The Journal of Open Source Software

teiphy: A Python Package for Converting TEI XML Collations to NEXUS and Other Formats

Joey McCollum [1] **and Robert Turnbull** [2]

1 Institute for Religion and Critical Inquiry, Australian Catholic University, Australia **2** Melbourne Data Analytics Platform, University of Melboune, Australia

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

Textual scholars have been using phylogenetics to analyze manuscript traditions since the early 1990s (Robinson & O'Hara, 1992). Many standard phylogenetic software packages accept as input the NEXUS file format (Maddison et al., 1997). The `teiphy` program takes a collation of texts encoded in TEI XML format and can convert it to any of the following formats amenable to phylogenetic analysis: NEXUS (with support for ambiguous states and clock model calibration data blocks for MrBayes or BEAST2), Hennig86, PHYLIP (relaxed for use with RAxML), FASTA (relaxed for use with RAxML), and STEMMA (designed for Stephen C. Carlson's stemmatic software tailored for textual data). For machine learning-based analyses, `teiphy` can also convert a TEI XML collation to a collation matrix in NumPy, Pandas DataFrame, CSV, TSV, or Excel format.

Statement of Need

For over a decade, the Text Encoding Initiative has endeavored to provide an international standard for digitally encoding textual information for the humanities (Ide & Sperberg-McQueen, 1995). Their guidelines describe a standard format for encoding material details, textual transcriptions, and critical apparatuses (TEI Consortium, 2022). Due to its rich and well-documented set of elements for expressing a wide range of features in these settings, the Text Encoding Initiative's Extensible Markup Language format (hereafter abbreviated TEI XML) has become the *de facto* format for textual data in the digital humanities (Fischer, 2020). Its expressive power has proven increasingly valuable since its release, as scholars have learned—sometimes the hard way—that digital transcriptions and collations should

1. preserve as much detail as they can from their material sources, including paratextual features;
2. reproduce the text of their sources as closely as possible, with editorial regularizations to things like orthography, accentuation, and scribal shorthand encoded alongside rather than in place of the source text; and
3. describe uncertainties about a source's contents as accurately as possible, allowing for degrees of uncertainty and multiple choices for disambiguations if necessary.

These principles have much bearing on the editing of critical texts, a task fundamental to the field of philology. Phylogenetic algorithms developed in the context of evolutionary biology have been popular approaches to this task, especially as philology itself has taken a digital turn over the years. Taking the most arduous part of reconstructing a textual tradition and delegating it to a computer proved to be a promising technique, and its successful demonstration with a portion of *The Canterbury Tales* was a milestone in the development of the field (Barbrook et al., 1998). Soon after this, the same methods were applied more comprehensively to the tradition of *Lanseloet van Denemerken* in a work that would formalize many practical rules

for computer-assisted textual criticism (Salemans, 2000). Over the decades preceding and following these developments, biologists have continued to develop and improve phylogenetic methods (Felsenstein, 2004), and textual critics have adapted these improvements and even added their own innovations to make the process more suitable for their purposes (Carlson, 2015; Edmondson, 2019; Hyytiäinen, 2021; Spencer et al., 2002, 2004; Turnbull, 2020).

Phylogenetic algorithms have a natural place in textual criticism given the deep analogy between textual traditions and evolutionary trees of life: a sequence alignment, which consists of taxa, sites or characters, and the states of taxa at those characters, corresponds almost identically to a collation, which consists of witnesses to the text, locations of textual variation (which we will call "variation units" from here on), and the variant readings attested by witnesses at those points.

The problem is that no currently available phylogenetic software accepts inputs in TEI XML format. Instead, most phylogenetic programs expect inputs in NEXUS format (Maddison et al., 1997). This format was conceived with versatility—including use in stemmatic and non-biological applications—in mind, and this design choice has been vindicated in its general applicability, but NEXUS is neither equipped nor meant to express the same kinds of details that TEI XML is. Conversely, for those interested primarily in working with the collation as an alignment, TEI XML is overkill. Thus, a chasm continues to separate data born in the digital humanities from phylogenetic tools born in the biological sciences, and the only way to bridge it is by conversion.

