A Look Back at the Quality of Protein Function Prediction Tools in CAFA

Morteza Pourreza Shahri[1], **Madhusudan Srinivasan**[1], **Diane Bimczok**[2], **Upulee Kanewala**[1], **and Indika Kahanda**[1]

[1]**Gianforte School of Computing, Montana State University, USA**
[2]**Department of Microbiology and Immunology, Montana State University, USA**

Corresponding author:
Indika Kahanda

Email address: indika.kahanda@montana.edu

ABSTRACT

The Critical Assessment of protein Function Annotation algorithms (CAFA) is a large-scale experiment for assessing the computational models for automated function prediction (AFP). The models presented in CAFA have shown excellent promise in terms of prediction accuracy, but quality assurance has been paid relatively less attention. The main challenge associated with conducting systematic testing on AFP software is the lack of a test oracle, which determines passing or failing of a test case; unfortunately, the exact expected outcomes are not well defined for the AFP task. Thus, AFP tools face the *oracle problem*. Metamorphic testing (MT) is a technique used to test programs that face the oracle problem using metamorphic relations (MRs). A MR determines whether a test has passed or failed by specifying how the output should change according to a specific change made to the input. In this work, we use MT to test nine CAFA2 AFP tools by defining a set of MRs that apply input transformations at the protein-level. According to our initial testing, we observe that several tools fail all the test cases and two tools pass all the test cases on different GO ontologies.

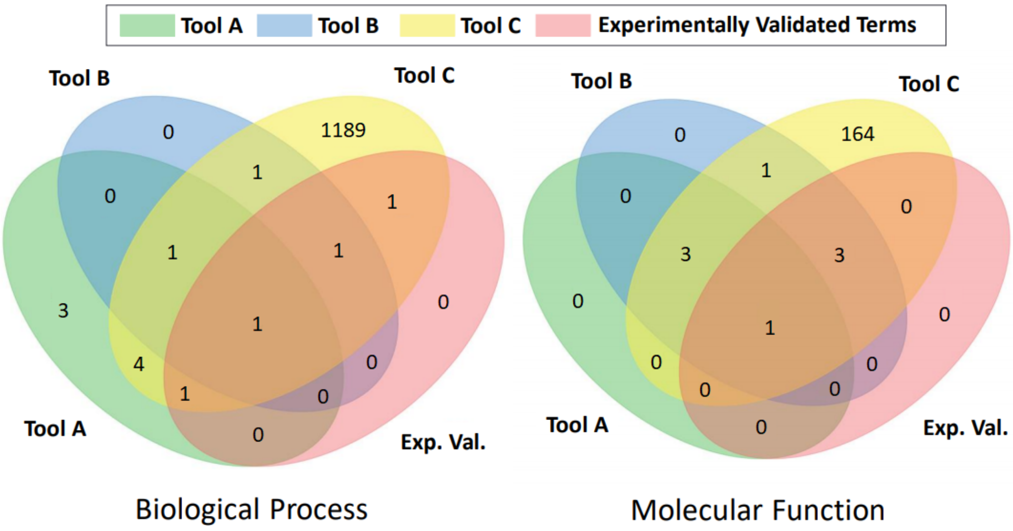
INTRODUCTION

Proteins are one of the main components of a living body. Proteins are important due to various vital functions they perform in living cells. Basically, a cell is alive because of the functions of proteins. However, these functions can sometimes be destructive. Therefore, knowing the functions of proteins is critical. Over the years, a lot of effort has been spent on creating databases of protein functions. Gene Ontology (GO) is a framework used for the model of biology which describes gene functions (Ashburner et al., 2000). Gene Ontology is composed of different classes, each of which demonstrates a gene function, and the hierarchical relations between classes. These relations can be *is-a* relations, *part-of* relations, etc. Each class in Gene Ontology is known as GO term. Gene Ontology, similar to any other ontology contains a DAG (Directed Acyclic Graph) form. Many proteins have GO annotations, however, large portion of proteins do not have GO annotations. Manually annotating Gene Ontology is very expensive and time-consuming. Therefore, computational models for predicting the corresponding GO terms are essential. Even though many studies have been done on automated prediction of protein functions, a criterion for evaluation of the tools was needed. As a result, Function Special Interest Group[1] started to evaluate these tools in the form of a challenge. This challenge is called Critical Assessment of protein Function Annotation (CAFA). At the time of writing this paper, CAFA2 was the latest challenge that its results of evaluation were publicly available (Jiang et al., 2016).

