr' method for the output of 'GGPA' method ('fit' in this example) further provides the matrix of local FDR that a SNP is not associated with each phenotype, where its rows and columns match those of input p-value matrix for 'GGPA' method. This method will be useful when users want to scrutinize association of each SNP more closely.

```
R> fdr.marg <- fdr(fit)
R> dim(fdr.marg)

[1] 20000     7

R> head(fdr.marg)

      [,1]  [,2]  [,3] [,4] [,5] [,6] [,7]
[1,] 0.990 1.000 1.000    1    1 1.00    1
[2,] 1.000 1.000 1.000    1    1 0.86    1
[3,] 1.000 1.000 0.995    1    1 1.00    1
[4,] 1.000 1.000 1.000    1    1 1.00    1
[5,] 1.000 1.000 1.000    1    1 1.00    1
[6,] 0.655 0.695 1.000    1    1 1.00    1
```

When users are interested in the association of a SNP for certain pair of phenotypes, users can specify it using 'i' and 'j' arguments in both 'assoc' and 'fdr' methods, where 'i' and 'j' indicate indices of phenotypes of interest. For example, if users are interested in SNPs associated with both the first and the second phenotypes, we can specify this by setting 'i=1, j=1'. If the 'i' and 'j' arguments are specified, 'assoc' and 'fdr' methods return a corresponding vector instead of a matrix. The association mapping results below indicate that there are 591 SNPs associated with both the first and the second phenotypes under the global FDR control at 0.10 level.

```
R> assoc.joint <- assoc( fit, FDR=0.10, fdrControl="global", i=1, j=2 )
R> length(assoc.joint)

[1] 20000

R> head(assoc.joint)

[1] 0 0 0 0 0 0

R> table(assoc.joint)

assoc.joint
    0     1
19409   591
```

