# Supplementary information

## 1. Data pre-process

### 1.1 Filter lowly expressed genes

Some genes may not consistently expressing due to tissue specific effect, and too many genes could cost too much computational resources. We filtered some genes that were expressed lower than the other most genes. For the time serious gene expression data of TNF induced inflammation, we set up the following procedure: if the expression values of a gene ranks at the bottom 20% in no less than 80% data sets, this gene is identified as lowly expressed. For the clinical drug response data set, we calculated the mean values of each gene and filtered out the bottom 20% genes with lowest mean values.

### 1.2 Calculate differential expression of each gene

For the time series gene expression data set of TNF induced inflammation, we calculated the largest log2 fold change of each gene. The basis value was set to be the initial gene expression at time 0, and the largest fold change was calculated as a differential between an expression value at time T and time 0. We do not treat this differential expression as up-regulated or down-regulated, since these genes may not consistently up or down regulated through all time points. Genes with fold change larger than 2 were identified as seed genes.

For the clinical drug response data set, we applied Wilcox rank-sum test to calculate the p value of each gene, under the null hypothesis that the mean values of expression in non-responders or responders are equal. We identified the genes with p value smaller than 0.01 as seed genes

### 1.3 Calculate the edge weights in the gene network

A gene network was set up by the common genes between gene expression data and a protein-protein network. The edges in the network were weighted by the Pearson correlation coefficient of the connected pairs of genes.

## 2. Implementations of ClustEx2

### 2.1 Density-based hierarchical clustering

In a gene network, each gene has a gene importance score . Each pair of genes has a similarity. Give a threshold , the process of module identification is:

1. Sort all genes according to in descending order. The gene has the largest gene importance score and ranks the first. is initiated as the first module.
2. Genehas the largest gene importance score. Considering each pair of similarities between and , the framework takes one of the three following actions:
3. If , then is initiated as a new module.
4. If , then is added to the module whichbelongs to.
5. If , then merge these modules which belongs to and addinto the newly merged module.
6. Iterate step 2 until all genes are clustered.

The threshold directly affects the results of clustering. However, the value of is very difficult to precisely control. An alternative parameter, called “neighborhood” is defined to represent the effect that brings to output modules. We can imagine a circle with one gene as its central and n as its radius. This circle is the gene’s “neighborhood” area. If another gene lies in this circle, the two genes are merged to be in one module. Its value is assigned to the rank of percentage of gene-gene similarities. By sorting all the gene-gene similarity scores decreasingly, one neighborhood is set to be quotient of the rank divided by the number of gene-gene pairs. For example, a neighborhood 0.1 ranks at the largest 10% of all gene-gene similarities. With different values of neighborhood ranging from 0 to 1, output modules can be different in both size and seed gene fraction. Thus, if the value of neighborhood is smaller, it is more difficult for cluster one gene with another.

