Chapter 15

Bioinformatic Approach to Identify Chaperone Pathway Relationship from Large-Scale Interaction Networks

Yunchen Gong, Zhaolei Zhang, and Walid A. Houry

Abstract

We describe a computational protocol to identify functional modules and pathway relationship of chaperones based on physical interaction data derived from high-throughput proteomic experiments. The protocol first identifies interacting proteins shared by the different chaperone systems to organize the chaperones into functional modules. The chaperone functional modules represent groups of chaperones that are involved in mediating the folding of the shared interacting proteins. Either the chaperones in a module can function along a single folding pathway of a given substrate protein or the substrate protein might have two or more different folding pathways that the chaperones act on independently. As described in our computational protocol, probabilities of these pathway relationships between two chaperones in a two-component chaperone module can be determined using whole-genome expression and cellular pathways as reference. This protocol is potentially useful for identifying functional modules and pathway relationships in other biological systems that involve multiple proteins with many identified interactions.

Key words: Chaperone, Protein interaction network, Functional module, Pathway relationship

1. Introduction

Molecular chaperones represent a large and diverse group of proteins whose general function is to maintain protein homeostasis in the cell (1, 2). Consequently, molecular chaperones play a wide range of cellular roles including protein folding and unfolding, protein disassembly and disaggregation, protein degradation, protein translocation, endoplasmic reticulum associated protein degradation (ERAD), and ribosomal RNA processing among many other functions. In the well-studied model organism *Saccharomyces cerevisiae* (budding yeast), there are 7 small heat shock proteins, 3 chaperones of the AAA+ family, 8 of the CCT/TRiC complex,

Stuart K. Calderwood and Thomas L. Prince (eds.), *Molecular Chaperones: Methods and Protocols*,
Methods in Molecular Biology, vol. 787, DOI 10.1007/978-1-61779-295-3_15,

6 of the prefoldin/GimC complex, 22 Hsp40s, 1 Hsp60 (& 1 Hsp10), 14 Hsp70s, and 2 Hsp90s (3). These 63 chaperones are localized in the cytoplasm/nucleus, mitochondria, and the endoplasmic reticulum.

Recently, we have identified the TAP-tag based interactors for all of these chaperones (4). A total of 21,687 unique pairs of interactions were identified with high confidence. These interactions are between the 63 chaperones and a total of 4,340 other proteins; in addition, there are 259 chaperone–chaperone interactions. All of our data is deposited in a publicly database that we created and termed ChaperoneDB (http://chaperonedb.ccbr.utoronto.ca/).

Two chaperones interacting with a given protein might functionally collaborate to assist in the folding of that protein or one chaperone might be redundant with the other. Both of these scenarios had been experimentally observed (see e.g. refs. 5, 6). In the former case, the chaperone-assisted folding of the substrate protein is along a single pathway, while in the latter case, the chaperone-assisted folding can proceed along alternate multiple parallel pathways (Fig. 1). Analysis of the large-scale chaperone interaction data that we have recently published (4) allowed us to identify "chaperone modules," herein defined as a group of chaperones interacting with a common set of proteins. It also allowed us to determine whether chaperones in a two-component module act along single or multiple folding pathways for a given protein substrate (Fig. 1). In this chapter, we present the details of the protocol applied in that previous work and focus on the algorithmic and programming aspects.

