

Managing Expectations When Publishing Tools and Methods for Computational Proteomics

Lennart Martens,^{†,‡,⊥} Oliver Kohlbacher,^{§,#} and Susan T. Weintraub^{*,||}

[†]Department of Biochemistry, Ghent University, Albert Baertsoenkaai 3, B-9000 Ghent, Belgium

[‡]Department of Medical Protein Research, VIB, Albert Baertsoenkaai 3, B-9000 Ghent, Belgium

[§]Center for Bioinformatics, Quantitative Biology Center, Department of Computer Science, and Faculty of Medicine, University of Tuebingen, Sand 14, 72076 Tuebingen, Germany

^{||}Department of Biochemistry, The University of Texas Health Science Center at San Antonio, 7703 Floyd Curl Drive, San Antonio 78229, Texas, United States

ABSTRACT: Computational tools are pivotal in proteomics because they are crucial for identification, quantification, and statistical assessment of data. The gateway to finding the best choice of a tool or approach for a particular problem is frequently journal articles, yet there is often an overwhelming variety of options that makes it hard to decide on the best solution. This is particularly difficult for nonexperts in bioinformatics. The maturity, reliability, and performance of tools can vary widely because publications may appear at different stages of development. A novel idea might merit early publication despite only offering proof-of-principle, while it may take years before a tool can be considered mature, and by that time it might be difficult for a new publication to be accepted because of a perceived lack of novelty. After discussions with members of the computational mass spectrometry community, we describe here proposed recommendations for organization of informatics manuscripts as a way to set the expectations of readers (and reviewers) through three different manuscript types that are based on existing journal designations. *Brief Communications* are short reports describing novel computational approaches where the implementation is not necessarily production-ready. *Research Articles* present both a novel idea and mature implementation that has been suitably benchmarked. *Application Notes* focus on a mature and tested tool or concept and need not be novel but should offer advancement from improved quality, ease of use, and/or implementation. Organizing computational proteomics contributions into these three manuscript types will facilitate the review process and will also enable readers to identify the maturity and applicability of the tool for their own workflows.

KEYWORDS: computational mass spectrometry, bioinformatics, manuscript types

Computational Proteomics: Managing Expectations



■ INTRODUCTION

A computational proteomics manuscript typically follows one of two submission paths: either it is sent to a bioinformatics journal or it is submitted to a journal in its proteomics application domain. The distinction lies not only in the audience that is being addressed through each type of journal but also in the manuscript review process. Indeed, a manuscript that will be enthusiastically received in a bioinformatics journal may not be suitable for consideration in an application domain journal and vice versa. It is crucial for manuscripts sent to an application domain to make the applicability of the work clearly evident.

Discussions among computational proteomics authors, reviewers, and editors quickly revealed that for proteomics journals, there currently are no useful guidelines in place that outline the format and basic content requirements for a proteomics bioinformatics manuscript. This effectively leaves authors, reviewers and editors to make ad hoc decisions for each manuscript. As a result, criteria for reviewing these manuscripts can become somewhat arbitrary, despite the best efforts of journal editors to maintain consistency. Moreover,

published tools can display considerable heterogeneity in ease of use, performance, or general applicability. Readers' heightened expectations can thus easily turn into frustration.

To remedy this situation, we propose guidelines for organization of computational proteomics manuscripts that are submitted to proteomics journals. These guidelines differentiate the types of contributions through manuscript types that are already in use in journals: *Brief Communications*, *Research Articles*, and *Application Notes*. As outlined in Table 1, these recommendations focus on the applicability of the presented work based on the following general topics: potential for reuse, provision of documentation, availability/licensing, system requirements, and limitations. The manuscript types are distinguished predominantly by their level of advancement of the software or informatics concept being described beyond the state-of-the-art. While *Research Articles* and *Brief Communications* are expected to demonstrate such advancement, this is not necessary for *Application Notes*. These designations are in

Received: December 23, 2014

Published: March 12, 2015

Table 1. Computational Manuscript Types and Their Requirements

	brief communication	research article	application note
substantial advancement	+	+	
potential for reuse		+	+
general limitations	+	+	+
system limitations		+	+
end-user documentation	+	+	+
developer documentation		+	+
sample data		+	+
benchmark data set		+	+
availability		+	+
license information		+	+
system requirements		+	+

no way intended to indicate the value or potential impact of the manuscript but rather to help set readers' expectations about the content of the paper.

A key requirement that is shared by all of the proposed informatics manuscript types focuses on an essential aspect of the applicability of any concept, approach, or tool: its limitations. A clear statement of the limitations of the tool being presented in the manuscript immediately defines the current status of the tool for readers and greatly facilitates the review process by setting the correct level of expectation for reviewers. It is important here to realize that the idea of specifying limitations is not new in computer science. Perhaps the best-known example is provided by Gamma, Helm, Johnson, and Vlissides in their landmark book *Design Patterns: Elements of Reusable Object-Oriented Software*.¹ In this book, each design pattern (a recurring solution to a common problem in software design) is accompanied both by a list of useful applications as well as a list of situations in which the design pattern would fail. This latter characteristic is often as useful as the former, as it discourages application of the design pattern outside of its intended use. The limitations section requirement of our proposed guidelines follows this example and mandates that authors specify the limitations in the application of their concept, approach, or tool. A specific subtopic of this requirement deals with any known system limits (e.g., a data set greater than one million entries will crash the tool). The statement of limitations is a crucial aspect of the applicability of any concept, approach, or tool as it allows the reader to understand clearly when and how the system performs as promised and where it may no longer do so. Additionally, it allows others to address these identified limitations in future developments, providing a more efficient path to the creation of improved methods and tools.

