overlappingCGM: Automatic detection and analysis of overlapping co-expressed gene modules

Quang-Huy Nguyen[1], and Duc-Hau Le[1,2*]

[1]Department of Computational Biomedicine, Vingroup Big Data Institute, Hanoi, Vietnam.

[2]College of Engineering and Computer Science, VinUniversity, Hanoi, Vietnam.

* corresponding author: Email: hauldhut@gmail.com; Tel: (84)912324564

Keywords: identification of modules; analysis of modules; co-expression; clinical feature association; gene expression

ABSTRACT
When it comes to the co-expressed gene module detection, its typical challenges consist of overlap between identified modules and local co-expression in a subset of biological samples. A recent study have reported that the decomposition methods are the most appropriate ones for solving these challenges. In this study, we represent a R tool, termed overlapping co-expressed gene module (overlappingCGM), which possesses those methods with a wholly automatic analysis framework to help non-technical users to easily perform complicated statistical analyses and then gain robust results. We also develop a novel auxiliary statistical approach to select the optimal number of principle components using a permutation procedure. Two example datasets are used, related to human breast cancer and mouse metabolic syndrome, to enable the illustration of the straightforward use of the tool. Computational experiment results show that overlappingCGM outperforms state-of-the-art techniques. The R scripts used in the study, including all information on the tool and its usage are made publicly available at https://github.com/huynguyen250896/overlappingCGM.

INTRODUCTION
The introduction of genome-wide gene expression profiling technologies observed so far has turned the biological interpretation of large gene expression compendia using module detection methods to be a crucial pillar [1-3]. Here, a module itself is a set of genes which are similarly functioned and jointly expressed. Co-expressed modules do not only help to globally and objectively interpret gene expression data [4, 5], but it is also used to discover regulatory relationships between putative target genes and transcription factors [6-8]. Also, it is useful to study the origin [9] and development [10] of complex diseases caused by many factors.

The nature of module detection is the use of unsupervised clustering approaches and algorithms. Those methods are advanced undoubtedly, but the selection of a certain clustering method for sample- and gene-clustering tasks is separate, in which the latter task is often more complicated. Indeed, users should predetermine the following limitations before applying clustering methods to gene expression. Firstly, not all of clustering methods have the ability to tackle the problem of overlap between modules. Whereas a cluster of patients into biologically distinct subgroups is our ultimate goal, the way to group genes into functional modules need to be more careful since we know that at least five genes work in concert [11] and that their interaction is associated with multiple pathways [12]. Secondly, clustering methods often ignore local co-expression effects which only appear in a subset of all biological samples and instead are interested in co-expression among all samples. This results in loss of meaningful information due to highly context specific of transcriptional regulation [13]. Thirdly, clustering potentially misses the regulatory relationships between genes. As the interpretation of the target expression change is based partly on the change in transcription factor expression [14], this information included may help to improve the ability of module identification. Among existing clustering methods, decomposition methods [15] and biclustering [16] are said to possibly handle the first two restrictions, whereas the last restriction may be solved well by direct network inference [14] and iterative NI [17]. These obviously affect the selection of which clustering method in the context of gene expression; however, it is rarely examined sufficiently, leading to a typical example is the tool weighted gene co-expression network analysis (WGCNA) [18] with a hierarchical agglomerative clustering [19].

Wouter Saelens *et al* [20] have conducted a holistic comparison of module detection methods for gene expression data and realized that the decomposition methods, including independent component analysis (ICA) [21-23], principle component analysis (PCA) [24], and independent principle component analysis (IPCA) [25], are the best. In this study, we have proposed a R tool, named overlappingCGM, which integrates these methods in the hope that it could be a potential alternative to rectify the limitations above. In particular, we apply a state-of-the-art statistical method to specify the optimal number of components in advance required by them and an implementation for module detection and analysis in an automatic manner. These help non-technical users to easily perform complicated statistical analyses and gain robust results in a surprisingly rapid way. We have also demonstrated a better performance of overlappingCGM with other high-tech methods.