### 2.2 Parameter tuning

#### 2.2.1 overview

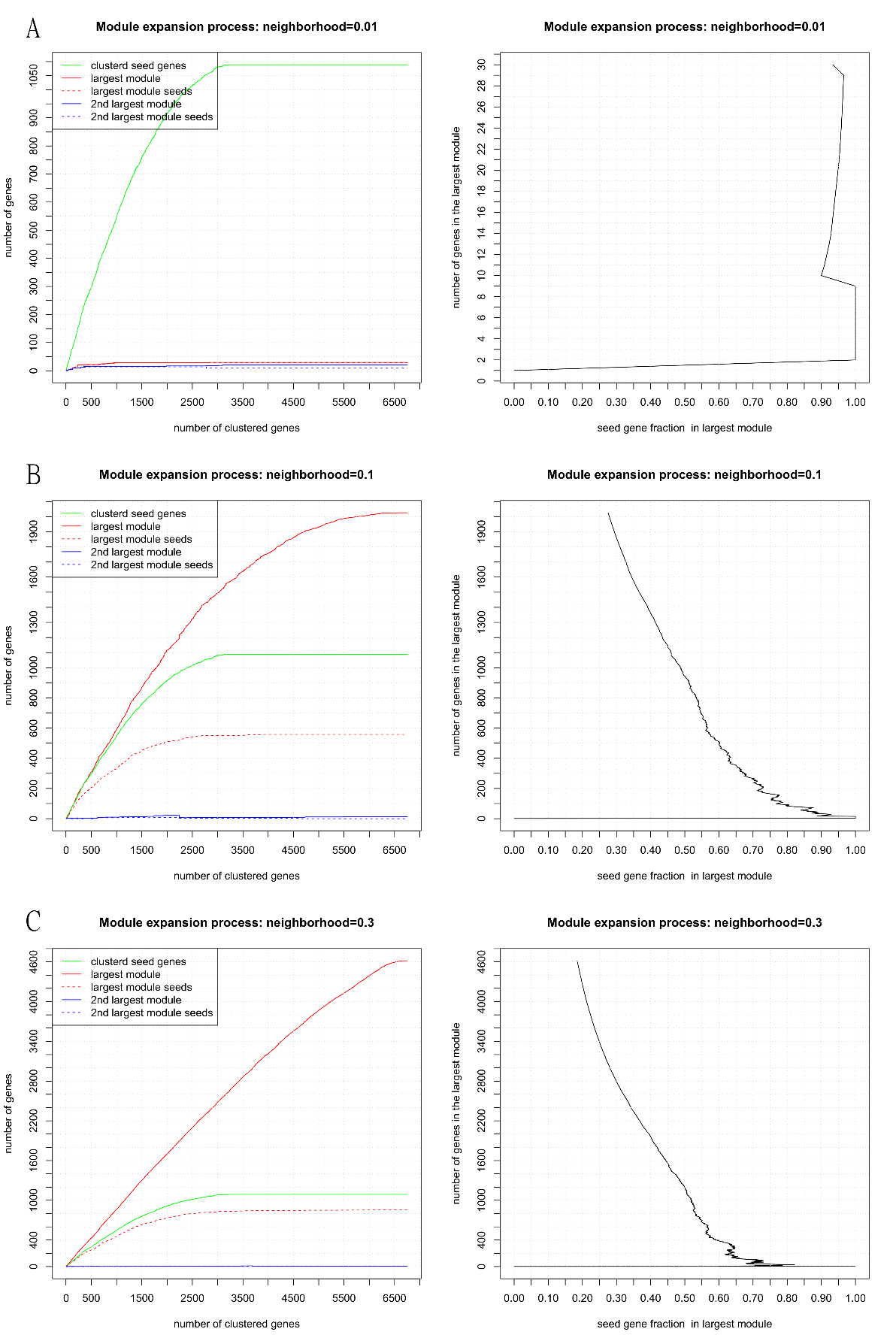
The density-based hierarchical clustering framework makes it possible to visualize the key features of the largest two modules during clustering processes, and therefore helps users flexibly control the output module size. Three parameters, including the above-mentioned neighborhood of the density-based hierarchical clustering, the coefficient of diffusion kernel and the restart probability of random walk, affect the identification of gene modules. Since module identification do not have a loss function similar to supervised learning, the fraction of seed genes in the largest module is used instead to measure whether a set of parameter values are suitable. As for the coefficient of clustering, which *directly* affects the fraction, we give a default value and a principle to tune it. As for the other two which *indirectly* affects the fraction, we suggest a set of default values. As a consequence, ClustEx2 is able to instruct users about what effects different parameters will bring to output modules.

#### 2.2.2 Neighborhood of density-based hierarchical clustering

The adjustable coefficient of the clustering method is called “neighborhood” and it controls whether a gene is close enough to some module. “n”, short for “neighborhood”, directly affects the seed gene fraction and controls the intra-cluster similarity of gene modules. According to its parameterization, 0.1 is the highest 10% edge similarity score amongst all the gene-gene similarities and is suggested to be the first value of neighborhood for usage. Users are encouraged to set it smaller than this default value, as long as the seed gene fraction and the size of the largest module are satisfactory,. The principle is that with the same number of genes clustered, a smaller neighborhood results in more modules and higher intra-cluster similarity of each module. If we choose an extremely small neighborhood as 0, then every gene forms a module; if extremely large as 1, all genes are in one module. We suggest choosing an appropriate neighborhood that is small enough to result in high intra-cluster similarity of each module but not too small to avoid that every gene forms a module by itself.