The challenge is made more daunting by the variety of tools available for phylogenetic and other analyses, some of which expect inputs other than NEXUS files or NEXUS files augmented in different ways. For instance, the cladistic software PAUP*, which has historically been the tool of choice for text-critical applications that evaluate candidate trees using the criterion of maximum parsimony, reads inputs in NEXUS format (Swofford, 2003), but the TNT software, its main competitor among maximum parsimony-based programs, expects inputs in Hennig86 format (Farris, 1988; Goloboff & Catalano, 2016). While Hennig86 format does not allow for as much flexibility in the input as NEXUS does (e.g., it does not support ambiguous character states that can be disambiguated as some states but not others), TNT's extensive support for morphological state models makes it potentially more suitable for textual data, and textual critics may prefer it to PAUP*. In the same regime, Stephen C. Carlson's STEMMA software, initially developed for the cladistic analysis of biblical texts known to be affected by contamination, makes substantial adaptations to the basic maximum-parsimony phylogenetic approach to account for this problem and other constraints common in a text-critical setting (Carlson, 2015); however, the input collation data must be provided in a STEMMA-specific format. Likewise, among programs that use the maximum likelihood criterion instead of maximum parsimony, IQ-TREE accepts inputs in NEXUS format (Minh et al., 2020), but RAxML interfaces primarily with inputs in PHYLIP and FASTA format (Stamatakis, 2014). Finally, for phylogenetic programs that attempt to estimate the posterior distribution of candidate trees in a Bayesian fashion, MrBayes and BEAST2 both accept inputs in NEXUS format (or can convert NEXUS inputs to their standard input format), but they expect taxon dates (for the calibration of evolutionary clock models) to be specified in the NEXUS file in different code blocks (Bouckaert et al., 2019; Ronquist et al., 2012).

Furthermore, end users of textual collations may be interested in non-phylogenetic analyses. In this case, the desired input format is often not a NEXUS-style sequence alignment, but a collation matrix with a row for each variant reading and a column for each witness (or vice-versa). For Python machine-learning libraries like Scikit-learn (Pedregosa et al., 2011), NIMFA (Zitnik & Zupan, 2012), and TensorFlow (Abadi et al., 2015), the standard input format is a NumPy array (Harris et al., 2020), although Pandas DataFrames, which support row and column labels (McKinney, 2010; The pandas development team, 2020), may also be supported. (The latter format also extends the conversion pipeline to many other formats, including CSV, TSV, and Excel files; Pandas DataFrames can even write their contents to database tables.) To give an example, the text of the New Testament has served as a testbed

for multiple analyses of this type, which have generally applied clustering and biclustering algorithms to collation matrices (Baldwin, 2010; Finney, 2018; McCollum, 2019; Thorpe, 2002; Willker, 2008). Given the prevalence of efforts like these, the need for a means of converting TEI XML collations to NumPy collation matrices or labeled Pandas DataFrames is clear.

Design

While the conversion process is a straightforward one for most collation data, various sources of ambiguity can make a one-to-one mapping of witnesses to readings impossible. One such source of ambiguity is lacunae, or gaps in the text due to erasure, faded ink, or damage to the page. Another is retroversions, or readings in the original language of the text reconstructed through the back-translation of subsequent versions of the text in other languages. Mechanisms for modeling ambiguous states resulting from situations like these exist in both TEI XML and NEXUS, and in both parsimony- and likelihood-based phylogenetic methods, ambiguities about the states at the leaves and even at the root of the tree can be encoded and leveraged in the inference process. For these reasons, it is imperative to ensure that these types of judgments, as well as other rich features from TEI XML, can be respected (and, where necessary, preserved) in the conversion process.

Collations should preserve as much detail as possible, including information on how certain types of data can be normalized and collapsed for analysis. Since one might want to conduct the same analysis at different levels of granularity, the underlying collation data should be available for use in any case, and only the output of the conversion should reflect changes in the desired level of detail. Likewise, as noted in the previous section, uncertainty about witnesses' attestations should be encoded in the collation and preserved in the conversion of the collation.

