Dear Professor Ziv Bar-Joseph,

Enclosed please find our substantially revised manuscript “**QUBIC: a Bioconductor package for qualitative biclustering analysis of gene co-expression data**”. In this revised manuscript, we have carefully addressed all the concerns by the two reviewers. We greatly appreciate the Referee’s comments on the previous draft of the paper and we hope that you and your reviewers find the revised version acceptable for publication in *Bioinformatics*. The following is our point-by-point response to each of the criticisms by the two reviewers. I would like to take this opportunity to thank you for handling the review of our paper.

*Our responses to the review comments are in blue and italic.*

Sincerely yours,

Qin Ma, Ph.D.

**Concern of Associate Editor:** The reviewers raised several concerns, but we decided to go forward and allow a revision because of the popularity of the software. However, if you decide to revise, keep in mind that the bar for an Application Note is rather high and that you would need to convince the reviewers that the changes made indeed helped beyond just porting to a new language.

***Response:*** *Thanks for allowing us work on a resubmission of this manuscript. We have substantially revised the previous manuscript and made significant efforts in responding to the review comments/concerns. QUBICR is not just porting to a new language. Additional computational functions have been added in the R package, and more biological datasets have been used to evaluate program efficiency. More details could be found below.*

**Reviewer #1**

**General comments**: Zhang et.al describe in this application note a new package for R for performing bicluster analysis. They made the QUBIC program, originally implemented in C, more accessible to the community by distributing it as an R-system’s package. The manuscript is straightforward to read and the required information to obtain and use their package is clearly exposed. Despite the authors made a good job in describing the capabilities of QUBIC, they fail to compare it to biclust, a currently available package for R from the CRAN repository. In fact, it seems the QUBIC package partially behaves as an add-on for the biclust package, adding the capability to biclust of performing bicluster analysis using the QUBIC algorithm.

***Response:*** *Thanks for your comments. We have carried out a systematic comparison between QUBIC and other six packages in* ***biclust****. The comparison results and more unique features of QUBIC R package can be found in* ***Figure 1*** *and the corresponding text in the revised manuscript.*

**Concern 1**: It will be very valuable for the readers to have a clear comparison of QUBIC-R vs. the other R-system package for performing bicluster analysis: biclust. Which are the differences of QUBIC vs. biclust. What is possible to do with QUBIC that biclust cannot, etc.

***Response:*** *Thank you for this comment. We have comprehensively compared QUBIC-R and biclust, with summarized results shown in* ***Figure1A****. Generally, QUBIC-R provides some unique functions that biclust does not provide, e.g., network visualization, query-based biclustering, bicluster expanding, and biclusters comparison. Meanwhile, the prediction performance of QUBIC and some other programs in biclust was also reviewed and evaluated in [1]. Based on that, some recommendations in support of various biclustering applications were also summarized in* ***Figure 1A****. The descriptions of newly added functions in QUBIC-R are summarized as following.*

1. *Query-based biclustering. A user can input additional biological information between any gene pairs, e.g. co-regulation and protein-protein interaction. QUBIC-R will utilize that information to guide the biclustering progress (newly-added parameter “weight”). This kind of function is so-called* ***query-based biclustering*** *and has been widely applied in bioinformatics [2-4]. Further details and an example of this function can be found in supplementary Method S1 and Example S2.4, respectively.*
2. *Bicluster expanding****.*** *Through the newly-defined parameter “seedbicluster”, a user can conduct the* ***bicluster expanding*** *function. It expands existing biclusters by recruiting more genes according to specified consistency level. The existing biclusters can be any biclustering results obtained from QUBIC-R or other algorithms in the biclust package. This function has been successfully applied in our previous study [5] and a flowchart of this function can be found in Figure 1 of [5]. Further details can be found in supplementary Method S2, and supplementary Example S2.5 showcases how to carry out this analysis in QUBIC-R.*
3. *Biclusters comparison****.*** *The “showinfo” parameter allows a user to compare the biclustering results obtained from different algorithms, e.g., QUBIC or another algorithm in biclust, or from the same algorithms with different combinations of parameters. Specifically, it will show 13 categories of comparison, including the number of detected biclusters, the overlap of first two biclusters, the maximum row number in all identified biclusters, etc. Example of this function can be found in supplementary Example S2.6.*

**Concern 2**: In the abstract, second sentence, the authors express: “The program, QUBIC, is recognized as one of the best biclustering methods in terms of its efficiency and effectiveness in biological data interpretation”. It would be informative to the readers to add "However, its availability is limited to a C implementation and to a low-throughput web interphase.”.