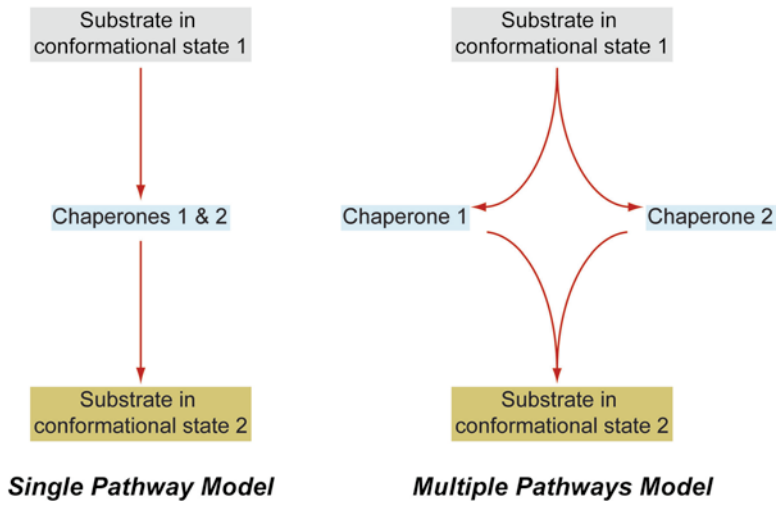

Fig. 1. The schematic depicts chaperone-mediated single pathway and multiple pathways folding models.

Briefly, our protocol consists of four steps: (1) Raw interactions are first filtered based on their experimental scores and demonstrated interactions in public interaction databases. (2) Chaperone functional modules are then inferred based on the numbers of shared interaction partners using a *Z*-score criteria. (3) To reveal the pathway relationships between the chaperones in a two-component functional module, genome-scale gene expression data are analyzed for all protein pairs located in the same and/or different pathways. (4) Finally, a statistical integration approach is applied for calculating the probability of the pathway relationship of the two chaperones.

2. Programs and Data Sources

This bioinformatic protocol relies heavily on implementations of a variety of algorithms for data processing and analysis. Some of them are simply implemented in a programming language of user's choice, while others are found in existing software packages. A scripting language such as PHP (http://www.php.net) and Perl (http://www.perl.org) is good enough for routine data processing such as custom data sorting, simple calculations, and visualization. The statistical package R (http://www.r-project.org/) is used to calculate enrichment of the documented protein–protein interactions in the target dataset. This is useful for data filtering. The network visualization package CytoScape (7) is used for visualization of complex networks. When colors of nodes and edges are varied, it is much easier to create graph files with a custom script than to manually input and modify the network components in CytoScape (see Note 1).

This protocol also needs several external low- or high-throughput datasets for data filtering and establishment of pathway relationships between the pair of chaperones in a two-component functional module. The interaction database BioGRID (8) provides manually curated known interactions from both low- and high-throughput experiments. The MIPS proteins complex database (9) provides yet another source of documented interactions. Our protocol uses BioGRID interactions for calculation of enrichment of known interactions in the target dataset, and MIPS interactions are used as criteria for reliable interactions, which are typically obtained from low-throughput methods, such as immunoprecipitation. Microarray datasets and KEGG pathway database (10) are used to establish a correlation between coexpression and pathway relationship. This correlation is the basis for inferring the pathway relationship between two chaperones in a functional module, as described below.

3. Methods

3.1. Data Filtering

Proteomics data are known to be noisy, with many false-positive interactions. It is, therefore, important to filter out these false positive interactions as much as possible. While proteomic interactions are often scored and ranked based on mass spectrometry database searches, comparison of the raw interactions obtained in proteomic studies with validated interactions from small-scale biochemical studies is also very important for data filtering. In our protocol, two datasets of such reliable interactions are used for this purpose.

3.1.1. Filtering Interactions Using MIPS Complexes

The MIPS database (9) contains 215 well-established complexes curated from numerous biochemical publications. Although this dataset has not been updated for a few years (latest complexes were derived from a publication in the year 2004), it is still frequently used as a gold standard for protein–protein interactions. To use this database in data filtering, complexes are downloaded from the MIPS Web site, and protein–protein interactions are assigned to each pair of proteins found in a complex. This interaction list comprises a reference interaction set. The raw interactions from the TAP-tag pulldown experiments in which the interactors are identified by mass spectrometry are then screened for their existence in the reference interaction set from MIPS. The raw interactions are further grouped into bins of scores derived from the mass spectrometry experiments. Histograms of the frequency distribution of the bins of both the raw interactions and those found in the reference interaction list are plotted. One can determine the score cutoff for reliable interactions in the raw data based on the score distribution for the reference interactions.