Figure S1 shows the clustering processes that record the changing fraction of seed genes as modules expand under three different values of n, which are small, medium and large, corresponding to 0.01, 0.1 and 0.3. The curves show their differences apparently in the largest module expansion and the fraction of seed genes in this largest module. In Figure S1(A), the value of n is 0.01 and the corresponding largest module expands to contain only 30 genes. This is because that n is so small that almost every gene forms a module by itself. In Figure S1(C), when n is 0.3, the expansion of the largest module and the number of clustered genes almost make up a diagonal line. This is because that the neighborhood is so large that almost all genes are clustered into one module. In Figure S1(B) when neighborhood is 0.1, the medium value of neighborhood ensures that the intra-cluster similarity of gene modules are large. Therefore we recommend 0.1 as the default value of n. But it is suggested using a neighborhood smaller than 0.1 if users are satisfied with the seed gene fraction and module size under a smaller neighborhood.

Typically we determine output modules under one value of the neighborhood. However, there are occasions that users want to get higher or lower seed gene fractions with a fixed module size, or sometimes larger or smaller module size with a fixed seed gene fraction of the largest module. In order to meet these needs, the features are visualized during different clustering processes with different values of the neighborhood (Figure S1). Therefore, we need to determine the value of neighborhood along with at least one feature of the largest module. The tip is that we only need to use the principle of tuning the value of neighborhood to obtain final results. For example, in order to identify modules with fixes sizes, a larger neighborhood typically result in a higher seed gene fraction, but a lower intra-cluster similarity.



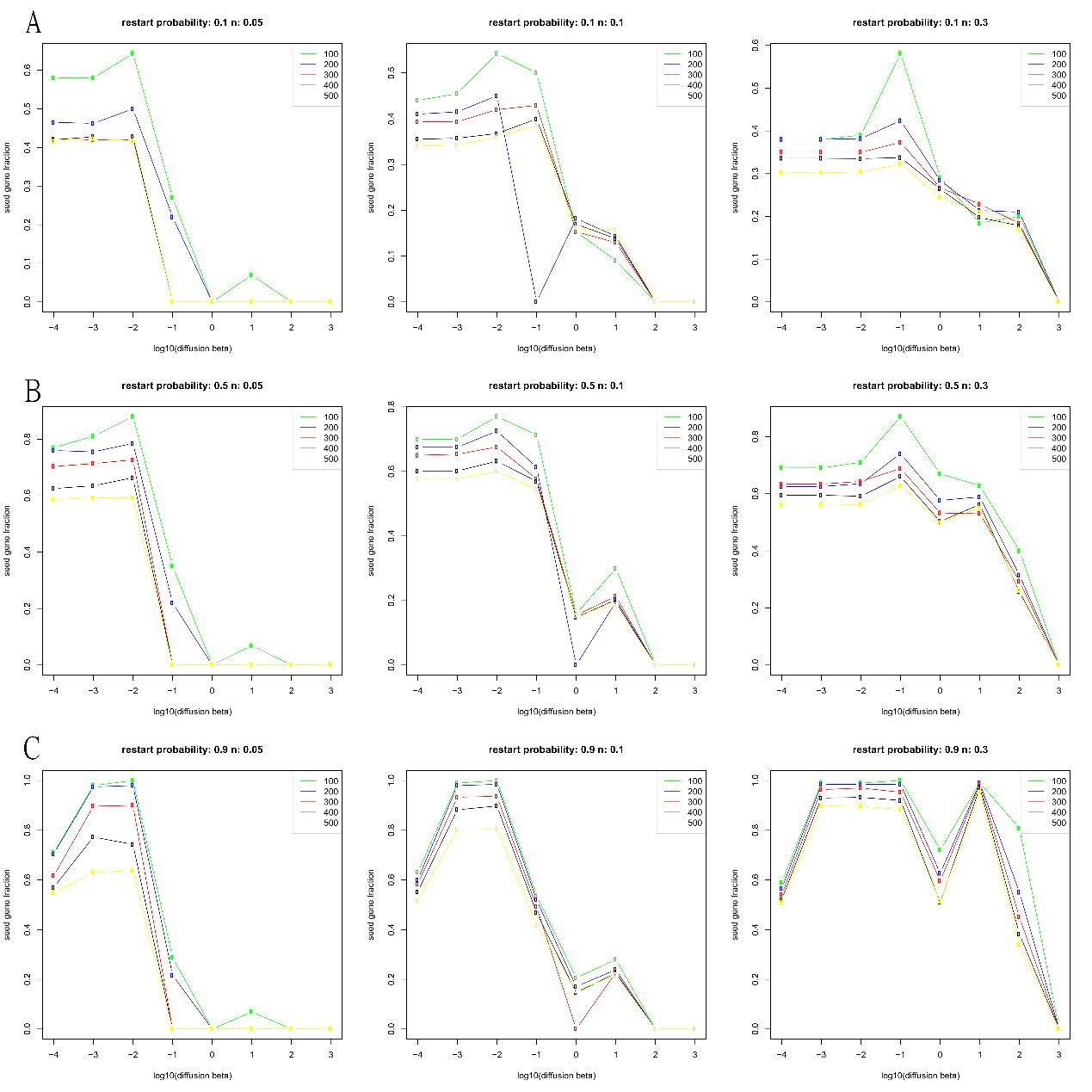
**Figure S1**. Variation of key features during module expansion processes (A) under the neighborhood equaling to 0.01, (B) under the neighborhood equaling to 0.1, (C) and under the neighborhood equaling to 0.3.