Many tools were presented in CAFA2. Based on the different criteria such as macro-AUROC, F-max, and S_{min} , which were employed to evaluate the tools, several tools fall into the category of top-performing tools. Nevertheless, selecting a tool from this list to perform experiments or research would be very difficult. One way to select a tool is randomly picking a few tools, feeding well-known sequences into the tools, and comparing the results with the experimentally validated terms, which are the results of

[1] https://biofunctionprediction.org/

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27161v1 | CC BY 4.0 Open Access | rec: 29 Aug 2018, publ: 29 Aug 2018



Biological Process

Molecular Function

Figure 1. Distribution of output GO terms for the protein TYRO_HUMAN on Biological Process and Molecular Function ontologies

a physical characterization of a gene that has supported its association with the GO term[2]. However, using a few tools and sequences, users would observe that each tool provides different set of output GO terms, and only a few terms would be in common with the experimentally validated terms. Figure 1 shows the distribution of GO terms using three randomly selected tools from the top-performing tools of CAFA2 beside the corresponding experimentally validated terms of the well-known protein *Tyrosinase*. The distribution of GO terms in Molecular Function ontology shows only one GO term is in common between the three tools and the experimentally validated terms. This observation is true for the Biological Process as well. The next important observation on Biological Process ontology is that one of the tools returns about 1200 GO terms, whereas another tool outputs only four GO terms for the same protein. This inconsistent behavior of the AFP tools makes it difficult for users to select a tool.

As mentioned earlier, there are a few problems associated with the AFP tools. The first problem is that each tool returns different set of GO terms. The second problem of the tools is having a few GO terms in common with the experimentally validated terms. Moreover, the experimentally validated terms set is incomplete due to the lack of knowledge or experiments. Therefore, these problems cause consequences on how a biologist would select a tool for their research, or how a developer would test their tool.

In this study, we explore the feasibility of applying a new method which is called Metamorphic Testing (MT) for testing bioinformatics software. The results of this study show that several AFP tools fail all the test cases and only at most two tools pass all the test cases.

Metamorphic Testing

In complex systems, such as AFP tools, it is practically difficult to determine whether the output provided by the system for a given input is correct. This is known as the *oracle problem*. MT can be used to test programs that face the oracle problem (Chen et al., 1998). The MT process involves deriving metamorphic relations (MRs) and generating test cases based on the MR. The MRs are a set of properties derived from the program under test and specifies how the output would change according to a specific change made to the input. Source test cases are typically derived using a traditional test case generation approach such as random test generation. Follow-up test cases are derived by applying the MR to the source test case. Then, the source and follow-up test cases are executed and outputs of these test cases are used to verify whether MR was violated or not. The violation of a MR indicates faults in the program.

Figure 2 shows how MT is applied to a sorting program. This sorting program arranges a random set of numbers provided as input in the ascending order. A MR for the sorting program states that when the original set of numbers are shuffled and used as a input to the program, the output must be equal to

[2] http://www.geneontology.org/page/guide-go-evidence-codes

the original output. Thus, the source test case can be created by generating a set of random numbers and the follow-up test case can be created by shuffling the source test case. A fault is detected in the sorting program if the output from the source and follow-up test case are not equal as defined in the MR.

Related Works

Srinivasan et al. worked on applying MT to LingPipe, a tool for processing text using computational linguistics, which is often used in bioinformatics for bio-entity recognition from biomedical literature (Srinivasan et al., 2018). The authors proposed 10 novel MRs and effectiveness of each of the MRs to detect faults was identified using mutation testing approach. The effectiveness of each the MR was calculated based on the ability to kill mutants. Mutation testing involves modifying a program in small ways and each mutated version is called a mutant. The faults were identified using the MRs.

Lundgren et al. examined the effectiveness of the MRs on genome alignment tool (BBMap) (Lundgren and Kanewala, 2016). The experiment results showed that MT could identify subtle faults. Ramanathan et al. used MT to develop workflow of epidemiological models (Ramanathan et al., 2012). They showed that MT can be useful when mathematical models fail. Pullum and Ozmen showed that MT could be effective in testing epidemiological models (Pullum and Ozmen, 2012). They used a differential equation and agent-based models for generating MR-transformed parameter values. Chen et al. used MT to test two open-source bioinformatics programs (Chen et al., 2009). The first program GNLab, a tool for large-scale analysis and simulation of gene regulatory networks. The second tool SeqMap deals with mapping a short sequence that reads with a reference genome. Eleni et al. conducted metamorphic testing on three commonly used NGS (Next Generation Sequencing) short-read alignment programs: BWA, Bowtie, and Bowtie2 (Giannoulatou et al., 2014).