3.1.2. Enrichment of Known Interactions Using BioGRID

To further confirm the selection of score cutoff in the above procedure, the interactions selected from the above protocol can be compared to those from a curated interaction database. We use BioGRID (8) for this purpose. BioGRID documents published interactions from both high- and low-throughput experiments. As with MIPS, we use only the low-throughput interactions, as they are deemed more reliable. BioGRID contains many more interactions than MIPS, since it is continuously being updated. The principle of testing for enrichment for known interactions in the chaperone interaction data is as follows. Given all interactors of the chaperones and the interactors for each chaperone from the filtered data, find the interactors already documented in BioGrid for a particular chaperone, and the interactors found in both filtered chaperone interactors and BioGrid. We will thus have four subsets:

1. Filtered interactors of all chaperones (N).
2. Filtered interactors of one chaperone (m).

3. BioGrid interactors present in filtered interactors of all chaperones (n).
4. BioGrid interactors present in filtered interactors of one chaperone (k).

The fold of enrichment of BioGrid interactors that are also found in the filtered data is then:

$$\text{Enrichment\_fold} = \frac{(k/m)}{(n/N)}. \quad (1)$$

To test the significance of this enrichment, we assume a hypergeometric distribution:

$$f(k,N,m,n) = \frac{\binom{m}{k}\binom{N-m}{n-k}}{\binom{N}{n}}. \quad (2)$$

The p-value is calculated as the sum of f for all $i \geq k$, where i is the number of chaperone interactors that are found in BioGrid when m interactors are randomly drawn. A significant enrichment is when $p < 0.05$. Implementation of this algorithm can be found in the statistical package R using the function *phyper*, which is called with command line, *phyper(k,n,N−n,m,lower.tail=FALSE)*. In plain language, it refers, for example, to the probability of getting k or more red balls by chance, when drawing m balls from a mix of n red balls and $N-n$ blue balls.

If $p < 0.05$, we consider that the known interactors of a chaperone are enriched in the filtered dataset. Enrichment of known interactors for most of the chaperones would indicate the reliability of the filtered dataset.

3.2. Identification of Functional Chaperone Modules

3.2.1. Determination of the Number of Shared Interactors Among a Group of Chaperones

Two or more chaperones that interact with a common protein either do so to work together to promote the proper folding of that protein or the common protein has several alternative (parallel) folding pathways that the chaperones act on independently. We call the former scenario, the "single pathway model," and we call the latter the "multiple pathways model" (Fig. 1). In either case, the chaperones are considered to form a functional module. We find that the number of chaperones in the module typically ranges from two to five (4).

As a first step to identify the functional chaperone modules, we count the shared protein interactors for each group of chaperones

with varied sizes. The following pseudo-code finds modules of arbitrary size:

```
Given n chaperones with their interactor list
Given the size of the functional module m < n
For i1=1 to n-(m-1)
  For i2=2 to n-(m-2)
      .....
          For im=m to n
              count the interactors shared by chaperone i1, i2, ..., im
          Endfor
      .....
  Endfor
Endfor
```

The numbers of shared interactors are stored in an array or written to a file for use in the next step.

3.2.2. Using Z-Score for Chaperone Module Identification

Given the number of shared interactors for a group of chaperones of a specified size, the *Z*-score is calculated for each group as follows:

$$Z = \frac{x - \bar{x}}{\sigma}, \tag{3}$$

where x is number of shared interactors for that group of chaperones, $\bar{x}$ is the average number of interactors for all chaperone groups, and σ is the standard deviation of the number of interactors, calculated as:

$$\sigma = \sqrt{\frac{1}{n}\left(\sum_{i=1}^{N} x_i^2 - n\bar{x}^2\right)}. \tag{4}$$

Here, x and $\bar{x}$ are defined as above, and n is the total number of chaperone groups. If there are a total of N chaperones, and if there are m chaperones in a given group, the calculation of n is as follows:

$$n = C_N^m = \frac{N!}{(N-m)!m!}. \tag{5}$$

Hence, the *Z*-score is calculated for each chaperone group. Chaperone modules are identified as those with *Z*-score ≥ 2.