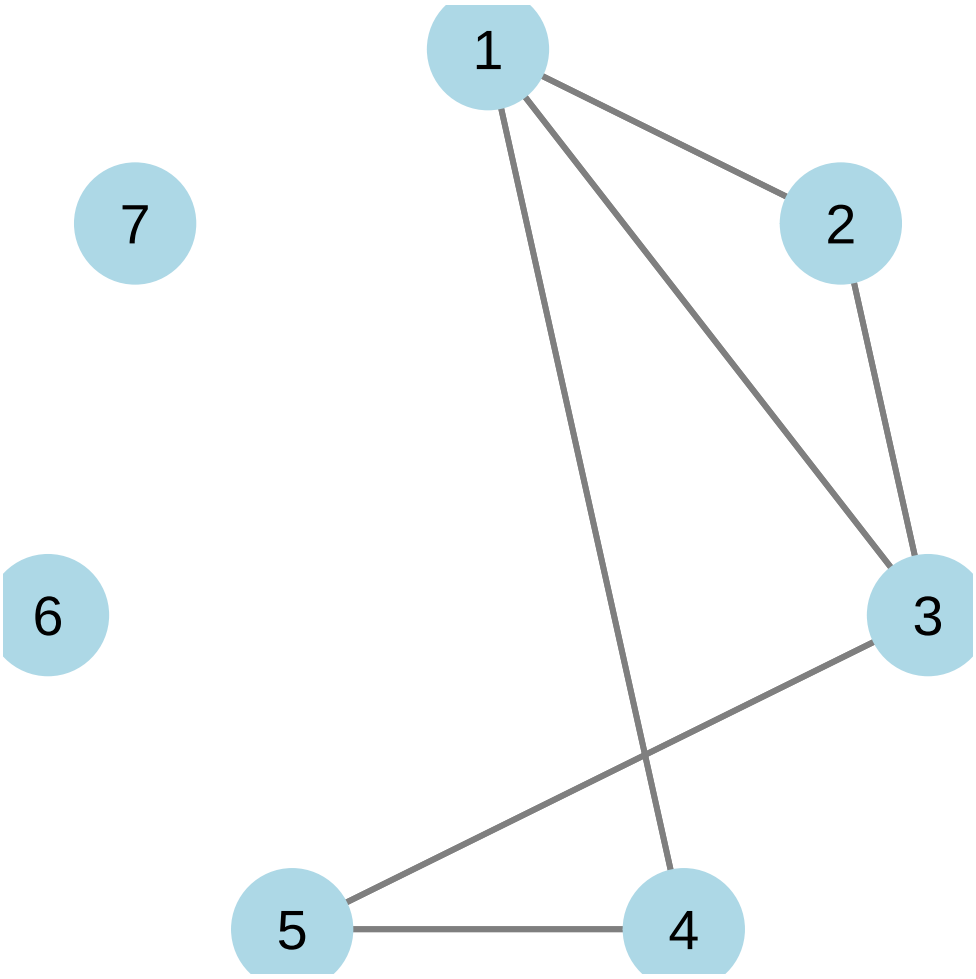

Figure 2: Phenotype graph estimated using graph-GPA.

3 Investigation of Pleiotropic Architecture Using the Phenotype Graph

In the joint analysis of multiple GWAS data, it is of interest to investigate the genetic relationship among the phenotypes. The graph-GPA framework allows users to check this using a phenotype graph. This phenotype graph can be generated by applying '`plot`' method to the output of '`GGPA`' method ('`fit`' in this example).

```
R> plot(fit)
```

Figure 2 shows the phenotype graph estimated using graph-GPA for the simulation data and users can see that it is identical to the true phenotype graph shown in Figure 1.

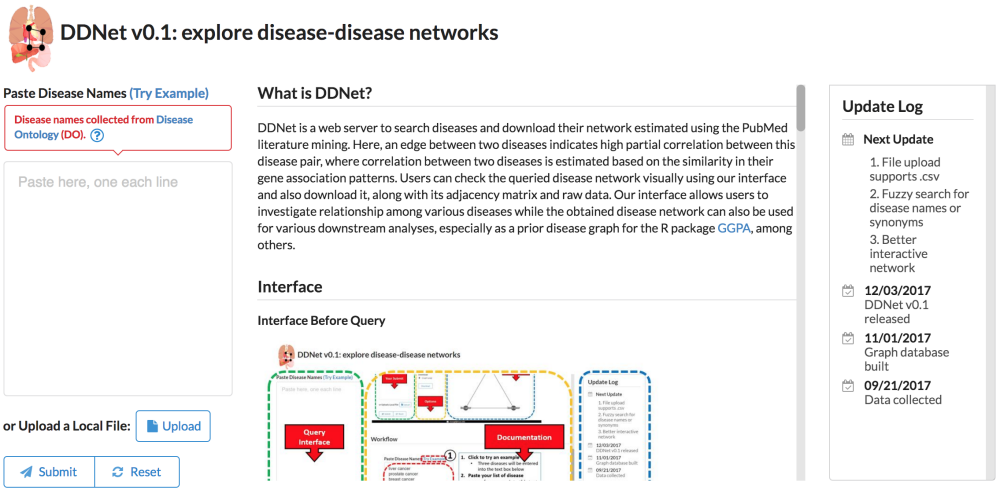

Figure 3: DDNet web interface: Step 1. Enter `http://www.chunglab.io/ddnet/` in your web browser.

4 graph-GPA Analysis Using a Prior Disease Graph

The graph-GPA was initially based on an uninformative prior disease graph [1] but later extended by allowing users to incorporate an informative prior disease graph [2]. Specifically, we proposed to generate a prior disease graph based on the gene sharing pattern between diseases in the literature mining. While we showed that this approach effectively improves the graph-GPA analysis in the sense of estimation accuracy, robustness against the collinearity, and reproducibilities between independent validation dataasets [2], it still remains burdensome for most users to implement this literature mining. Hence, in order to facilitate users' convenience, we developed *DDNet* (`http://www.chunglab.io/ddnet/`), a web interface that allows users to query diseases of interest, investigate relationships among them visually, and download the adjency matrix for the graph-GPA analysis.