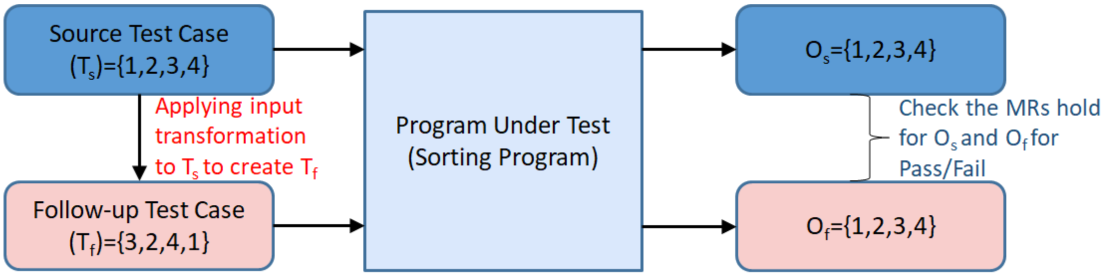


Figure 2. MT example for sorting program

METAMORPHIC TESTING FOR AFP TOOLS

The first step of applying MT to any problem is defining valid MRs for the problem. Defining the MRs for the AFP tools would be tricky because they really depend on the input sequences. One of the simplest MRs that could be defined is ”expecting a change” in the output GO terms for the proteins and their corresponding well-studied variants. For instance, if the source test case is the canonical sequence of the protein *Tyrosinase*, and the follow-up test case is a disease variant of this protein which causes *Albinism (OCA1A)*, biological knowledge says that the output GO terms of the canonical sequence and the disease variant must be different.