For text-critical purposes, differences in granularity typically concern which types of variant readings we consider important for analysis. At the lowest level, readings with uncertain or reconstructed portions are almost always considered identical with their reconstructions (provided these reconstructions can be made unambiguously) for the purpose of analysis. Defective forms that are obvious misspellings of a more substantive reading are often treated the same way. Even orthographic subvariants that reflect equally "correct" regional spelling practices may be considered too common and of too trivial a nature to be of value for analysis. Other readings that do not fall under these rubrics but are nevertheless considered manifestly secondary (due to late and/or isolated attestion, for instance), may also be considered uninformative "noise" that is better left filtered out.

Use Case

Due to the availability of extensive collation data for the Greek New Testament, and because this project was originally developed for use with such data, we tested this library on a sample collation of the book of Ephesians in thirty-eight textual witnesses (including the first-hand texts of manuscripts, corrections made to manuscripts by later hands, translations to other languages, and quotations from church fathers). The manuscript transcriptions used for this collation were those produced by the University of Birmingham's Institute for Textual Scholarship and Electronic Editing (ITSEE) for the International Greek New Testament Project (IGNTP); they are freely accessible at https://itseeweb.cal.bham.ac.uk/epistulae/XML/igntp.xml. To achieve a balance between variety and conciseness, we restricted the collation to a set of forty-two variation units in Ephesians corresponding to variation units in the United Bible Societies Greek New Testament (Aland et al., 2014), which highlights variation units that affect substantive matters of translation.

In our example collation, witnesses are described in the `listWit` element under the `teiHeader`. Because most New Testament witnesses are identified by numerical Gregory-Aland identifiers,

these witnesses are identified with `@n` attributes; the recommended practice is to identify such elements by `@xml:id` attributes, but this software is designed to work with either identifying attribute (preferring `@xml:id` if both are provided), and we have left things as they are to demonstrate this feature.

The `witness` elements in the example collation also contain `origDate` elements that provide dates or date ranges for the corresponding witnesses. Where a witness can be dated to a specific year, the `@when` attribute is sufficient to specify this; if it can be dated within a range of years, the `@from` and `@to` attributes or the `@notBefore` and `@notAfter` attributes should be used; the software will work with any of these options. While such dating elements are not required, our software includes them in the conversion process whenever possible. This way, phylogenetic methods that employ clock models and other chronological constraints can benefit from this information when it is provided.

Each variation unit is encoded as an `app` element with a unique `@xml:id` attribute. Within a variation unit, a `lem` element without a `@wit` attribute presents the main text, and it is followed by `rdg` elements that describe variant readings (with the first `rdg` duplicating the `lem` reading and detailing its witnesses) and their attestations among the witnesses. (Situations where the `lem` reading is not duplicated by the first `rdg` element, but has its own `@wit` attribute, are also supported.) For conciseness, we use the `@n` attribute for each reading as a local identifier; the recommended practice for readings that will be referenced elsewhere is to use the `@xml:id` attribute, and this software will use this as the identifier if it is specified, but we have only specified `@xml:id` attributes for `rdg` elements referenced in other variation units to demonstrate the flexibility of the software. For witnesses with missing or ambiguous readings at a given variation unit, we use the `witDetail` element. For ambiguous readings, we specify their possible disambiguations with the `@target` attribute and express our degrees of certainty about these disambiguations using `certainty` elements under the `witDetail` element.

The TEI XML file for this example is available in the `example` directory of the GitHub repository. Full instructions for converting this file using `teiphy` and analyzing it with several different phylogenetic packages are provided in the documentation, but here, we will walk through the command-line arguments involved in installing `teiphy` and using it to convert our example TEI XML collation (1) to a NEXUS file suitable for use with IQ-TREE, and (2) to input for the STEMMA program.

Because `teiphy` is published in the Python Package Index (PyPI), it can be installed via the command

```
pip install teiphy
```

Now we are ready to convert our TEI XML collation to a NEXUS file for IQ-TREE. Let us suppose that we would like to treat reconstructions of unclear or missing text, defective spellings, and orthographic variations in spelling as trivial variants for the purposes of our phylogenetic analysis. We can specify this to `teiphy` with the `-t` flag for each trivial type of reading as follows:

```
-t reconstructed -t defective -t orthographic
```

In addition, suppose we would like to treat placeholders for overlapping variants from larger units and lacunae as missing data. We can specify this to `teiphy` with the `-m` flag for each type of reading to be read as missing data as follows:

```
-m overlap -m lac
```

If, in variation units where manuscripts are corrected or have alternate readings provided by other hands, our collation adds `*` and `T` suffixes to manuscript sigla to mark the work of the original hand, we can tell `teiphy` to ignore these suffixes using the `-s` flag with each trivial suffix:

```
-s"*" -s T
```

(Note that because the * character is reserved on the command-line, we must place it between quotation marks directly after the -s flag.)