MATERIAL AND METHODS
Automatic framework of the overlappingCGM
Figure 1 shows the automatic framework for module detection and analysis included in overlappingCGM. Gene expression matrix first suffered from the two pre-processing steps as excluding outlier individuals and normalization prior to being input of overlappingCGM. The result of normalization was the distribution of each gene expression was centered and standardized across samples. The users now put the data to overlappingCGM and it printed automatically out co-expressed gene modules, corresponding hub-genes in each module, and analysis result of associations between each module and each clinical feature of choice. Note that overlappingCGM decomposed the expression matrix into the product of two or more sub-matrices by only using one of the two methods belonging to the decomposition methods, including ICA (the *fastICA* algorithm) and IPCA (the *ipca* algorithm) (detailed in Supplementary Methods). A module was then determined from a particular component using several optional post-processing steps. Moreover, the R function *daBEST* provided in overlappingCGM aimed not only to help the users to know which method should be selected, but also to specify the optimal number of principle components to extract by ICA or IPCA (detailed in Supplementary Method and FigureS2).

Since the output of the decomposition methods generally consisted of two parts, the components for genes and the components for samples, we, in this study, distinguished the two by the term of signatures and patterns, respectively (Supplementary Methods and FigureS1). When it came to the first matrix product (vertical rectangle in FigureS1), overlappingCGM described the characteristic of different signatures by, between them, a set of genes of which the

overlap was allowed. In contrast, for the second matrix product (horizontal rectangle in FigureS1), overlappingCGM characterized each component by its expression patterns in samples.