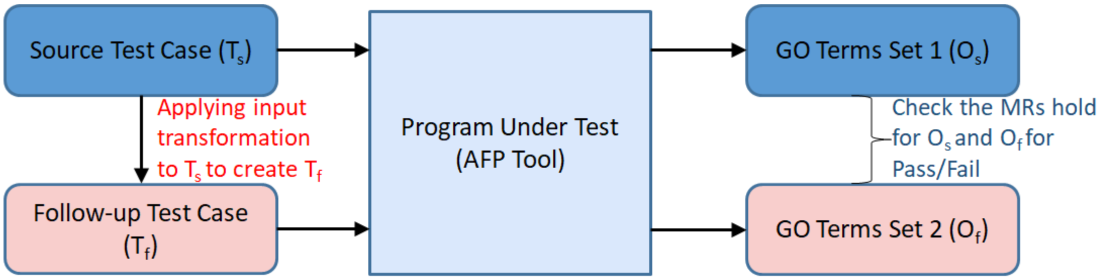


Figure 3. Architecture of the Metamorphic Testing system on AFP tools

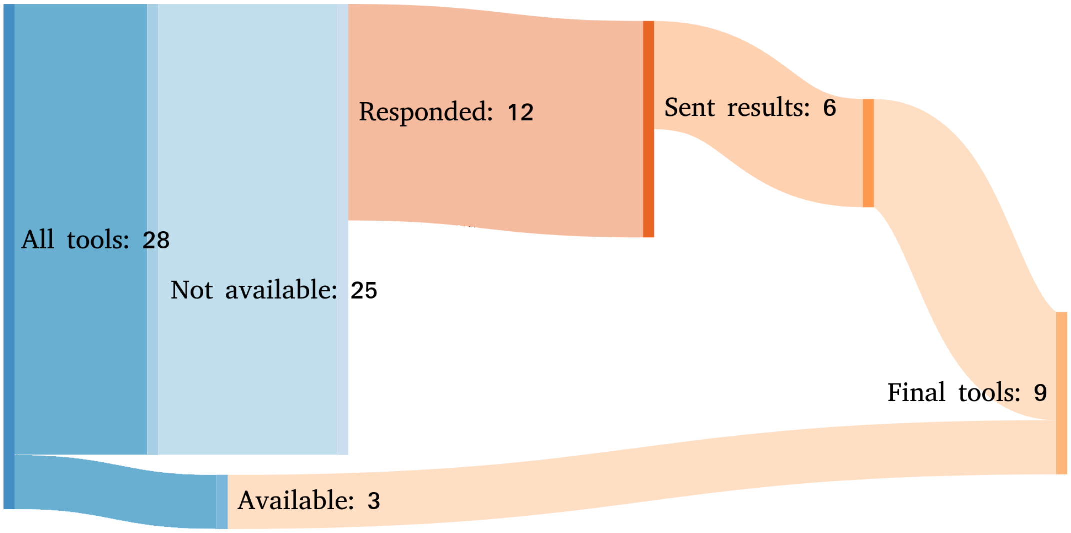


Figure 4. The flow of selection of the tools

Therefore, as shown in Figure 3, we perform the following steps:

1. Running the program with the source test case and achieving O_s as the output. The source test case is a FASTA sequence and O_s will be a set of GO terms.

2. Generating a follow-up test case using the source case, and executing the program with the follow-up test case and getting O_f. The follow-up test case also is a FASTA sequence derived from a known variation of the source test case, and the output is a set of GO terms as well.

3. Checking whether the MR holds for O_s and O_f. In this example, the MR is any change in the list of output GO terms, i.e. additions, deletions, etc. If the expected change is satisfied, it will be a *pass*, otherwise, it will be a *fail*.

Tools Selection Criteria

To evaluate the feasibility of applying MT to AFP tools, first we need a set of AFP tools. For this purpose, we started with the 28 top-performing tools from the CAFA2 challenge. From these 28 tools, most are not publicly available, and some are very hard to setup and run. So, we selected tools that can be set-up for execution by spending a maximum of thirty minutes by a graduate student. At the time of this investigation, only three tools were publicly available or worked. As we want to do the experiments on as many tools as possible, we contacted authors of the remaining 25 tools, asking them to run the sequences used as source and follow-up test cases on their tools and send us the outputs. Twelve authors replied the email, and 6 out of 12 authors sent us the outputs. Thus, we could evaluate the following nine tools using the proposed MT approach: EVEX (Van Landeghem et al., 2012), PFP (Hawkins et al., 2009), CONS (Khan et al., 2015), GORBI (Škunca et al., 2013), CBRG (Altenhoff et al., 2014), ProFun (Cao and Cheng, 2016), PANNZER (Koskinen et al., 2015), Argot2 (Falda et al., 2012), and INGA (Piovesan et al., 2015). Figure 4 shows the flow of selection of the tools.

Source and follow-up test Cases

We used 18 sequences from three carefully selected proteins to test the AFP tools. These proteins are:

- Tyrosinase (TYRO_HUMAN)
- Cytokine receptor common subunit gamma (IL2RG_HUMAN)
- Toll-like receptor 4 (TLR4_HUMAN)

The sequences consist of the canonical sequence and the sequences of variants as follows:

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27161v1 | CC BY 4.0 Open Access | rec: 29 Aug 2018, publ: 29 Aug 2018

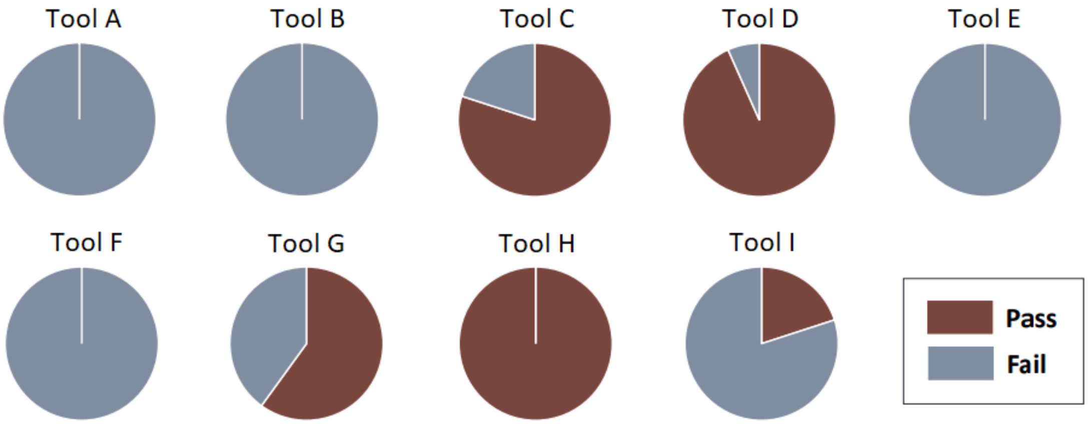


Figure 5. Results on Molecular Function ontology

- TYRO_HUMAN: (Canonical sequence + 7 disease variants)
- IL2RG_HUMAN: (Canonical sequence + 4 disease variants)
- TLR4_HUMAN: (Canonical sequence + 2 splice variants + 2 natural variants)

For each protein, we use the canonical sequence as the source test case and each of the variant sequences as the follow-up test case. Therefore, we have 15 pairs of source and follow-up test cases.