When corrections are made to a manuscript, they are typically sporadic, and as a result, the text of corrector witnesses like 06C1 and 06C2 will tend to be too fragmentary to be useful for analysis. But if we wish to assume that each corrector tacitly adopted all of the readings from the previous hand that he or she did not change, then `teiphy` can "fill out" each corrector's text using the text of the first hand (in the case of the first corrector) or the filled-out text of the previous corrector (for all subsequent correctors). Thus, 06C1 would replicate the text of 06* (i.e., the first hand responsible for the text of 06) where it does not introduce its own readings, and 06C2 would then replicate the text of 06C1 where it does not introduce its own readings. If we want to apply this transformation during the conversion process, then we can specify this with the `--fill-correctors` flag.

Finally, we must specify the required arguments to `teiphy`, which are the input TEI XML file (`example/ubs_ephesians.xml`) and the name of the output NEXUS file (`ubs_ephesians-iqtree.nexus`). Note that we do not have to specify the desired output format explicitly; `teiphy` will determine from the output filename that it should write a NEXUS file. Combining the previous options and arguments, the complete command is

```
teiphy -t reconstructed -t defective -t orthographic -m overlap -m lac
-s"*" -s T --fill-correctors
example/ubs_ephesians.xml ubs_ephesians-iqtree.nexus
```

If we pass the resulting NEXUS file to IQ-TREE and specify appropriate settings for our textual data (in this case, the Lewis Mk substitution model with ascertainment bias correction), we will get an output tree like the one shown in Figure 1.

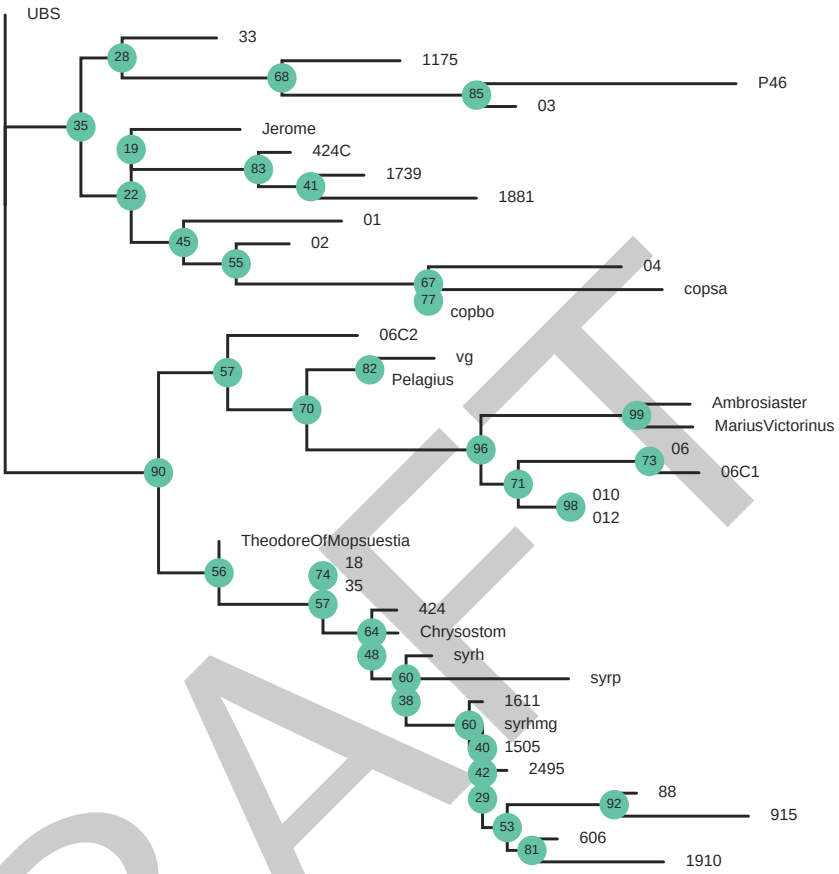

Figure 1: A phylogenetic tree inferred by IQ-TREE for the UBS Ephesians example data with support values on the branches based on 1000 bootstrap replicates. Reconstructed, defective, and orthographic sub-variants were treated as identical to their parent readings, and changes made to the text by later correctors (represented as distinct witnesses with sigla like 06C1 and 06C2) were filled in with the readings of the first hand or the previous corrector where the corrector was not active.