#### 2.2.3 Coefficient of diffusion kernel and random walk

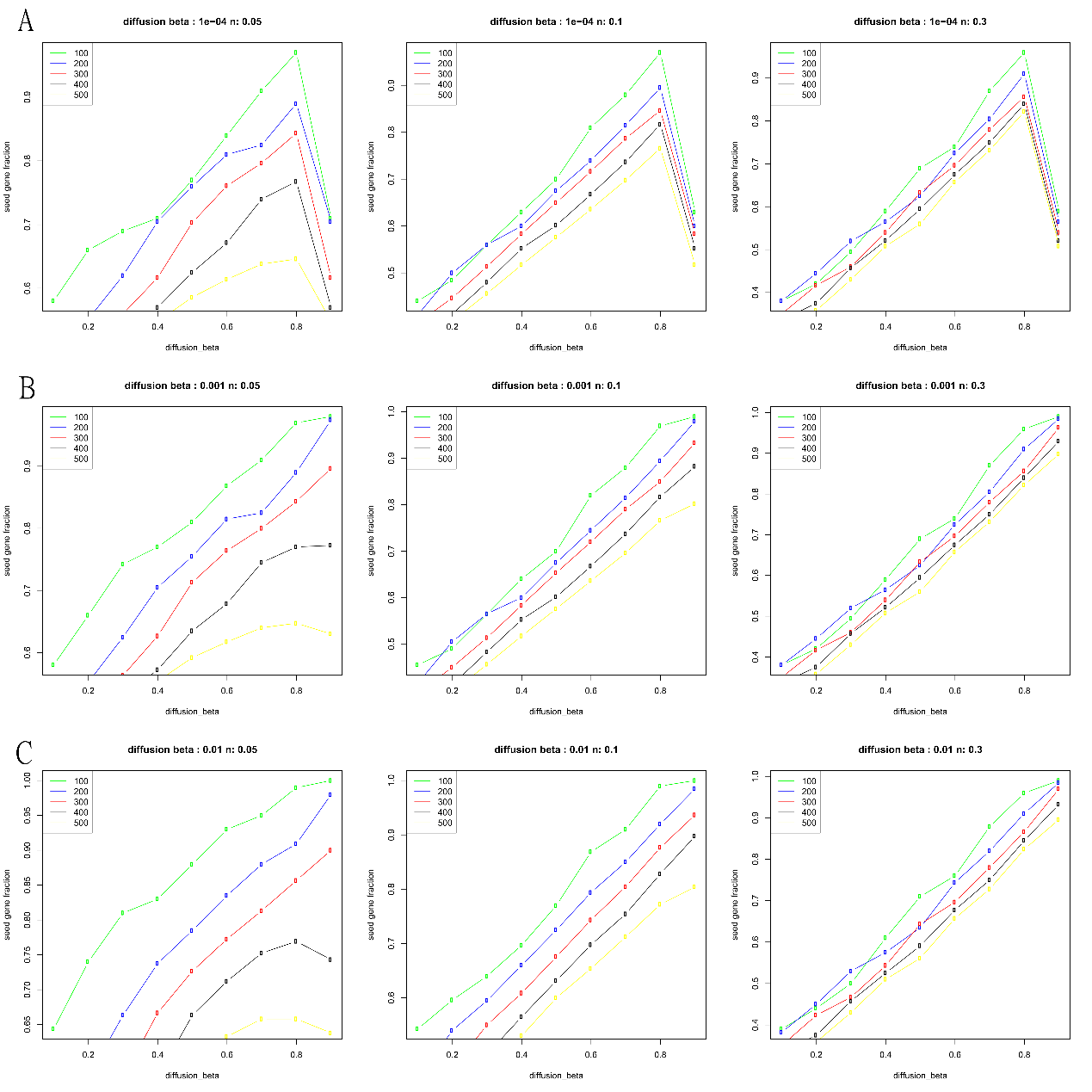
The coefficient of diffusion kernel, “beta”, is set to be 0.01 by default. The coefficient of random walk, “r”, short for the restart probability, is set to be 0.5 by default. These two parameters indirectly affect the fraction of seed genes in modules. We used different value combinations of beta and r for module identification using the two real dataset discussed in the main article. To obtain default values of beta and r, modules with sizes ranging from 100 to 500 at a step of 100 are identified, under three values of neighborhood as discussed above. The goal is to see what values of beta and r lead to as large seed gene fraction as possible. Since beta impacts on the gene-gene similarity scores which are used to define the value of neighborhood and r affects the order of all genes, beta are more closely related to neighborhood and thus have a more direct impact on output modules.

