Author's response to reviews

Title: Genomic Analyses with Biofilter 2.0: Knowledge Driven Filtering, Annotation, and Model Development

Authors:

Sarah A. Pendergrass (sap29@psu.edu)
Alex Frase (alex.frase@psu.edu)
John Wallace (jrw32@psu.edu)
Daniel Wolfe (wolfe@psu.edu)
Neerja Katiyar (nvk5095@psu.edu)
Carrie B. Moore (ccb12@psu.edu)
Marylyn D. Ritchie (marylyn.ritchie@psu.edu)

Version: 3
Date: 26 November 2013

Author's response to reviews: see over
November 26, 2013

Dear Editor,

Enclosed you will find a revised manuscript for submission titled “Genomic Analyses with Biofilter 2.0: Knowledge Driven Filtering, Annotation, and Model Development” by Pendergrass et al. for publication as a software article in BioData Mining.

We appreciate the responses of the reviewer. We have included specific responses to the reviewer in the attached document, as well as made revisions to the text.

Biofilter is a software tool that provides a flexible way to use expert biological knowledge to direct filtering, annotation, and complex predictive model development for elucidating the etiology of complex phenotypic outcomes. We have recently extensively revised and updated Biofilter, and we describe in our submitted manuscript these software changes as well as examples of the functionality of Biofilter. We look forward to sharing more information about this software with the larger scientific community.

Thank you for considering our manuscript for publication in BioData Mining. We look forward to hearing from you.

Sincerely,

Marylyn D. Ritchie, PhD
Director, Center for Systems Genomics
Associate Professor of Microbiology and Cell Biology
Department of Biochemistry & Molecular Biology
Huck Institutes of the Life Sciences
Eberly College of Science
Responses to the Reviewer:

Major Comments

1) It appears that the largest change from previous versions of Biofilter is the way ambiguous annotations/results are handled. I think it would be helpful if the authors could provide additional background on some of the reasons researchers might end up with ambiguous results from a biofilter query. The author’s mention the possibility of multiple names for given genes or groups of genes in the first paragraph of the “Ambiguity and Biofilter” section, but I think a more broad description of such scenarios would be helpful (e.g., differences in SNP annotations to genes from source to source or depending on window size).

Response:

Although ambiguity handling is a significant addition to Biofilter in version 2.0, it is not the only difference. Biofilter 2.0 was completely redesigned and redeveloped from the ground up, and does not include any original source code from earlier versions. This includes the development of the completely separate Library of Knowledge Integration (LOKI). The features that were supported by versions 0.5 and 1.0 were re-implemented in a more generalized way that allows users significantly more flexibility, both in the type and format of their input data, and in the range of possible annotations from the prior knowledge.

Important to note: SNP annotations to genes will not change from source to source, SNP identifiers will either map to genes (depending on the gene boundaries set by the user), or SNPs will not map to genes. The user is provided feedback indicating SNPs not mapped to genes. For instance, if the user supplies a list of SNPs and annotated those SNPs with gene information, if the SNP does not fall within any known gene region or does not have a known position with which to search for, the SNP will be in the output without any annotation.

In terms of genes, if the user provides a gene identifier that is ambiguous (matches more than one gene, Biofilter will ignore it unless the ALLOW_AMBIGUOUS_GENES option is used.

We have modified the text to clarify these points.

2) It is not entirely clear whether Biofilter communicates that ambiguous results are present after a filtering or annotation step is complete; did the programmers include some type of message for ambiguous results if they are included? The option to include or exclude is helpful, but it might be nice to know whether ambiguous annotations are present, particularly when snp-gene lists are long.
The default ambiguity mode in Biofilter 2.0 is “strict,” which will not generate any ambiguous results. This corresponds to the behavior of previous versions of Biofilter, which made no attempt to allow or compensate for gene name ambiguity.

If a user provides an input gene list which contains an ambiguous gene identifier, a warning is always displayed: in “strict” mode, the warning indicates that the ambiguous identifiers were ignored; otherwise, the warning indicates that one or more ambiguous identifiers have caused multiple genes to be included in the analysis. An option also exists to generate a detailed report of the particular identifiers which caused these warnings.