If we want to generate input files for the STEMMA program using the same options, only a few adjustments are required. First, since multiple files are written for STEMMA input (namely, a collation file with no file extension and a `.chron` file containing information about witness dates), we only specify the base of the filename for our output. Second, since the filename now has no extension, we must specify the desired output format to `teiphy` with the argument

```
--format stemma
```

Combining the options and arguments we used before with these changes, the complete command is

```
teiphy -t reconstructed -t defective -t orthographic -m overlap -m lac
-s"*" -s T --fill-correctors --format stemma
example/ubs_ephesians.xml stemma_example
```

If we process the output files with the PREP utility that accompanies STEMMA and then pass the resulting files to STEMMA, we will get an output tree like the one shown in Figure 2.

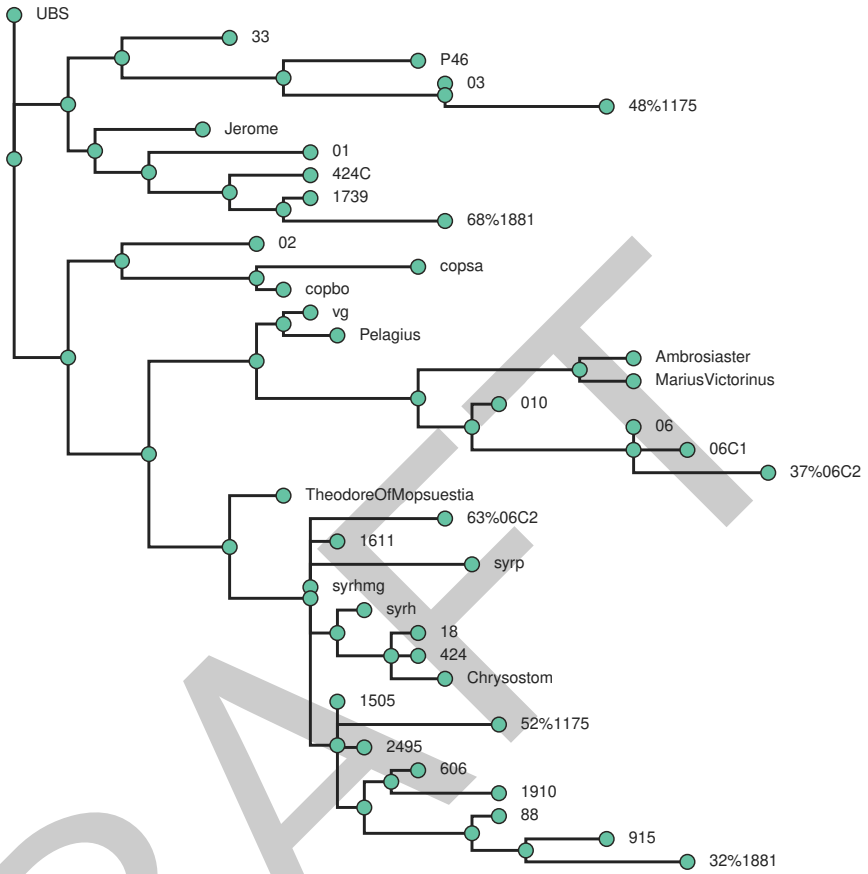

Figure 2: A phylogenetic tree inferred by STEMMA for the UBS Ephesians example data using 100 iterations of simulated annealing. Mixed witnesses are split (with proportions of their readings indicated by the percentages before their sigla) and located at different parts of the tree. Note that some witnesses (e.g., 012, 35) from the collation are excluded from this tree by STEMMA because they have the same reading sequence as another witness after their reconstructed, defective, and orthographic readings have been regularized.