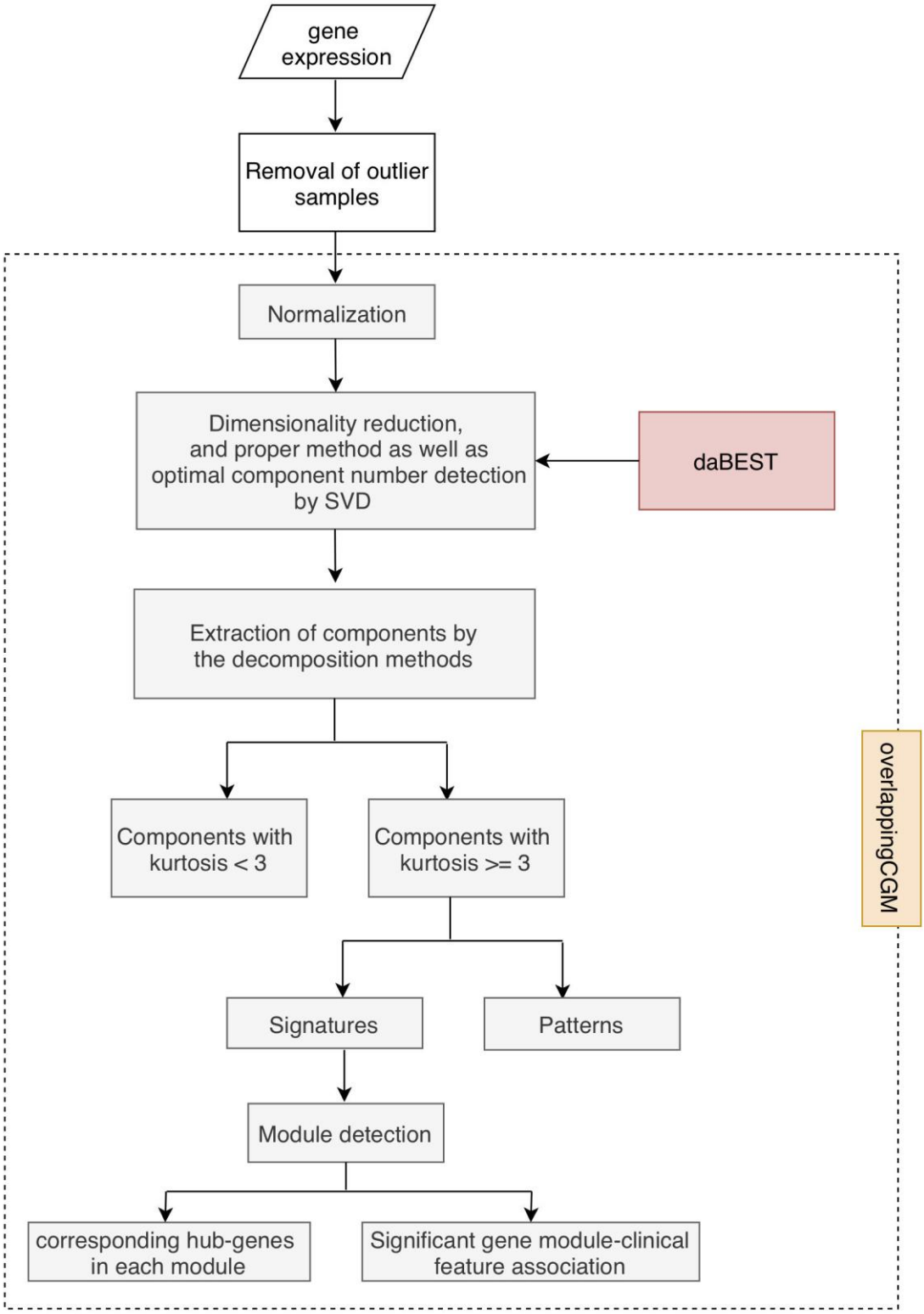

Figure 1. Automatic analysis framework of overlappingCGM. Gene expression data first underwent the two pre-processing steps: removal of outlier samples and normalization. Then, the processed data were inputted into overlappingCGM and rendered co-expressed gene modules, corresponding hub-genes in each module, and analysis result of associations between each module and each clinical feature of choice. Note that the users could refer to the recommendation of overlappingCGM regarding which decomposition method should be selected and how many component numbers were optimal by using the function *daBEST*. Abbreviation: SVD, singular value decomposition.

Removal of signatures with (almost) Gaussian distribution

overlappingCGM equipped with ICA and IPCA required the distributions of the signatures across genes must be as non-Gaussian as possible; ideally, they should be heavy-tailed normal distributions. Due to this requirement, the kurtosis

was recruited, which statistically describes the "tailedness" of the distribution [22], and truncated signatures whose distribution was platykurtic (kurtosis value < 3).

Detection of co-expressed gene modules with corresponding hub-genes

It was evident that a few genes at the tails of a heavy-tailed distribution would be the most important elements in a particular signature, and conversely, the influence of the majority of genes became more and more weak, or even was over, in that signature when they lay at the center of the distribution [22, 23]. Based on this, overlappingCGM provided the users with three optional post-processing steps attached with ICA and IPCA (two for ICA and one for IPCA) to detect co-expressed gene modules with corresponding hub-genes.

For the first option of the post-processing step ("ICA-FDR" assigned to the *method* argument of the function *overlappingCGM*), overlappingCGM did extraction of non-gaussian signatures by ICA (the *fastICA* algorithm was configured using *parallel* extraction method and the default measure of non-Gaussianity *logcosh* approximation of negentropy with $\alpha=1$), then the *fdrtool* R tool [26] modeled those signatures as a two-distribution mixture (null and alternative). The null (Gaussian) distribution was fitted around the median of the signature distribution. At last, an user-defined probability threshold (e.g., 0.1, 0.01, 0.001, …), called tail area-based false discovery rate (FDR), was chosen to distribute genes to modules on the condition that a gene whose FDR lesser than the threshold at a signature was assigned to that signature (module). Here we suggested the selection of the sufficiently stringent threshold of 0.001 if appropriate for robustness.

The second option ("ICA-Zscore" assigned to the *method* argument of the function *overlappingCGM*) was similar to the first one, but overlappingCGM first did z-score transformation for genes in each signature. A gene belonged to a module if the absolute of its z-score was greater than an user-defined standard deviation threshold (e.g., 0.5 σ,1 σ, 1.5 σ,…). We suggested choosing the sufficiently strict threshold of 3 σ on either side from the zero mean, which picks only out a few genes in the tails of the distributions, at any time as possible.

The last option ("IPCA-FDR" assigned to the *method* argument of the function *overlappingCGM*) was similar to the first one, but here overlappingCGM used IPCA (the *ipca* algorithm was configured using *deflation* extraction method and the default measure of non-Gaussianity *logcosh* approximation of negentropy with $\alpha=1$) instead of ICA. This algorithm was more robust to noise.

Genes at both extremes of the distribution were considered as hub-genes. The Pearson's correlations of each resulting co-expressed module to each clinical feature of interest were then calculated based on correlations between the patterns and those clinical features.