***Response:*** *Thanks for your comments. We have made the suggested changes.*

**Concern 3**: The first sentence of the introduction is not clear. Do the authors mean that “Advances in high-throughput technologies have made possible the generation of massive quantities of gene expression data. Such data revolution is being only partially paralleled by the development of new algorithms for its interpretation.”

***Response:*** *Thanks for your comment. We have rewritten this sentence as suggested.*

**Concern 4**: I would consider removing the sentence “It is noteworthy that our in-house…” I don’t think is relevant for the reader know that their method have 100 citations. However, knowing that it is among the best in performance tests is useful, as expressed in the next sentence.

***Response:*** *Thanks for your comment. We have made the suggested changes.*

**Concern 5**: Point (ii) and (iii) in page 2: the sentence is not clear, do the authors mean that QUBIC can be used as the biclustering method by the biclust function from package “biclust”?

***Response:*** *Thanks for this comment. We have rewritten the sentence as suggested. The biclustering algorithms in QUBIC, i.e. BCQU and BCQUD, can be called independently. Meanwhile, they can also be called as biclustering functions in the “biclust” package. QUBIC provides compatibility for the biclust package to facilitate more users, especially those that are very familiar with “biclust”.*

**Concern 6**: In conclusions, the sentence “… and has become a more important bioinformatics analysis method in this big data era” seems to not have sense. A more important that what? why not simply saying “and has become a useful approach for the interpretation of gene expression profile data.”?

***Response:*** *Thanks for your comment. We have made the suggested changes.*

**Concern 7**: QUBIC is in the development repository of Bioconductor, I think it is important to mention this in the manuscript.

***Response:*** *Thanks for your comment. QUBIC is currently in the release repository of Bioconductor, and we have added this point in the abstract.*

**Reviewer #2**

**No comments.**

**Reviewer #3**

**General comments**: Paper extends existing QUBIC algorithm to R language. However, we doubt about the originality of this work. R can provide heatmap, graph visualizations, network analysis etc. by default and we can find many related R language codes around. Also, what authors intend to do is to speed up already existing code of QUBIC. Here we have an objection, there will be always a better code of an algorithm. What they do is just optimized implementation. Providing algorithm in different programming language is not the case. If we accept this paper, what if some other researcher says "we found better implementation faster than QUBIC-R". So, we expect to see more accurate results. What makes QUBIC-R faster has any effect on original algorithm.

***Response:*** *Thanks for your comments. The core algorithm of biclustering in this R package is same to the original QUBIC algorithm. As we mentioned in the manuscript, QUBIC’s availability is limited to a C implementation and to a low-throughput web interface. Hence, we designed this R package, which (1) is more efficient than QUBIC C program and other R packages in biclust; and (2) has more comprehensive biclustering-related functions. In this revised version, we add three new computational functions to enhance the novelty and originality of our work. Hence, the R package of QUBIC has advanced power in its availability and application. Detailed information of the added new functions can be found in the revised manuscript and the response to Concern 1 of Reviewer #1.*

**Concern 1**: Authors must provide software limits, we do not understand that part as well "the QUBIC source code is optimized and converted from GNU C to C++, thus has better memory control and is more efficient than the one published in 2009. For three large-scale gene expression datasets (E. coli, Arabidopsis and Human tumor), QUBIC-R can averagely save 44% of CPU running time compared with the original program (more details can be found in Table S1);" How the code has changed? Does it really work as original algorithm? What is the maximum size of data that researcher can input? It is better if authors can demonstrate running time analysis on artificial data. Use 100x100, 200x200, ...., 1000x1000 matrices and as a suggestion provide run time analysis with different bicluster ratios. Sometimes, code can run faster on some inputs, but let's see how it works for variety of inputs. You can use real data as well from Gene Expression Omnibus.