For the small sample of variation units covered in the UBS apparatus for Ephesians, the phylogenetic results depicted in Figures 1 and 2 are impressive. The trees produced by IQ-TREE and STEMMA agree on several traditionally established groupings of manuscripts, including Family 1739 (1739, 1881, and the corrections to 424); the "Western" tradition (as preserved in the Greek-Latin diglots 06, 010, and 012, the Latin Vulgate, and the early Latin church fathers Ambrosiaster, Marius Victorinus, and Pelagius); and the later Byzantine tradition (with representative manuscripts 18 and 35 and church fathers Chrysostom and Theodore of Mopsuestia). The Harklean Syriac translation (syrh) and the witnesses to its Greek *Vorlage* (1505, 1611, 2495) are correctly placed within the Byzantine tradition, although the two programs disagree on how to describe their relationships within that tradition. While IQ-TREE does not account for mixture complicating the tradition, STEMMA identifies three witnesses suspected to exhibit Byzantine contamination: 1175, 1881, and the second corrector of 06. Both programs also identify the Codex Alexandrinus (02) as closely related to both the Sahidic and Bohairic Coptic translations of Ephesians (copsa, copbo), although they disagree on where this clade is located in the larger tradition. Despite their discrepancies regarding

certain subtrees, the extent of their agreements speaks to the level of genealogically significant detail preserved in the TEI XML apparatus and the NEXUS and STEMMA inputs generated from it.

Availability

As noted above, the software is published in PyPI and can be installed from there using `pip`. The source code is available under the MIT license from the GitHub repository. The automated testing suite has 100% coverage, and functional tests where our example TEI XML file is converted and run through RAxML, IQ-TREE, MrBayes, and STEMMA are part of `teiphy`'s continuous integration (CI) pipeline.

Acknowledgements and Funding

The authors wish to thank Stephen C. Carlson for his feedback during the development of `teiphy` and the *JOSS* reviewers for their thorough and insightful comments on earlier drafts of this work. This work was supported by an Australian Government Research Training Program (RTP) Scholarship.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., … Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems*. https://www.tensorflow.org/

Aland, B., Aland, K., Karavidopoulos, J., Martini, C. M., & Metzger, B. M. (Eds.). (2014). *The Greek New Testament* (5th ed.). Deutsche Bibelgesellschaft.

Baldwin, C. S. (2010). Factor analysis: A new method for classifying New Testament Greek manuscripts. *Andrews University Seminary Studies*, *48*(1), 29–53.

Barbrook, A. C., Howe, C. J., Blake, N., & Robinson, P. (1998). The phylogeny of *The Canterbury Tales*. *Nature*, *394*, 839. https://doi.org/10.1038/29667

Bouckaert, R., Vaughan, T. G., Barido-Sottani, J., Duchêne, S., Fourment, M., Gavryushkina, A., Heled, J., Jones, G., Kühnert, D., De Maio, N., Matschiner, M., Mendes, F. K., Müller, N. F., Ogilvie, H. A., du Plessis, L., Popinga, A., Rambaut, A., Rasmussen, D., Siveroni, I., … Drummond, A. J. (2019). BEAST 2.5: An advanced software platform for Bayesian evolutionary analysis. *PLOS Computational Biology*, *15*(4), 1–28. https://doi.org/10.1371/journal.pcbi.1006650

Carlson, S. C. (2015). *The text of Galatians and its history*. Mohr Siebeck. https://doi.org/10.1628/978-3-16-153324-2

Edmondson, A. C. (2019). *An analysis of the coherence-based genealogical method using phylogenetics*. University of Birmingham. http://etheses.bham.ac.uk/id/eprint/9150

Farris, J. S. (1988). *Hennig86, ver. 1.5. Program and documentation*. James S. Farris.

Felsenstein, J. (2004). *Inferring phylogenies*. Sinauer Associates.

Finney, T. J. (2018). How to discover textual groups. *Digital Studies/Le Champ Numérique*, *8*. https://doi.org/10.16995/dscn.291

Fischer, F. (2020). Representing the critical text. In P. Roelli (Ed.), *Handbook of stemmatology: History, methodology, digital approaches* (pp. 405–427). De Gruyter.