Performance validation of overlappingCGM

Gene expression data

We used two example data, human breast cancer [27] and mouse metabolic syndrome [28], to illustrate the straightforward use of overlappingCGM as well as be convenient for comparing its ability with other tools. In particular, the first case study, downloaded from the cBioPortal for Cancer Genomics (http://www.cbioportal.org) [29, 30], was the METABRIC breast cancer cohort in the United Kingdom and Canada. The gene expression data were generated using the Illumina Human v3 microarray for 1,904 samples. The second case study, related to mouse metabolic syndrome (obesity, insulin resistance, and dyslipidemia), was liver gene expressions from 134 female mice including 3600 physiologically relevant genes. The data were employed by the authors of WGCNA to indicate how to use this tool.

Comparison of overlappingCGM with WGCNA and its improved version

We used the two expression data above to validate the performance of overlappingCGM with WGCNA and an improved version of WGCNA proposed by us [31], temporarily called improved WGCNA (iWGCNA) in this study. For WGCNA, we applied it to the gene expression data using the *blockwiseModules* function (v1.69). All tuning parameters were left as defaut. For iWGCNA, its improvement was that we added an additional step to the gene clustering process, the determination of the optimal cluster agglomeration method for each particular case. All other tuning parameters were set to their default value, except for the selection of the soft-thresholding value [18].

To compare the power of them, we estimated the pairwise Pearson's correlation coefficients, *r*, between module eigengenes (MEs, characterized by its first principal component) of resulting modules given by WGCNA and iWGCNA versus patterns given by overlappingCGM. This helped us to determine which modules could be missed by WGCNA and iWGCNA. Then, g:Profiler (https://biit.cs.ut.ee/gprofiler/gost) (ver *e102_eg49_p15_7a9b4d6*; accessed on 20 Feb. 2021) [32] verified biological processes and KEGG pathways related to those missed modules. Biological processes and KEGG pathways with adjusted P-values ≤ 0.05 (G:SCS multiple testing correction method [32], two-tailed) were considered to be statistically significant.

RESULTS

In our previous study [31], the breast cancer data were used to detect validated 31 breast-cancer-associated genes, and we then did a cluster of those genes to functional modules using iWGCNA. Here, we revisited the results to be convenient for the comparison. Due to the small number of genes, WGCNA failed to identify any co-expressions across the 1904 breast cancer patients (the 31 genes were in the gray module), while iWGCNA and overlappingCGM indicated two and three modules, respectively. These implied that the ability of iWGCNA and overlappingCGM was better than WGCNA in the co-expressed gene module identification. We further investigated the power of the three methods by estimating the pairwise Pearson's correlation coefficients between the one and two MEs given by WGCNA and iWGCNA, respectively, versus the three patterns given by overlappingCGM as described in the Material and Methods

section. As expected, one WGCNA ME (100.0%) and one iWGCNA ME (50.0%) respectively showed $r > 0.4$ with at least one overlappingCGM pattern, whereas only 33.3% of overlappingCGM patterns correlated to at least one ME obtained by both WGCNA and iWGCNA with the same intensity (Supplementary TableS1). TableS1 shows that both WGCNA and iWGCNA potentially missed two color-coded modules turquoise and brown ($r < 0.4$). We functionally enriched the two and realized that they possessed an overlapping set of genes significantly associated with numerous biological processes and pathways (Supplementary TableS2), suggesting that overlappingCGM was most likely to identify biologically relevant modules that were not represented by WGCNA or iWGCNA modules. Figure 2 illustrates the analysis results of the three identified modules by overlappingCGM highlighted as turquoise, blue, and brown. The brown module was positively significantly correlated with all three clinical features, including the number of lymph nodes, Nottingham prognostic index, and tumor stages of the breast cancer patients. In addition, the turquoise module showed a significant negative association with the Nottingham prognostic index only. Besides, overlappingCGM also reported the top 10 hub-genes in each of these modules, including *KMT2C, BAP1, PTEN, NF1, RUNX1, ZFP36L1, CDKN1B, BRCA2, MAP3K1,* and *PIK3CA* in the turquoise module; *CDH1, PIK3R1, GATA3, CDKN2A, TBX3, SMAD4, KRAS, RB1, MEN1,* and *RUNX1* in the blue module; and *KRAS, GPS2, SF3B1, AGTR2, RB1, NCOR1, SMAD4, ERBB3, FOXO3,* and *NF1* in the brown module.