***Response:*** *Thank you for the comments.*

*First, regarding the limits of QUBIC-R, (1) it depends on R version 3.2 or higher and it is based on C ++11, it does not support some old platform like Mac OS X 10.9 or older versions; (2) we also tested the maximum size of input data using synthetic datasets with different bicluster ratios. We found that (1) a dataset as large as 30,000*x*30,000 can be finished in 30 minutes; and (2) different bicluster ratios almost have no impact on the running time (more details can be found in supplementary Figure S1). In addition, five large-scale gene expression datasets (in E. coli, Grape, Arabidopsis, Switchgrass, and Human) were used to test the package compared to other R packages in biclust, the running time details can be found in Figure1B.*

*Second, the optimized code can get exactly the same biclustering results of original algorithm, hence keeps the performance advantages of QUBIC as reviewed in [1]. Furthermore, QUBIC-R is much more efficient than QUBIC, and some explanations of the source code optimization are shown in the following. The original QUBIC program was written in GNU C and it can be compiled/run under UNIX only.* *The legacy code works but refactoring was needed in order to put it into a form we can understand and work with to extend its functionality. The C++ part of QUBIC-R can be compiled by any compliers that follow C++ 11 standard, making it possible to run on Windows platform as well. In original QUBIC, memory was allocated via pointers, making it difficult to debug or read, especially for multiple developers in a team development. In QUBIC-R, STL containers were used instead, making the debugging much easier and avoiding possible undetected memory leak. OpenMP (Open Multi-Processing) was used for parallelization, in support of the significant speedup. In summary, the main changes are as follows: GNU C was converted to C++, C pointer was replaced by STL container, source code was refactored to improve readability and maintainability, and OpenMP was used to speed up processing.*

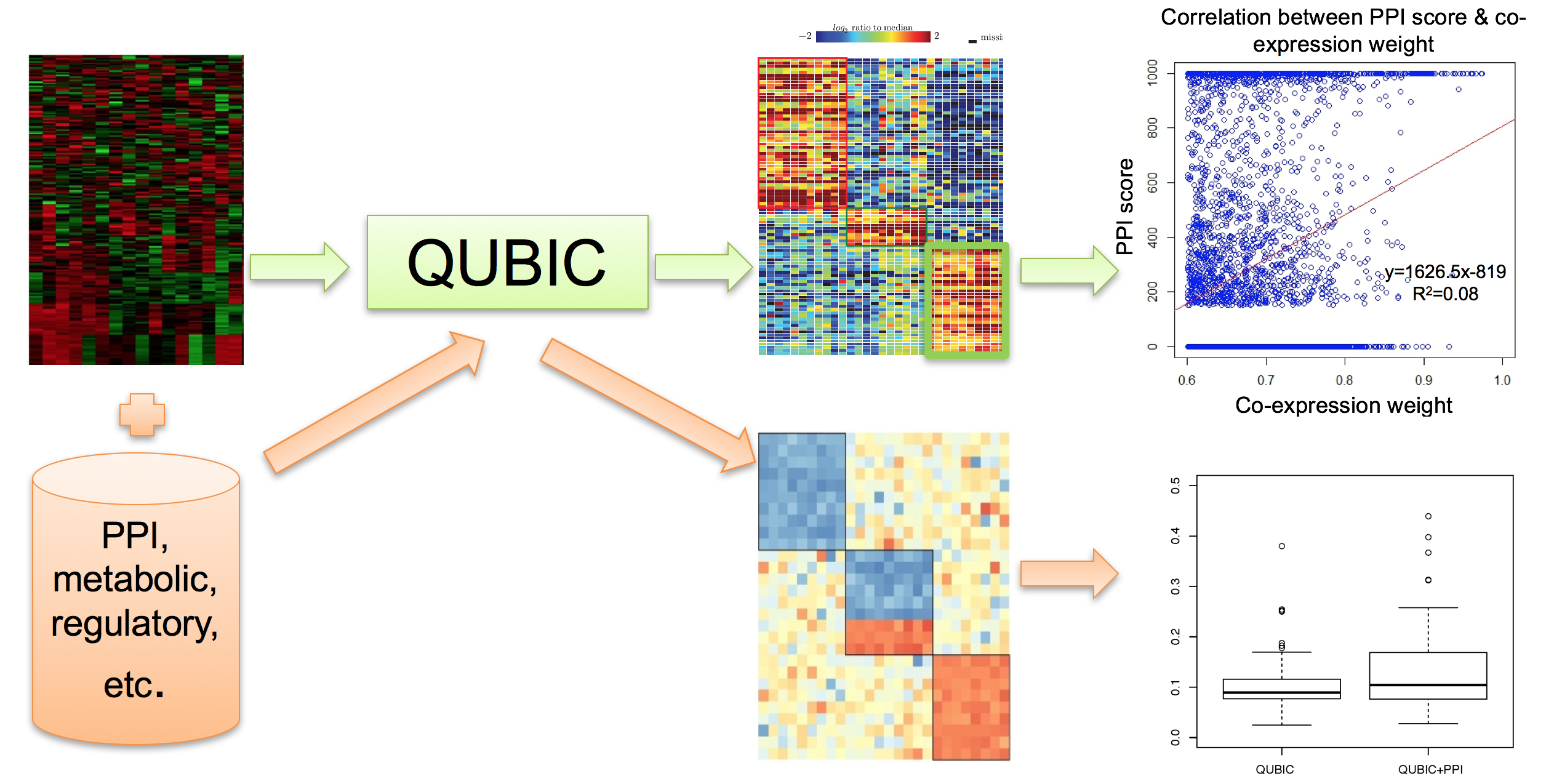
**Concern 2**: It is better to analyze co-expression network using real PPI network data as well. From biclusters, they find co-expressed results, but what is the meaning of such relations compared to vivo PPI network data? They can use weighted PPI networks to correlate co-expression edges and PPI weighted edges. The problem is, assume that as a biologist, how can we trust this tool. To trust, they should show more results compared to vivo data if it is possible.