First, if you open the web address `http://www.chunglab.io/ddnet/` in your web browser, you can see the web interface that looks like Figure 3. In the left side, you can a box and you can query diseases of interest. If you want to try an example list of diseases, just click "Try Example" on the top (Figure 4). Alternatively, you can upload a text file of disease names of interest using the "Upload" button. Note that we constructed our disease dictionary using the Disease Ontology database (`http://disease-ontology.org/`). Hence, if you cannot find a disease of your interest, please check the Disease Ontology database. Then, please click the "Submit" button.

Upon clicking the "Submit" button, you will see a network of the diseases you queried in the right side, as depcited in Figure 5. By either using a bar of typing a value below the "Cut-Off Value" section, you can dynamically investigate disease network structures. Here, an edge is connected between a pair of diseases if the corresponding partial correlation coefficient is larger than the specified cut-off. If you click "Download" button, you can also download the disease network plot in PNG file format.

If you click the "Table" tab above the disease graph, you can check the adjency matrix corresponding to the disease network for the specified cut-off (Figure 6). You can also check the raw partial correlation coefficient matrix by clicking the "Raw Matrix" tab below the "Table" tab. By clicking "Download" button, you can download the adjacency matrix in the CSV file format and

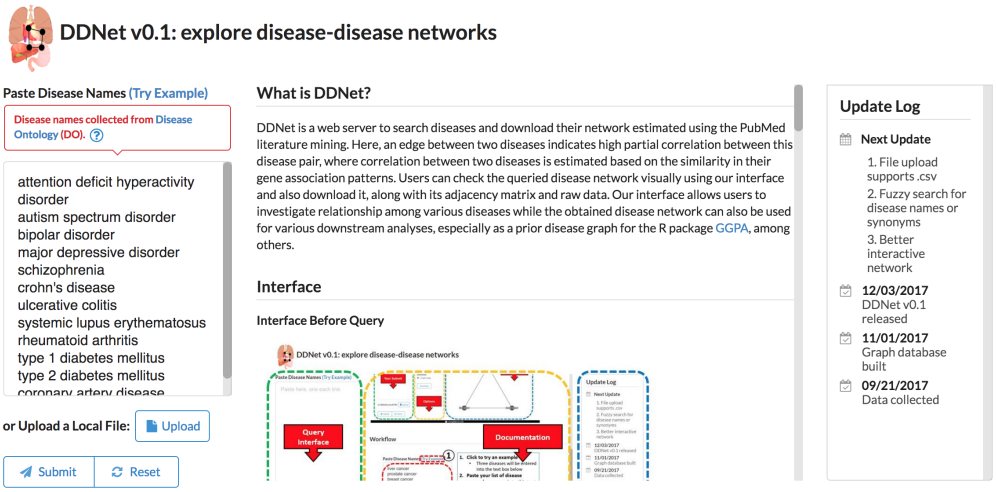

Figure 4: DDNet web interface: Step 2. Enter a list of diseases. Click "Try Example" for an example list of diseases.

this can be used as a direct input for the `GGPA` package.

Supposed that the downloaded CSV file is loaded to the R environment with the object name `pgraph` while the `pmat` has the corresponding genotype-phenotype association p-value matrix. Note that it is assumed that objects `pgraph` and `pmat` have the same number of columns and also share the same column names. Then, you can fit a graph-GPA model using the downloaded disease network as a prior distribution using the following command line. Other functions will work exactly in the same way as described in Section 2.

```
R> fit <- GGPA( pmat, pgraph )
```

References

[1] Chung D, Kim H, and Zhao H (2017), "graph-GPA: A graphical model for prioritizing GWAS results and investigating pleiotropic architecture," *PLOS Computational Biology*, 13(2): e1005388.

[2] Kim H, Yu Z, Lawson A, Zhao H, and Chung D (2018), "Improving SNP prioritization and pleiotropic architecture estimation by incorporating prior knowledge using graph-GPA," Bioinformatics, bty061.