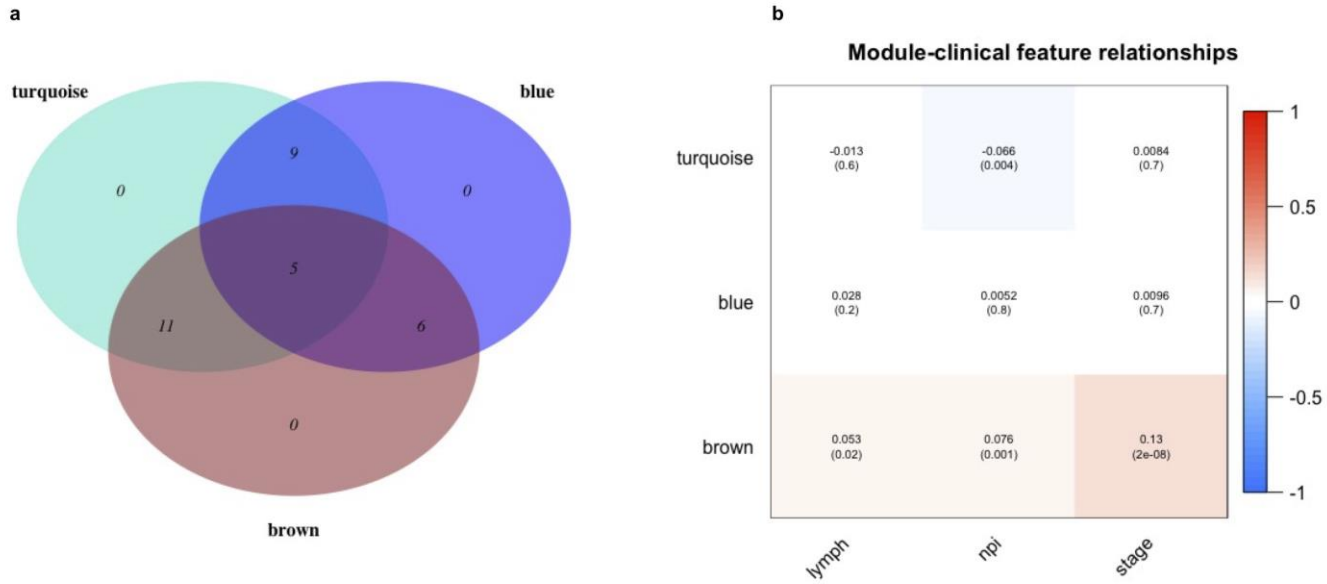

Figure 2. Analysis results of identified modules by overlappingCGM. (a) Venn diagram shows the overlap of the 31 driver genes between the three functional modules (turquoise, blue, and brown). (b) Associations of each pattern with each of three clinical features of interest. Abbreviation: lymph, the number of lymph nodes; npi, Nottingham prognostic index; stage, tumor stages of all the breast cancer patients.

Similarly, we again applied the three tools to 2281 gene expressions in liver of the 134 female mice. As a result, WGCNA, iWGCNA, and overlappingCGM detected 17, 12, and 18 modules, respectively. In this turn, four out of 17 WGCNA MEs (23.5%) and four out of 12 iWGCNA MEs (33.3%) respectively yielded $r > 0.8$ with at least one overlappingCGM pattern. In contrast, those numbers for overlappingCGM were three out of 18 patterns (16.7%) and four out of 18 patterns (22.2%) related to at least one WGCNA MEs and one iWGCNA MEs with the same intensity, in which WGCNA and iWGCNA could ignore 15 and 14 important color-coded modules ($r < 0.8$), respectively (Supplementary TableS3). We analyzed enrichment on those missed modules, rendering all of them associated significantly with biological processes and pathways. More details of the pre-processing procedures, analysis processes, and comparisons were shown in the Supplementary Results.