Firstly, we checked the variation of seed gene fractions under different values of beta. Figure S2 describes that when beta ranges from -3 to 4 (log10 scale), how the seed gene fraction varies in the largest module: 1) each three subfigures in a column shows no matter what the value of r is, if beta is smaller than 0.1, the seed gene fraction is much larger; 2) each three subfigures in a row shows when the value of r is fixed, if beta is smaller than 0.1, the seed gene fraction is larger. Therefore, the value of beta can be 0.0001, 0.001 and 0.01 no matter what the value of r is, however, one thing that have to be mentioned is the when beta is smaller, the calculation of diffusion kernel relies more on the precision of Eigen and Clapack. Thus we recommend beta to be 0.01, which results in good seed gene fraction and does not rely too much on the precision of the two matrix calculation libraries.

Secondly, we checked the fraction of seed genes under different values of r. Notice that the value of beta ranging from -4 to -2 (log10 scale) is needed, since results under larger values of beta including -1 to 3 are left out as discussed above. Figure S3 describes that when r ranges from 0.1 to 0.9 with a step of 0.1, how the seed gene fraction varies in the largest module: 1) each three subfigures in a row shows that when beta is fixed, a larger r leads to a larger fraction except when r is 0.9; 2) each three subfigures in a column shows when neighborhood is fixed, a larger value of r also to a larger fraction. Therefore, a larger restart probability leads to larger fraction. However in this case, about 1,100 seed genes take up almost one sixth of all genes and might incline to be closely connected to each other. Conceptually, restart probability is a balance between gene context information and network topology. Thus, we set it to be 0.5 by default, which represents the same amount of information from network topology and gene context information.