If the user elects to use any of the ambiguity options, then some additional results may be reported, with no further warning alongside any particular result. It is expected that users who enable these options have read the user manual and understand the options’ effects; if in doubt, it is also straightforward to run the same analysis both with and without the ambiguity options and compare the results.

Finally, there is never any ambiguity in the mapping of SNPs to genes, because Biofilter 2.0 uses only one source for gene region boundaries (NCBI Entrez Gene) and one source for SNP positions (NCBI dbSNP). Ambiguity is only a factor when the user provides an input gene list, or when performing analyses that rely on the mapping of genes to groups (where “group” could be a pathway from KEGG, an ontology term from GO, a protein family from Pfam, etc).

We have modified the text to clarify these points.

3) The authors highlight the use of two difference heuristics for reducing ambiguity and mention that: “when set to ‘any’ winner(s) from each one collectively become the preferred choice(s)”. It is not entirely clear what would happen when the two heuristics disagree, although the authors argue this is unlikely to occur. Would both choices be present in the output and would the user have any way of knowing that both choices were present?

Response:

If the user sets the ambiguity reduction heuristic to “any” and the two heuristics disagree, then the candidates chosen by the two heuristics would both be considered to be members of the group. This possibility is represented in the LOKI testing database, but has never been observed to occur in the actual prior knowledge data retrieved from any of the supported sources. We have modified the text to clarify this.

Nonetheless, were it ever to happen, there would be no further warning that more than one apparent group member were actually ambiguous candidates for the same “spot” in
the group. We will investigate the feasibility of detecting and reporting this situation in a future version of the software.

4) The authors highlight the necessity to archive a copy of LOKI whenever submitting a paper or updating the database. Could the authors include an option during a LOKI update to archive the previous database? Does the entire 10gb database have to be archived for each analysis? Could some type of change log be used to reproduce previous version rather than having to store the entire 10gb each time using something like git?

Response:

LOKI database version management is the responsibility of the user, since not all use-cases require the same (or any) archival strategy.

However, the LOKI build script does include a “finalize” option which deletes all of the intermediate data that is necessary in order to efficiently update the database, but is not required for regular use of the database by Biofilter. This function generally reduces the file size of the database by approximately 25% and prevents any future attempts to update the database using the LOKI build script.

Unfortunately, due to the nature and format of the LOKI database, efficient “deltas” or change logs are not feasible at this time. The “finalize” option will reduce the storage requirements for prior snapshots, but the entire database file must still be retained if it will be needed again in the future.

Minor Comments

2) The authors mention in the filtering section that the “result is every combination of SNP and gene from the two lists where the SNP is within the gene”. Does filtering require that the SNPs be within the gene? Can windows be added to the ends of genes during the filtering process? May not need to change this in the manuscript, but it was the first question that came to mind when reading it.

The REGION_POSITION_MARGIN option allows for positions (such as SNPs) to be mapped to regions (such as genes) using a window of any size. In addition, our other software “ldspline” can be employed during the LOKI build process to generate LD-adjusted gene regions, which can then optionally be used in Biofilter with the LD_PROFILE option. This allows for a more precise LD-based “window” around each gene, rather than a constant-sized window.

3) The figures have a few annotations that aren’t clearly explained. For example, why do some of the groups have different aliases (Figure 8, panel A shows gray group with white alias and a gray group with a black alias)? This might also clarify whether one group can be populated from multiple sources, which does not appear to be present from the included schematics.
Groups cannot presently be connected to more than one source, however they may have more than one name provided by the source that defines them. For example, each KEGG pathway has both a numeric ID number as well as a textual pathway name. Likewise, it is possible for the same textual group name to be associated with more than one group. In the testing knowledge diagram, the two nodes labeled “gray” are intended to depict two separate groups (provided by two separate sources) which happen to share the “gray” identifier, but which each also have another name which is distinct (“white” and “black”).

However, ambiguous group names have no effect on any of Biofilter’s internal analyses; they are only important if the user wishes to provide an input list of groups in order to limit their analysis. If the user provides an ambiguous group name, Biofilter’s behavior is similar to the case of ambiguous gene names: a warning will be displayed, and Biofilter will either include all groups which match the name or none of them, according to the option ALLOW_AMBIGUOUS_GROUPS.

We have updated the figure legend for Figure 2, to provide more clarification of what the various symbols in the schematic are.