Goloboff, P. A., & Catalano, S. A. (2016). TNT, version 1.5, including a full implementation of phylogenetic morphometrics. *Cladistics*, *32*(3), 221–238. https://doi.org/10.1111/cla.12160

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*, 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hyytiäinen, P. (2021). The changing text of Acts: A phylogenetic approach. *TC: A Journal of Biblical Textual Criticism*, *26*, 1–28.

Ide, N. M., & Sperberg-McQueen, C. M. (1995). The TEI: History, goals and future. *Computers and the Humanities*, *29*(1), 5–15. https://doi.org/10.1007/978-94-011-0325-1_2

Maddison, D. R., Swofford, D. L., & Maddison, W. P. (1997). NEXUS: An extensible file format for systematic information. *Systematic Biology*, *46*(4), 590–621. https://doi.org/10.1093/sysbio/46.4.590

McCollum, J. (2019). Biclustering readings and manuscripts via non-negative matrix factorization, with application to the text of Jude. *Andrews University Seminary Studies*, *57*(1), 61–89.

McKinney, W. (2010). Data structures for statistical computing in Python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 56–61). https://doi.org/10.25080/Majora-92bf1922-00a

Minh, B. Q., Schmidt, H. A., Chernomor, O., Schrempf, D., Woodhams, M. D., von Haeseler, A., & Lanfear, R. (2020). IQ-TREE 2: New models and efficient methods for phylogenetic inference in the genomic era. *Molecular Biology and Evolution*, *37*(5), 1530–1534. https://doi.org/10.1093/molbev/msaa015

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(85), 2825–2830.

Robinson, P., & O'Hara, R. J. (1992). Report on the Textual Criticism Challenge 1991. *Bryn Mawr Classical Review*, *3*(4), 331–337.

Ronquist, F., Teslenko, M., van der Mark, P., Ayres, D. L., Darling, A., Höhna, S., Larget, B., Liu, L., Suchard, M. A., & Huelsenbeck, J. P. (2012). MRBAYES 3.2: Efficient Bayesian phylogenetic inference and model selection across a large model space. *Systematic Biology*, *61*(3), 539–542. https://doi.org/10.1093/sysbio/sys029

Salemans, B. J. P. (2000). *Building stemmas with the computer in a cladistic, neo-Lachmannian, way: The case of fourteen text versions of Lanseloet van Denemerken*. Katholieke Universiteit Nijmegen. https://hdl.handle.net/2066/147058

Spencer, M., Wachtel, K., & Howe, C. J. (2002). The Greek Vorlage of the Syra Harclensis: A comparative study on method in exploring textual genealogy. *TC: A Journal of Biblical Textual Criticism*, *7*. http://jbtc.org/v07/SWH2002/index.html

Spencer, M., Wachtel, K., & Howe, C. J. (2004). Representing multiple pathways of textual flow in the Greek manuscripts of the Letter of James using reduced median networks. *Computers and the Humanities*, *38*, 1–14. https://doi.org/10.1023/B:CHUM.0000009290.14571.59

Stamatakis, A. (2014). RAxML version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, *30*(9), 1312–1313. https://doi.org/10.1093/bioinformatics/btu033

Swofford, D. L. (2003). *PAUP*: Phylogenetic analysis using parsimony (*and other methods). Version 4*. Sinauer Associates.

TEI Consortium. (2022). *TEI P5: Guidelines for electronic text encoding and interchange: Critical apparatus [v.4.4.0]*. https://www.tei-c.org/release/doc/tei-p5-doc/en/html/TC.html.

The pandas development team. (2020). *Pandas-dev/pandas: pandas*. Zenodo. https://doi.org/10.5281/zenodo.3509134

Thorpe, J. C. (2002). Multivariate statistical analysis for manuscript classification. *TC: A Journal of Biblical Textual Criticism*, *7*. http://jbtc.org/v07/Thorpe2002.html

Turnbull, R. (2020). *The textual history of Codex Sinaiticus Arabicus and its family*. Ridley College.

Willker, W. (2008). *Principal component analysis of manuscripts of the Gospel of John*. http://www.willker.de/wie/TCG/PCA/index.html

Zitnik, M., & Zupan, B. (2012). NIMFA: A Python library for nonnegative matrix factorization. *Journal of Machine Learning Research*, *13*, 849–853.