3.2.3. Further Validation of Chaperone Modules: Retrieving Functional Modules from a Consensus Chaperone Network

It is likely that for some well-studied chaperones, models of their networks have been previously proposed. For example, Young et al. (11) compiled a chaperone network with the components HSP40, HSP70, HSP90, PFD, and CCT. In these networks, the nodes represent chaperones or chaperone complexes and the edges are flow of the protein substrates as they change conformation from newly translated peptides to the folded native state. Chaperone modules can be identified in these networks based on the above definition. Identifying chaperone modules from an expert annotated chaperone network is useful in validating modules obtained from the experimental high-throughput data.

We identify functional modules from the chaperone network models in three steps. The first step is to identify all chaperones on pathways starting from the nascent peptide to the folded protein. The second step is to group the pathways, with each group containing a specified number of pathways. In the third step, each pathway group is "cleaned" so that each chaperone appears only once.

Step 1: Searching all chaperone pathways from the network
There are two algorithms in graph theory for a search of all pathways from a network, namely, breadth-first and depth-first search, respectively (12). Here, we list the pseudo-code for breadth-first algorithm as an example, starting from the list of edges in the model network, which is a directed acyclic graph (DAG). Note that there is a function expanding_path, which could be called iteratively.

```
Given the edges of the network
For each edge
    Put the nodes in a two-dimensional array as edges[start] ≥
array(ends)
Endfor
Define candidate_pathway array with the nascent polypeptide as both the
key (candidate pathway) and value (end node of the candidate pathway)
Define all_pathway as an empty array
all_pathway = expanding_path(edges,all_pathway,candidate_pathway)
```

```
Function expanding_path(edges,all_pathway,candidate_pathway)
    Define candidate_pathway_temp array for storing unfinished pathways
    For each candidate pathway and its end node from candidate_pathway
        For each child node of the end node from edges
            If the child node is the native polypeptide
                Put the pathway to all_pathway
            Else
                Concatenating the child node to candidate pathway and
store it in candidate_pathway_temp as key and the child node as value
            Endif
        Endfor
    Endfor
    candidate_pathway=candidate_pathway_temp
    If candidate_pathway is not empty
        call                                                    function
expanding_path(edges,all_pathway,candidate_pathway)
    Else
        return all_pathway
    Endif
Endfunction
```

Step 2: Grouping pathways

The algorithm and pseudo-code is very much the same as that used for identifying shared interactors among the chaperones. The *m* is the size of pathway group. For different *m*, the number of nested For-Endfor loops will be different.

```
Given n pathways from the model as identified in step 1
For m=1 to n
    For i1=1 to n-(m-1)
        For i2=2 to n-(m-2)
            .....
                For im=m to n
                    combine chaperones in pathways 1 to m
                Endfor
            .....
        Endfor
    Endfor
Endfor
```

Step 3: Listing the chaperone modules
In this step, chaperone modules are listed by finding unique set of chaperones in each group from step 2. If a module occurs multiple times, only one is listed. This step is trivial, for example, a single function array_unique() in the programming language PHP is sufficient for this task.

3.2.4. Comparing Inferred Chaperone Modules with Consensus Modules

The purpose of comparing chaperone functional modules obtained from consensus models with those obtained from the new proteomics data is to confirm known modules and predict new ones. The comparison is straightforward in terms of programming. In PHP, one can use the functions array_diff() and array_intersect().