Test execution

The next step in applying MT to AFP tools is feeding the test cases into the tools, and checking whether the MRs hold for each execution. Therefore, we have nine tools and 15 pairs of source and follow-up test cases.

For each pair of source and follow-up test case, we store the output GO terms for the Molecular Function and Biological Process ontologies in different files, and compare the GO terms of O_s and O_f, and we report the results of different ontologies separately. We ignore the Cellular Component ontology for this study because the selected proteins have expected changes in the other two ontologies, and we are not sure about the behavior of them on the Cellular Component ontology.

RESULTS

Figures 5 and 6 show the results of executing the tools with 15 test case pairs on Molecular Function ontology and Biological Process ontology, respectively. Each pie chart shows the number of passes and fails of the 15 test case pairs for a given tool.

As shown in Figure 5, four out of nine tools fail all the test cases. This phenomenon can happen if the tools are not designed to detect the results of variations in the protein sequence. Another observation from the Molecular Function ontology states that only one tool passes all the test cases. The rest of the tools have a mix of passes and fails.

We validated the results by executing the tools on Biological Process ontology. As shown in Figure 6, the same four tools failed again. Besides, on the Biological Process ontology, two tools passed all the test cases.

CONCLUSIONS AND FUTURE WORKS

In this study, we explore the feasibility of applying MT to AFP tools. Biological knowledge enabled us to expect some changes in the functions of some proteins. Therefore, it is possible to create MRs using carefully selected protein examples such as disease variants.

Our results show that several tools do not pass all the test cases. We believe that the AFP community need to work together for the quality assurance of these tools. The results also show that using MRs

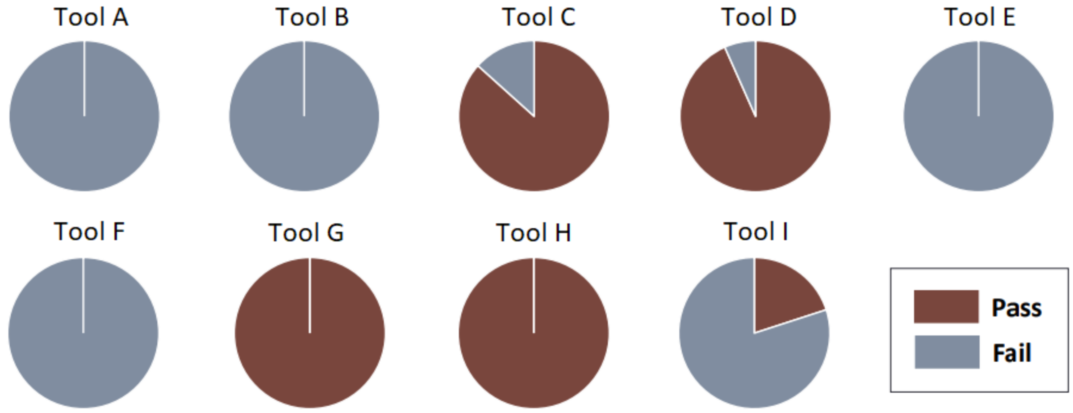


Figure 6. Results on Biological Process ontology

developed using the biological knowledge, we can detect faults in the tools. As a result, we conclude that MT can be a promising avenue for testing AFP tools.

In future, we will communicate the results to the authors of the respective tools. Currently, we have not analyzed the exact changes that caused the pass and fail. Therefore, as the next step, we will do a post-analysis on the results to find the changes. We will also create more specific MRs such as the *actual change* instead of just checking for the *existence of a change*.

We plan to develop an expanded test suite by exploring MRs for Cellular Component ontology and Human Phenotype ontology, and by employing different types of protein examples. We will also work with the AFP community to increase the number of tools to be tested. Eventually, we will develop a testing framework which is readily available for the users and developers.