■ BRIEF COMMUNICATIONS

Brief Communications are aimed at presenting a novel concept or idea rather than a directly applicable method or software tool. Indeed, the only additional requirement for a *Brief Communication* is that it should discuss the limitations of the approach (see above). In this way, a reader will know instantly from the contribution type what can be expected in terms of applicability. This manuscript type serves as a way to publicize an informatics concept, algorithm, or software before development of a polished user interface and makes the community

aware of the concept/algorithm/software for potential incorporation into other tools and applications.

■ RESEARCH ARTICLES

All of the manuscript elements in Table 1 are needed for a *Research Article*. In addition to demonstrating substantial advancement over currently available tools, the potential for reuse must be verifiable. This means that an implementation must either be sufficiently detailed in the manuscript or be made available for reviewers to test and, after publication, for readers to use. Moreover, one or more sample data sets needs to be provided, ideally via a publicly available, permanent repository, so that the reviewer or reader can try the approach even if they have no data locally available. Another requirement is that the necessary documentation be made available for developers and end users. This documentation must, at the very least, enable an end user to operate the tool or a developer to reuse or automate the code library or application. As such, availability and licensing of the tool are key aspects. A location where the tool can be found (typically a URL) needs to be specified; this should not be the authors' local server to maintain reviewer anonymity. SourceForge, BitBucket, and GitHub are examples of sites that are suitable for hosting the software (at least during the review process). In addition to the location, the applicable license needs to be specified. It is important to note that a tool without any associated license cannot be legally reused by anyone, thus necessitating the provision of a license. Often used examples for academic software include a Creative Commons license, GNU GPL and L-GPL licenses, BSD license, and the Apache2 license. For more information on different license models, see Morin et al.²

A short description of the system requirements needs to be presented in an informatics *Research Article* so that reviewers and readers can understand what is needed for the tool to work on their hardware. The final requirement is unique to *Research Articles* and stipulates that a publicly available benchmark data set be provided for these contributions. This benchmark data set allows the reported advancement beyond the state-of-the-art to be verified during the review process and also permits future work to be fairly and correctly compared to the system described in the journal article.

■ APPLICATION NOTES

Application Notes share all but one of the content requirements with *Research Articles*; they do not need to present a novel concept/algorithm/tool. This manuscript type serves a key role in providing an avenue to report on advancement of an existing concept or tool, or application of a tool to a new type of data set. *Application Notes* can also be used as a reference point for publicly available software that is regularly being updated. In this type of article, there would be a statement of the purpose of the software, a brief description of how it works, an explanation of the type of hardware needed, and examples of applications. Readers would be directed to the software Web site for more information and to access/download the software. Here, too, it is important to host the software at a reliable third-party site.

■ CONCLUSIONS

By specifying formalized requirements for proteomics bioinformatics contributions, authors will be better equipped to prepare their manuscript and enhance the likelihood of acceptance, editors will be able to more readily determine

whether a contribution will be of interest to journal readers, and reviewers can more easily assess the value of the work described. Most importantly of all, however, readers will be able to immediately identify the applicability of the work presented, taking into account their own needs, their local hardware and software environment, and their data. It is hoped that corresponding guidelines will be adopted by the other proteomics journals. Any additional effort associated with application of these guidelines will, no doubt, be offset by providing a mechanism to enhance commonality of manuscript requirements for computational mass spectrometry for the benefit of authors, reviewers, and readers. These guidelines will need to be evaluated periodically so that they can evolve over time in keeping with progression of the field.

■ AUTHOR INFORMATION

Corresponding Author

*Tel: 210-567-4043. E-mail: weintraub@uthscsa.edu.

Notes

The authors declare no competing financial interest.

[†]L.M.: E-mail: lennart.martens@UGent.be.

[#]O.K.: E-mail: oliver.kohlbacher@uni-tuebingen.de.

■ ACKNOWLEDGMENTS

We acknowledge the participants of Dagstuhl seminar 13491 “Computational Mass Spectrometry” (December 2013) (<http://www.dagstuhl.de/de/programm/kalender/semhp/?seminr=13491>) for their contributions to the discussion on computational manuscripts. In particular, we thank John Cottrell, Henning Hermjakob, William Noble, Olga Vitek, and Witold Wolski for their input on a first draft of reviewing recommendations. We are grateful for the support of Schloss Dagstuhl - Leibniz-Zentrum für Informatik for supporting the seminar that led to this work.

■ REFERENCES

- (1) Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*; Addison Wesley: Boston, 1994.
- (2) Morin, A.; Urban, J.; Sliz, P.; Quick, A. Guide to Software Licensing for the Scientist-Programmer. *PLoS Comput. Biol.* **2012**, *8* (7), e1002598.