DISCUSSION

Co-expressed gene module identification or sample clustering rely mostly on unsupervised clustering methods, resulting in the development of new tools or new analysis frameworks [18, 31, 33, 34]. However, module detection is unique due to necessity of ensuring biological reality in the context of gene expression, such as overlap and local co-expression. In this study, we therefore have presented a new tool, overlappingCGM, for module discovery; especially, it differentiates from other advanced methods on the ability to identify different modules which allow having the overlap between them, better reflecting biological reality than methods that stratify genes into separate subgroups. The fact that overlappingCGM outperforms the well-known tools, such as WGCNA or iWGCNA, in identifying functional modules of genes. Moreover, overlappingCGM is sufficiently flexible to be applied to any organisms, like human, mouse, yeast, etc. In addition, overlappingCGM is well able to automatically and easily do the two tasks as identification and analysis of modules. These clearly help to support a community of users with diverse backgrounds, such as biologists, bioinformaticians, and bioinformaticists, who are interested in this field.

When using the decomposition methods, the selection of the optimal number of principle components is vital. Here we also introduce daBEST that performs a permutation procedure (see Supplementary Methods and FigureS2) in the hope that the extracted components are generated not-at-random. Based on the above two examples, we can realize that the number of modules indicated by overlappingCGM is approximately equal to these by WGCNA and iWGCNA. Although further studies are required, these results imply that daBEST has the potential to provide a suggestion having high value of reference before using the decomposition methods.

In conclusion, we believe that overlappingCGM software may be useful, not only to improve module detection, but also to discover novel biological insights in complex diseases.

SOFTWARE AND DATA AVAILABILITY

R package of overlappingCGM and the raw data used in the study are available on GitHub (https://github.com/huynguyen250896/overlappingCGM), respectively. Approval by a local ethics committee was not required, and all the data can be immediately downloaded.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR online.

CONFLICT OF INTEREST

We have no conflicts of interest to disclose.

REFERENCES

1. Eisen, M.B., et al., *Cluster analysis and display of genome-wide expression patterns.* Proceedings of the National Academy of Sciences, 1998. **95**(25): p. 14863-14868.
2. D'haeseleer, P., *How does gene expression clustering work?* Nature biotechnology, 2005. **23**(12): p. 1499-1501.
3. Chaussabel, D. and N. Baldwin, *Democratizing systems immunology with modular transcriptional repertoire analyses.* Nature Reviews Immunology, 2014. **14**(4): p. 271-280.
4. Voineagu, I., et al., *Transcriptomic analysis of autistic brain reveals convergent molecular pathology.* Nature, 2011. **474**(7351): p. 380-384.
5. Jostins, L., et al., *Host–microbe interactions have shaped the genetic architecture of inflammatory bowel disease.* Nature, 2012. **491**(7422): p. 119-124.
6. Yosef, N., et al., *Dynamic regulatory network controlling TH 17 cell differentiation.* Nature, 2013. **496**(7446): p. 461-468.
7. Jojic, V., et al., *Identification of transcriptional regulators in the mouse immune system.* Nature immunology, 2013. **14**(6): p. 633-643.
8. Paul, F., et al., *Erratum: Transcriptional Heterogeneity and Lineage Commitment in Myeloid Progenitors (Cell (2015) 163 (1663-1677)).* Cell, 2016. **164**(1-2).
9. Alsina, L., et al., *A narrow repertoire of transcriptional modules responsive to pyogenic bacteria is impaired in patients carrying loss-of-function mutations in MYD88 or IRAK4.* Nature immunology, 2014. **15**(12): p. 1134-1142.
10. Chaussabel, D., et al., *A modular analysis framework for blood genomics studies: application to systemic lupus erythematosus.* Immunity, 2008. **29**(1): p. 150-164.
11. Arnone, M.I. and E.H. Davidson, *The hardwiring of development: organization and function of genomic regulatory systems.* Development, 1997. **124**(10): p. 1851-64.
12. Miklos, G.L. and G.M. Rubin, *The role of the genome project in determining gene function: insights from model organisms.* Cell, 1996. **86**(4): p. 521-9.
13. Neph, S., et al., *Circuitry and dynamics of human transcription factor regulatory networks.* Cell, 2012. **150**(6): p. 1274-1286.
14. Marbach, D., et al., *Wisdom of crowds for robust gene network inference.* Nature methods, 2012. **9**(8): p. 796-804.
15. Rotival, M., et al., *Integrating genome-wide genetic variations and monocyte expression data reveals trans-regulated gene modules in humans.* PLoS Genet, 2011. **7**(12): p. e1002367.