3.3. Establishing Pathway Relationship Between Two Chaperones in Two-Component Functional Modules Using Coexpression and KEGG Pathway Information

Two chaperones in two-component functional modules might either act on target protein along a single pathway or multiple pathways or both (Fig. 1). Genes coding for proteins that function in the same pathway are likely coexpressed (13). Therefore, it is rational to estimate pathway relationships between a pair of chaperones in a functional module based on their gene coexpression data. To this end, gene coexpression results are combined with KEGG pathway information (14). It should be emphasized that this analysis is restricted to two-component chaperone modules.

3.3.1. Measuring Gene Coexpression

The degree of gene coexpression is measured with Pearson correlation coefficient (PCC). PCC is calculated using the formula:

$$\mathrm{PCC} = \frac{\sum XY - \left(\sum X \sum Y / N\right)}{\sqrt{\left(\sum X^2 - \left(\left(\sum X\right)^2 / N\right)\right)\left(\sum Y^2 - \left(\left(\sum Y\right)^2 / N\right)\right)}}. \quad (6)$$

Many software packages such Microsoft Excel and R, implement the calculation. Use of R is potentially more efficient because the calculation can be parallelized on a computer cluster or multicore desktops. This is particularly useful if multiple gene expression datasets are to be analyzed. The R function for calculating PCC is cor(X, Y), where the parameters X and Y are vectors representing the expression values of two genes. N is the size of the vectors.

3.3.2. KEGG Pathway Relationship of Proteins

KEGG (Kyoto Encyclopedia of Genes and Genomes) (14) pathways are downloadable from http://www.genome.jp/kegg/. The pathway list table maps relevant genes to a pathway(s). The pathway relationships between each pair of genes are calculated. These relationships include what we term single pathway (two proteins

involved in one pathway), multiple pathway (two proteins involved in two pathways), and both single and multiple pathways. A pseudocode for this purpose is as follows:

```
Read the pathway list into an array in which the protein is the key and
a 2nd dimensional array containing the pathways it is involved in is the
value
For each protein1
    For each protein2 after protein1 in the array
        Find intersect of the pathway arrays of protein1 and protein2
        If no overlap between the two pathway arrays
            The pathway relationship is single pathway
        Else
            If there is only one pathway in the two list
                The pathway relationship is multiple pathways
            Else
                The pathway relationship is both
            Endif
        Endif
    Endfor
Endfor
```

3.3.3. Combining Gene Coexpression Results with KEGG Pathway Relationship

The coexpression results (PCCs) are binned for all pairs of genes that are involved in the KEGG pathways. PCCs range from –1 to 1, and the bins start from –1 and are incremented 0.1 at a time. For each bin, the number of gene pairs functioning in single pathway, multiple pathways, or both are counted. The probability for each pathway relationship is calculated as the ratio of each number to the sum of three numbers in a bin. Using the expression data from Cho et al. (15) and Gasch et al. (16), it is found that the probability of single pathway relationship increases with PCC, the probability of multiple pathways relationship decreases with increase in PCC, and the probability of involvement in both single and multiple pathways does not show as dramatic a change as for other relationships, as expected. This association between gene coexpression and protein pathway relationships provides a reference for inferring the latter based on the former.

3.3.4. Determining Probability of Pathway Relationship Between Two Chaperones in a Two-Component Functional Module

With the association of pathway relationship and coexpression correlation as a reference, we are able to assign the probability of pathway relationship to chaperone functional modules based on coexpression strength of two chaperones in the module. This involves the following two steps.

Step 1: Assign the probability for each pathway relationship based on gene coexpression data

As previously described, gene expression data are binned based on coexpression strength, and for each bin the probability for each pathway relationship is calculated. Given a pair of chaperones in a functional module together with their coexpression coefficient from an expression dataset, the probabilities of pathway relationships are transferred to this pair of chaperones. If multiple gene expression datasets are available, then the coexpression data is integrated as described in step 2 below.