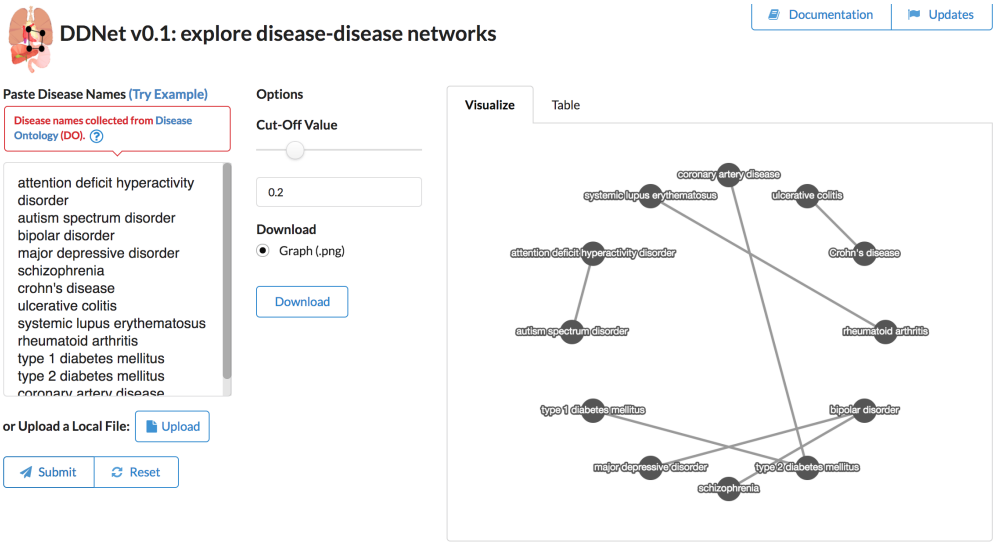

Figure 5: DDNet web interface: Step 3. Investigate a disease-disease network visually.

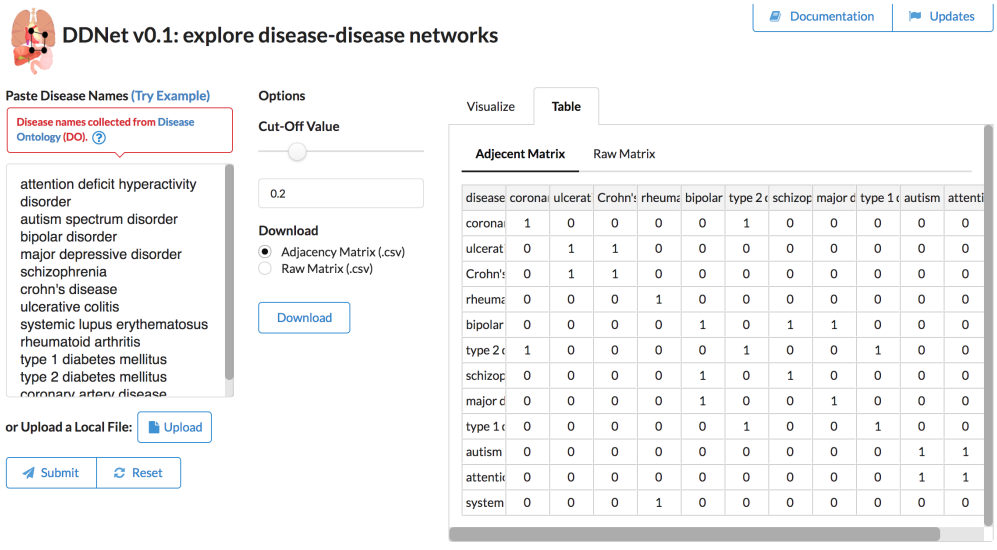

Figure 6: DDNet web interface: Step 4. Download an adjacency matrix for the `graph-GPA` analysis.