***Response:*** *We appreciate this comment very much as it inspires us to add the* ***query-based biclustering*** *function in current package and makes the updated package better. Regarding this PPI-related concern, we first conducted a correlation analysis between co-expression weights and PPI network data for each of the identified biclusters in E. coli, and the results showed that their correlations were very week. Actually, Li et al. evaluated the biclusters in terms of their functional enrichments and the biclusters are generally found enriched with GO biological processes, KEGG pathways and regulatory pathways (from RegulonDB) simultaneously [6]. It indicates that multiple biological processes at different levels (metabolic, regulatory processes, etc.) can affect observed gene expression and co-expression patterns among multiple genes (e.g. biclusters). Hence, it is difficult to find good correlation between gene co-expression and one of these biological processes, e.g. PPI.*

*To serve a user with specific prior biological information, we provide a function in QUBIC-R, that allows users to add additional biological information as input (e.g. PPI). QUBIC-R will prioritize output biclusters more related to that specific biological aspect. Specifically, we conducted a default biclustering in QUBIC-R on the E. coli data (parameters: c = 0.95, r = 1, q = 0.06, o = 50, k =23), giving rise to 50 biclusters. Then we got another 50 biclusters through a query-based biclustering in QUBIC-R on the same data along with given PPI scores of E. coli (all the other parameters are same). We then assessed the goodness of fit between PPI scores and co-expression weights (i.e., Spearman Correlation) of gene pairs in each of the 100 biclusters. The generated R-square values are shown in a boxplot in the following flowchart. The Wilcoxon text p-value for the R-square values of the two sets of biclusters (default and query-based) is 0.06 (<0.1), indicating that the query-based biclusters are more enriched with PPI-related functions than those default biclusters.*

*Overall, we believe that the relationship between observed gene expression and various biological processes is very complex, and a manuscript is in preparation for a systematic study on how to integrate additional biological information in our biclustering algorithm. Meanwhile our response is also summarized in the following flowchart.*

*P.S. – the following flowchart and above preliminary results are not integrated into current manuscript due to the two-page limitation.*



***Figure Legend****: Comparison of default biclustering and query-based biclustering in QUBIC-R. Green arrows denote the process of default biclustering in QUBIC-R: (i) input gene expression data, (2) carry out biclustering, and (3) visualize identified biclusters via heatmap. These biclusters have weak correlation with documented PPI information. The PPI and Spearman correlation of gene pairs in a specific bicluster were plotted in the top right figure. Orange arrows show the process of query-based biclustering in QUBIC-R: (1) input gene expression data and additional biological information, (2) carry out the query-based biclustering, and (3) visualize biclusters. Overall, these biclusters give rise to a stronger correlation with PPI (see the bottom-right boxplot).*

**Concern 3**: If this is a software and it is visual software, let us to see video demonstration of the package. What if bicluster sizes increased in visualizations? Some algorithms can produce many biclusters, how they can use this tool visualization for their results? We see that in heatmap code there are only 11 colors to visualize. Is that the standard?