Step 2: Integration of the information from multiple expression datasets

A data integration method (Hon Nian Chua, National University of Singapore, personal communication) is used for combining the probabilities of pathway relationships from multiple expression datasets. This involves the following equation:

$$P = 1 - \prod_{k \in Du,v} (1 - P(k)), \tag{7}$$

where Du,v is the set of expression data that contains both chaperones, $P(k)$ is the probability that the two chaperones have a particular pathway relationship determined using KEGG pathways as training dataset, and P is the integrated probability that the two chaperones have a particular pathway relationship. Three probabilities are then derived for single, multiple, or single and multiple (both) folding pathways. The final probabilities are calculated by normalization, that is, each probability is divided by the total of the three. The results of such an analysis for our chaperone proteomic data are shown in Fig. 2.

In conclusion, the protocol discussed here describes (1) how to filter large-scale interaction data, (2) how to identify chaperone functional modules from this large-scale interaction data, (3) how to compare modules obtained from the large-scale interaction data with modules derived from a consensus chaperone network, and (4) how to determine the probability of the two chaperones in a two-component chaperone module act along single, multiple, or both single and multiple folding pathways (Figs. 1 and 2). Our approach provides important insights into the organization of chaperone networks inside the cell and provides first hints into how cellular protein folding is regulated by molecular chaperones. Finally, the protocol we describe above can be used for any other biological system for which large-scale interaction data might be available.

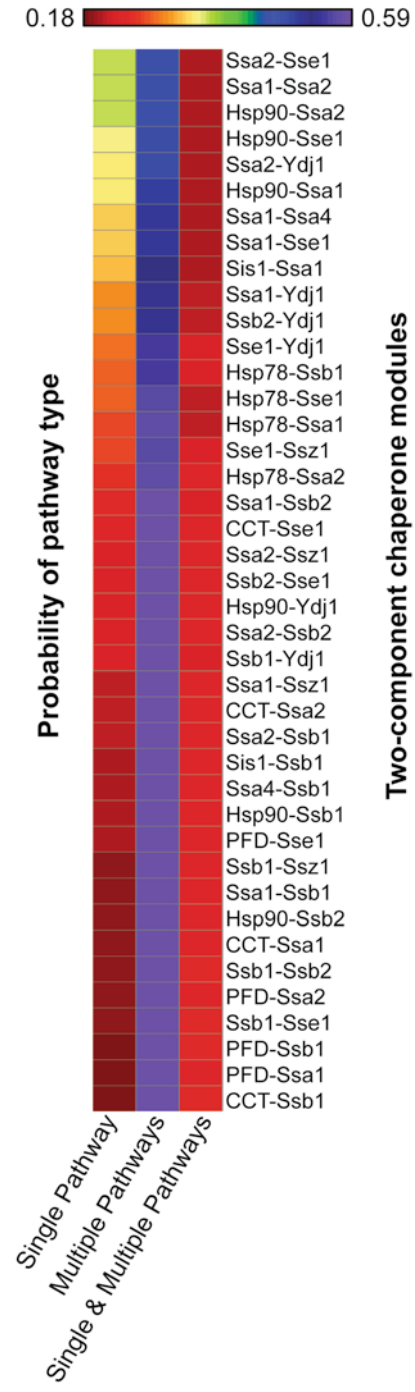

Fig. 2. Pathway relationship of two-component chaperone modules obtained from our yeast chaperone interaction proteomic data (4).

4. Note

1. Cytoscape accepts a variety of data format, from simple sif format, which does not contain node and edge features, such as shape and color, to gml format which does contain node and edge features. A brief but complete gml file is shown below:

```
graph
[
      label ""
      directed    1
      node
      [
            id    0
            label "Hsp70"
            graphics
      [
            x     1916.0
            y     4409.0
            w     20.0
            h     20.0
            type  "ellipse"
            width 1.00000
            fill  "#E1E1E1"
            outline     "#000000"
      ]
]
```

```
node
[
      id    1
      label "Hsp90"
      graphics
      [
            x     2021.0
            y     4333.0
            w     20.0
            h     20.0
            type  "ellipse"
            width 1.00000
            fill  "#FF0000"
            outline     "#000000"
      ]
]
edge
      [
            source      0
            target      1
            label "pp"
            graphics
            [
                  width 2
                  type  "line"
                  fill  "#0000E1"
            ]
      ]
]
```