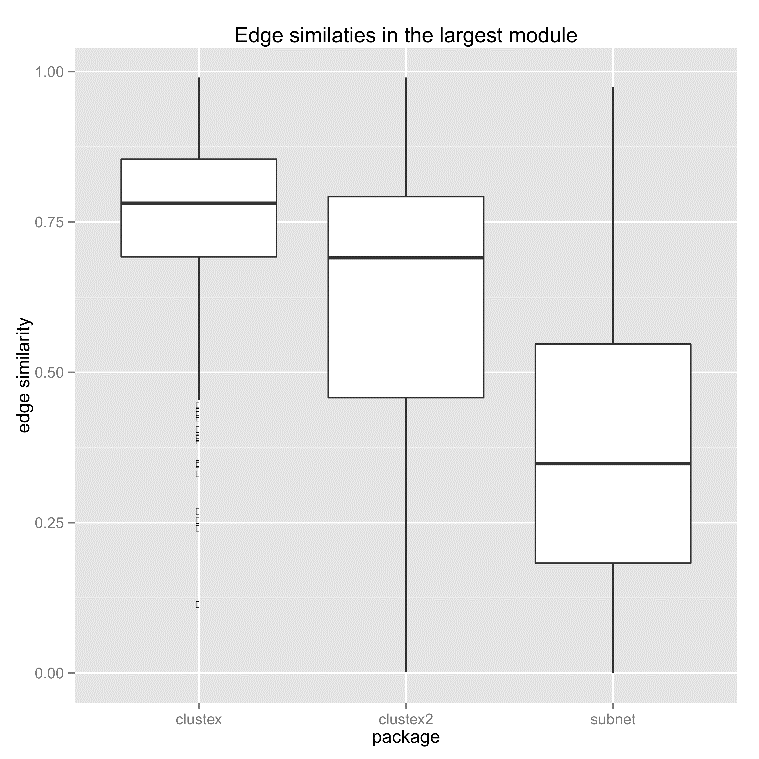
**Figure S2**. TNF dataset (A) the seed gene fraction in the largest module of which the size ranges from 100 to 500 when random walk restart probability is 0.1 (B) the seed gene fraction when restart probability is 0.5 (C) the seed gene fraction when restart probability is 0.9



**Figure S3**. TNF dataset, (A) the seed gene fraction in the largest module when the module size ranges from 100 to 500 and beta is 0.0001 (B) the seed gene fraction when beta is 0.001 (C) the seed gene fraction when beta is 0.01

## 3. Results

### 3.1 Responsive gene modules of TNF induced HUVECs



**Figure S4**. The box plots of the absolute value of co-expressions between gene pairs that are linked by edges in the modules identified by ClustEx, CLustEx2 and SubNet.

#### 3.1.1 Basic information of the largest modules

Table S1 The largest responsive gene module in HUVECs

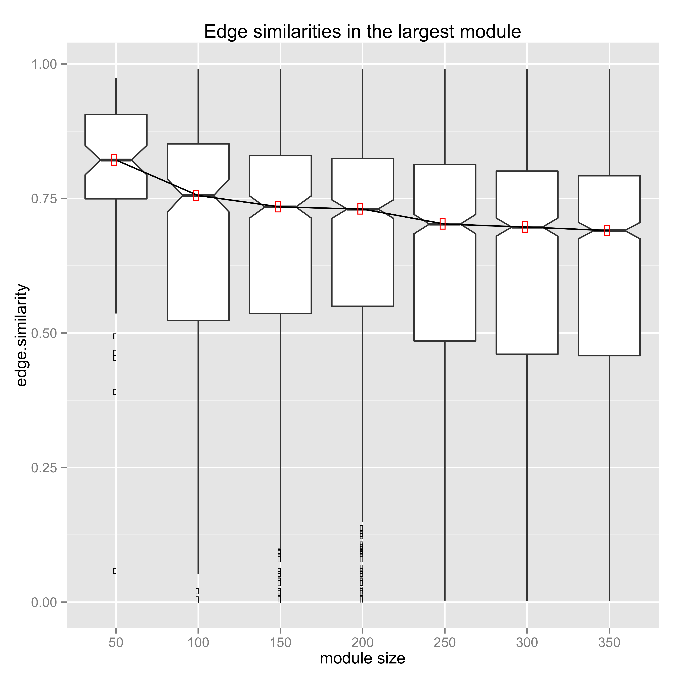
|  |  |  |  |
| --- | --- | --- | --- |
|  | ClustEx2 | SubNet | ClustEx |
| Fraction of seed genes | 0.669 | 0.802 | 0.644 |
| Size | 353 | 358 | 233 |
| # of KEGG pathways\* | 27 | 34 | 25 |