16. Eren, K., et al., *A comparative analysis of biclustering algorithms for gene expression data.* Briefings in bioinformatics, 2013. **14**(3): p. 279-292.
17. Roy, S., et al., *Integrated module and gene-specific regulatory inference implicates upstream signaling networks.* PLoS Comput Biol, 2013. **9**(10): p. e1003252.
18. Langfelder, P. and S. Horvath, *WGCNA: an R package for weighted correlation network analysis.* BMC Bioinformatics, 2008. **9**(1): p. 559.
19. Lance, G.N. and W.T. Williams, *A General Theory of Classificatory Sorting Strategies: 1. Hierarchical Systems.* The Computer Journal, 1967. **9**(4): p. 373-380.
20. Saelens, W., R. Cannoodt, and Y. Saeys, *A comprehensive evaluation of module detection methods for gene expression data.* Nature Communications, 2018. **9**(1): p. 1090.
21. Comon, P., *Independent component analysis, a new concept?* Signal processing, 1994. **36**(3): p. 287-314.
22. Hyvärinen, A. and E. Oja, *Independent component analysis: algorithms and applications.* Neural networks, 2000. **13**(4-5): p. 411-430.
23. Liebermeister, W., *Linear modes of gene expression determined by independent component analysis.* Bioinformatics, 2002. **18**(1): p. 51-60.
24. Jolliffe, I.T. and J. Cadima, *Principal component analysis: a review and recent developments.* Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 2016. **374**(2065): p. 20150202.
25. Yao, F., J. Coquery, and K.-A. Lê Cao, *Independent Principal Component Analysis for biologically meaningful dimension reduction of large biological data sets.* BMC Bioinformatics, 2012. **13**(1): p. 24.
26. Strimmer, K., *A unified approach to false discovery rate estimation.* BMC bioinformatics, 2008. **9**(1): p. 1-14.
27. Pereira, B., et al., *The somatic mutation profiles of 2,433 breast cancers refine their genomic and transcriptomic landscapes.* Nature Communications, 2016. **7**(1): p. 11479.
28. Ghazalpour, A., et al., *Integrating Genetic and Network Analysis to Characterize Genes Related to Mouse Weight.* PLOS Genetics, 2006. **2**(8): p. e130.
29. Cerami, E., et al., *The cBio Cancer Genomics Portal: An Open Platform for Exploring Multidimensional Cancer Genomics Data.* Cancer Discovery, 2012. **2**(5): p. 401-404.
30. Gao, J., et al., *Integrative analysis of complex cancer genomics and clinical profiles using the cBioPortal.* Sci Signal, 2013. **6**(269): p. pl1.
31. Nguyen, Q.-H. and D.-H. Le, *Improving existing analysis pipeline to identify and analyze cancer driver genes using multi-omics data.* Scientific Reports, 2020. **10**(1): p. 20521.
32. Raudvere, U., et al., *g:Profiler: a web server for functional enrichment analysis and conversions of gene lists (2019 update).* Nucleic Acids Research, 2019. **47**(W1): p. W191-W198.
33. Nguyen, Q.-H., et al., *Multi-omics analysis detects novel prognostic subgroups of breast cancer.* Frontiers in Genetics, 2020.
34. Nguyen, H., et al. *Disease subtyping using community detection from consensus networks*. in *2020 12th International Conference on Knowledge and Systems Engineering (KSE)*. 2020.

FIGURES LEGENDS

Figure 1. Automatic analysis framework of overlappingCGM. Gene expression data first underwent the two pre-processing steps: removal of outlier samples and normalization. Then, the processed data were inputted into overlappingCGM and rendered co-expressed gene modules, corresponding hub-genes in each module, and analysis result of associations between each module and each clinical feature of choice. Note that the users could refer to the recommendation of overlappingCGM regarding which decomposition method should be selected and how many component numbers were optimal by using the function *daBEST*. Abbreviation: SVD, singular value decomposition.

Figure 2. Analysis results of identified modules by overlappingCGM. (a) Venn diagram shows the overlap of the 31 driver genes between the three functional modules (turquoise, blue, and brown). (b) Associations of each patterns with each of three clinical features of interest. Abbreviation: lymph, the number of lymph nodes; npi, Nottingham prognostic index; stage, tumor stages of all the breast cancer patients.