This gml script describes an edge between two nodes. The node features, namely, label, position, size, shape, and color, and the edge features, namely, width and color, are defined. To create a complex network file with many nodes and edges, one can write a script in a language of his/her choice. The script reads the data file and writes a gml file. For example, if the interactors shared between chaperones are to be visualized, chaperones are depicted as nodes with different colors indicating the chaperone groups, and the edges represent shared interactors colored to indicate the number of shared interactors. The created gml file can then be imported into CytoScape.

Acknowledgments

This work was supported by grants from the Canadian Institutes of Health Research (MOP-81256) to W.A.H., and Genome Canada through the Ontario Institute of Genomics to Z.Z.

References

1. Doyle SM, Wickner S. Hsp104 and ClpB: protein disaggregating machines. Trends Biochem Sci 2009;34:40–8.
2. Hartl FU, Hayer-Hartl M. Converging concepts of protein folding in vitro and in vivo. Nat Struct Mol Biol 2009;16:574–81.
3. Sghaier H, Le Ai TH, Horiike T, Shinozawa T. Molecular chaperones: proposal of a systematic computer-oriented nomenclature and construction of a centralized database. In Silico Biol 2004;4:311–22.
4. Gong Y, Kakihara Y, Krogan N, et al. An atlas of chaperone-protein interactions in *Saccharomyces cerevisiae*: implications to protein folding pathways in the cell. Mol Syst Biol 2009;5:275.
5. Shaner L, Wegele H, Buchner J, Morano KA. The yeast Hsp110 Sse1 functionally interacts with the Hsp70 chaperones Ssa and Ssb. J Biol Chem 2005;280:41262–9.
6. Matsumoto R, Rakwal R, Agrawal GK, et al. Search for novel stress-responsive protein components using a yeast mutant lacking two cytosolic Hsp70 genes, SSA1 and SSA2. Mol Cells 2006;21:381–8.
7. Cline MS, Smoot M, Cerami E, et al. Integration of biological networks and gene expression data using Cytoscape. Nat Protoc 2007; 2: 2366–82.
8. Breitkreutz BJ, Stark C, Reguly T, et al. The BioGRID Interaction Database: 2008 update. Nucleic Acids Res 2008;36:D637-40.
9. Mewes HW, Dietmann S, Frishman D, et al. MIPS: analysis and annotation of genome information in 2007. Nucleic Acids Res 2008; 36:D196-201.
10. Aoki KF, Kanehisa M. Using the KEGG database resource. Curr Protoc Bioinformatics 2005;Chapter 1:Unit 1 12.
11. Young JC, Agashe VR, Siegers K, Hartl FU. Pathways of chaperone-mediated protein folding in the cytosol. Nat Rev Mol Cell Biol 2004; 5:781–91.
12. Knuth DE. The Art of Computer Programming 3ed. Boston: Addison-Wesley; 1997.
13. Adler P, Peterson H, Agius P, Reimand J, Vilo J. Ranking genes by their co-expression to subsets of pathway members. Ann N Y Acad Sci 2009;1158:1–13.
14. Kanehisa M, Araki M, Goto S, et al. KEGG for linking genomes to life and the environment. Nucleic Acids Res 2008;36:D480-4.
15. Cho RJ, Campbell MJ, Winzeler EA, et al. A genome-wide transcriptional analysis of the mitotic cell cycle. Mol Cell 1998;2:65–73.
16. Gasch AP, Spellman PT, Kao CM, et al. Genomic expression programs in the response of yeast cells to environmental changes. Mol Biol Cell 2000;11:4241–57.