# of KEGG pathways\*: number of significantly enriched KEGG pathways with p values corrected by Bonferroni

#### 3.1.2 How to determine the module sizes

The largest module size of ClustEx was chosen according to the principles in its paper. The corresponding figure (Additional file 2) shows that there is a big change after about 130 genes are clustered. After that, the clustering process is quite smooth. Therefore, we chose the largest module to have about 150 seed genes. In order to compare the results of ClustEx2 and ClustEx without differences caused by the different fractions of seed genes, we chose the module size of ClustEx2 to be around 350. In order to compare the performances of ClustEx2 and SubNet, we made their modules have close sizes. This is also partially because that SubNet is not able to control the number of seed genes in the largest module, which makes it impossible to consider similar seed gene fractions.

The proceeding of the density-based clustering algorithm is stopped, since the intra-cluster similarity of one module decreases as more genes are clustered into it. This is demonstrated by the growth of the largest responsive gene module (Figure S4).



**Figure S5**. As the module grows, the median of edge similarity decreases. genes in the module are becoming less closely connected.

#### 3.1.3 Performance on reference gene sets

The percentage of overlapping genes between the genes in the module and the reference gene set is defined as sensitivity, which is formulated as

The signal-to-noise ratio was used to evaluate the significance of overlapping. The signal is defined as the number of overlapping genes between module genes and reference genes. The noise is defined as the mean of mean of the numbers of overlapping genes between control modules and reference gene set. These definitions were adopted by ClustEx. The signal-to-noise ratio is defined as follows.

### 3.2 Gene modules associated with clinical drug response

C:\Users\zding\workspace\projects\dissertation\clustex2\Figures\Figure5 - 20170507-01.tif

**Figure S6**. Clinical drug response associated gene modules of cisplatin in (A) BLCA and (B) CESC.

**Table S2** The seed genes in the two drug response associated gene modules

|  |  |  |
| --- | --- | --- |
| **module** |  | **Seed genes** |
| BLCA |  | HLA-DRB5, CD3D, IL18RAP, HCLS1, HLA-DPA1, PDCD1, HLA-DQB1, THEMIS, HLA-DPB1, IL18BP, KLRK1, HLA-DQA2, ZAP70, CD300LB, LY9, KLRC3, HLA-DRA |
| CESC |  | NDUFA12, THOP1, DDX39, TPRKB, NDUFA6, GCDH, NDUFA7 |

**Table S3** The seed genes in the two drug response associated gene modules

|  |  |  |
| --- | --- | --- |
| **module** |  | **Non-seed genes** |
| BLCA |  | CCL5, CD3E, CD74, HLA-DOA, HLA-DQA1, IL18, SLAMF1, CD8A, HLA-DMB, HLA-DMA, ITK, SH2D1A, SLA, CD200R1, CD2, IKZF1, FGR, TBX21, KLRD1, HLA-DRB1, HLA-E, LAT, CEACAM21, IL2RG, LCP2, CD4, HCST, TYROBP, GRAP2, PTPRC, CCR5, TNFSF13B, B2M |
| CESC |  | FKBP3, NDUFB7, BIRC5, ATP5H, C14orf166, NDUFA8, PELP1, PTMS, NDUFA13, TIMM9, TIMM8A, DPP7, LAGE3, BLOC1S1, SARNP, NDUFA11, NDUFB6, TIMM44, NDUFB1, DHPS, OSGEP, SURF2, HTRA2, NDUFS5, COX5A, NDUFS7, EIF2B3, PSMG2, ATP5O, NDUFS3, PSMB7, DIABLO, NDUFB8, TIMM13, ECSIT, SART1, PARP2, NDUFB9, NDUFA2, NDUFS4, THOC7, NDUFA9, DTNBP1 |

## References

1. Hsu SD, Tseng YT, Shrestha S, Lin YL, Khaleel A, Chou CH, Chu CF, Huang HY, Lin CM, Ho SY *et al*: **miRTarBase update 2014: an information resource for experimentally validated miRNA-target interactions**. *Nucleic Acids Res* 2014, **42**(Database issue):D78-85.