*Response: Thanks for your comments. First, a video demonstration has been prepared and can be accessed at* [*https://github.com/zy26/QUBIC/releases/download/0.99.0/DEMO.wmv*](https://github.com/zy26/QUBIC/releases/download/0.99.0/DEMO.wmv)*. Second, QUBIC-R can correctly draw heatmap either for a single bicluster or for two biclusters. It can also show the correct heatmap for more than two biclusters as long as there are no overlaps among them. It should be noted that this situation (no overlaps) is usually not the case for real data, thus we just show the heatmap for one or two bicluster(s). Finally, in our examples we use 11 colors to visualize the heatmap (default setting). Actually the quheatmap function has the color argument, a user can define their own palette and pass it to the color argument (Supplementary Example S3).*

**Concern 4**: As a researcher, I want to compare QUBIC with my XYZ algorithm. Why should we use that package is a good question. Authors shall already notice that, researchers will tend to cite old paper although you have better running times. How can you force users to use that package is a good question.

Also it is sometimes unfair to compare just running time of algorithms, implemented with different languages. We don't expect to see faster matlab code than C as an example. So our suggestion is to embed more algorithms. For example, on co-expression analysis there is a tool named Robinviz (Aladag et al 2011, Bioinformatics). They already provided tool for visualizations and implemented other algorithms including their new method. Co-expression, co-localization, co-function are their graph constructions methods as well. Indeed, I strongly suggest embed other algorithms to allow comparisons. Authors can also refer to the success of BiCat tool in that area. They are successful because they provide software for the first time and allow to use more than one algorithm.

***Response:*** *Thanks for the comments. Compared to the original C version of QUBIC, QUBIC-R provides more comprehensive functions, e.g., heatmap visualization, co-expression network visualization, query-based biclustering, bicluster results comparison, etc. Regarding to performance, Eren et al.[1] conducted a comparative analysis of biclustering algorithms, in which they evaluated the performance of 12 algorithms (including QUBIC). It was found that QUBIC (1) handles noise much better than other algorithms that seek local patterns, (2) was unaffected by the number of biclusters, (3) is the choice of constant-upregulated biclusters; and (4) is the one that have the highest enriched bicluster ratios in real data sets (The performance comparisons is summarized in Figure1A).*

*The R package biclust already embeds six biclustering algorithms and QUBIC-R is compatible with biclust, which means that after loading the QUBIC-R, users can call seven biclustering algorithms. In this updated version of R package, a bicluster comparison function is added. With this function, users can compare the biclusters generated by different biclustering algorithms in biclust or different parameter combinations of a same algorithm. Specifically, the number of biclusters detected, overlap of the first two biclusters, etc., will be summarized in a table. More details can be found in supplementary Example S2.6.*

**Concern 5.** As we said above, there are some tools about this area. Authors shall show originality of their software among these. The speedup is not enough to be original. Please show some other features. Heatmaps, network visualizations, co-expression networks, those all have been done before.

***Response:*** *Thanks for the comment. Besides heatmap, network visualizations, QUBIC-R also provides some unique functions, e.g., query-based biclustering, bicluster expanding, and biclusters comparison. Some examples of these new functions can be found in Supplementary Example S2.4, S2.5 and S2.6, respectively.*

**REFERENCES**

1. Eren, K., et al., *A comparative analysis of biclustering algorithms for gene expression data.* Briefings in bioinformatics, 2013. **14**(3): p. 279-292.

2. Dhollander, T., et al., *Query-driven module discovery in microarray data.* Bioinformatics, 2007. **23**(19): p. 2573-2580.

3. De Smet, R. and K. Marchal, *An ensemble biclustering approach for querying gene expression compendia with experimental lists.* Bioinformatics, 2011. **27**(14): p. 1948-1956.

4. Alqadah, F., et al. *Query-based Biclustering using Formal Concept Analysis*. in *SDM*. 2012. SIAM.

5. Wang, S., et al., *Genome-scale identification of cell-wall related genes in Arabidopsis based on co-expression network analysis.* BMC Plant Biology, 2012. **12**(1): p. 1-12.

6. Li, G., et al., *QUBIC: a qualitative biclustering algorithm for analyses of gene expression data.* Nucleic Acids Research, 2009. **37**(15): p. e101.