REFERENCES

Altenhoff, A. M., Škunca, N., Glover, N., Train, C.-M., Sueki, A., et al. (2014). The OMA orthology database in 2015: function predictions, better plant support, synteny view and other improvements. *Nucleic acids research*, 43(D1):D240–D249.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., et al. (2000). Gene Ontology: tool for the unification of biology. *Nature genetics*, 25(1):25.

Cao, R. and Cheng, J. (2016). Integrated protein function prediction by mining function associations, sequences, and protein–protein and gene–gene interaction networks. *Methods*, 93:84–91.

Chen, T. Y., Cheung, S. C., and Yiu, S. M. (1998). Metamorphic testing: a new approach for generating next test cases. Technical report, Technical Report HKUST-CS98-01, Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong.

Chen, T. Y., Ho, J. W., Liu, H., and Xie, X. (2009). An innovative approach for testing bioinformatics programs using metamorphic testing. *BMC bioinformatics*, 10(1):24.

Falda, M., Toppo, S., Pescarolo, A., Lavezzo, E., Di Camillo, B., Facchinetti, A., et al. (2012). Argot2: a large scale function prediction tool relying on semantic similarity of weighted Gene Ontology terms. *BMC bioinformatics*, 13(4):S14.

Giannoulatou, E., Park, S.-H., Humphreys, D. T., and Ho, J. W. (2014). Verification and validation of bioinformatics software without a gold standard: a case study of bwa and bowtie. *BMC bioinformatics*, 15(16):S15.

Hawkins, T., Chitale, M., Luban, S., and Kihara, D. (2009). PFP: Automated prediction of Gene Ontology functional annotations with confidence scores using protein sequence data. *Proteins: Structure, Function, and Bioinformatics*, 74(3):566–582.

Jiang, Y., Oron, T. R., Clark, W. T., Bankapur, A. R., D'Andrea, D., Lepore, R., et al. (2016). An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome biology*, 17(1):184.

Khan, I. K., Wei, Q., Chapman, S., Kihara, D., et al. (2015). The PFP and ESG protein function prediction methods in 2014: effect of database updates and ensemble approaches. *GigaScience*, 4(1):43.

Koskinen, P., Törönen, P., Nokso-Koivisto, J., and Holm, L. (2015). PANNZER: high-throughput functional annotation of uncharacterized proteins in an error-prone environment. *Bioinformatics*, 31(10):1544–1552.

Lundgren, A. and Kanewala, U. (2016). Experiences of testing bioinformatics programs for detecting subtle faults. In *Proceedings of the International Workshop on Software Engineering for Science*, pages 16–22. ACM.

Piovesan, D., Giollo, M., Leonardi, E., Ferrari, C., and Tosatto, S. C. (2015). INGA: protein function prediction combining interaction networks, domain assignments and sequence similarity. *Nucleic acids research*, 43(W1):W134–W140.

Pullum, L. L. and Ozmen, O. (2012). Early results from metamorphic testing of epidemiological models. In *2012 ASE/IEEE International Conference on BioMedical Computing (BioMedCom)*, pages 62–67. IEEE.

Ramanathan, A., Steed, C. A., and Pullum, L. L. (2012). Verification of compartmental epidemiological models using metamorphic testing, model checking and visual analytics. In *BioMedical Computing (BioMedCom), 2012 ASE/IEEE International Conference on*, pages 68–73. IEEE.

Škunca, N., Bošnjak, M., Kriško, A., Panov, P., Džeroski, S., et al. (2013). Phyletic profiling with cliques of orthologs is enhanced by signatures of paralogy relationships. *PLoS computational biology*, 9(1):e1002852.

Srinivasan, M., Pourreza Shahri, M., Kahanda, I., and Kanewala, U. (2018). Quality assurance of bioinformatics software: a case study of testing a biomedical text processing tool using Metamorphic Testing. In *Proceedings of the 3rd International Workshop on Metamorphic Testing*, pages 26–33. ACM.

Van Landeghem, S., Hakala, K., Rönnqvist, S., Salakoski, T., Van de Peer, Y., and Ginter, F. (2012). Exploring biomolecular literature with EVEX: connecting genes through events, homology, and indirect associations. *Advances in bioinformatics*, 2012.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27161v1 | CC BY 4.0 Open Access | rec: 29 Aug 2018, publ: 29 Aug 2018