International Journal of Digital Humanities (2019) 1:141–163
https://doi.org/10.1007/s42803-019-00012-w

RESEARCH ARTICLE

From graveyard to graph

Visualisation of textual collation in a digital paradigm

Elli Bleeker[1] · Bram Buitendijk[1] · Ronald Haentjens Dekker[1]

Published online: 19 June 2019

Abstract
The technological developments in the field of textual scholarship lead to a renewed focus on textual variation. Variants are liberated from their peripheral place in appendices or footnotes and are given a more prominent position in the (digital) edition of a work. But what constitutes an informative and meaningful visualisation of textual variation? The present article takes visualisation of the result of collation software as point of departure, examining several visualisations of collation output that contains a wealth of information about textual variance. The newly developed collation software HyperCollate is used as a touchstone to study the issue of representing textual information to advance literary research. The article concludes with a set of recommendations in order to evaluate different visualisations of collation output.

Keywords Collation software · Textual scholarship · Visualisation · Markup · Hypergraph for variation · Tool evaluation

1 Introduction

Scholarly editors are fond of the truism that the detailed comparison ('collation') of literary texts is a tiresome, error prone, and demanding activity for humans and a task suitable for computers. Accordingly, the past decades have born witness to the development of a number of software programs which are able to collate large numbers of text within seconds, thus advancing significantly the possibilities for textual research. These developments have led to a renewed focus on textual variation, liberating variants from their peripheral place in appendices or footnotes and giving them a more prominent position in the edition of a work. Still, automated collation continues to engross researchers and developers, as it touches upon universal topics including (but not limited to) the computational modelling of humanities objects, scholarly editing

✉ Elli Bleeker
elli.bleeker@di.huc.knaw.nl

[1] Research and Development – KNAW Humanities Cluster, Amsterdam, Netherlands

theory, and data visualisation. The present article takes visualisation of collation result as its point of departure. We use the representation of the results of a newly developed collation tool, 'HyperCollate', as a use case to address the more general issue of using data visualisations as a means of advancing textual and literary research. The underlying data structure of HyperCollate is a hypergraph (hence the name), which means that it can store and process more information than string-based collation programs. Accordingly, HyperCollate's output contains a wealth of detailed information about the variation between texts, both on a linguistic/semantic level and a structural level. It is a veritable challenge to visualise the entire collation hypergraph in any meaningful way, but the question is, really, do we want to? In particular, therefore, we investigate which representation(s) of automated collation results best clear the way for advanced research into textual variance.

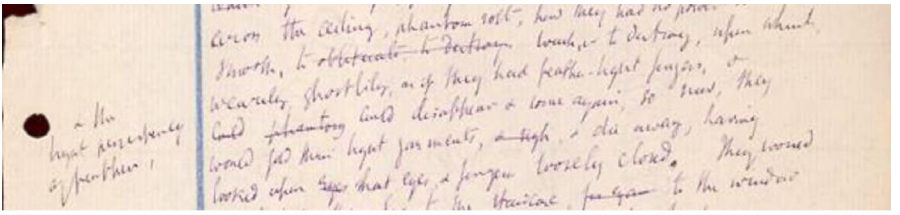
The article is structured as follows. After a brief introduction of automated collation immediately below, we define a list of textual properties relevant for any study into the nature of text. We then consider the strengths and weaknesses of the prevailing representations of collation output, which allows us to define a number of requirements for a collation visualisation. Subsequently, the article explores the question of visual literacy in relation to using a collation tool. Since visualisations function simultaneously as instruments of study and as means of communication, it is vital they are understood and used correctly. In line with the idea of visual literacy, we conclude with a number of recommendations to evaluate the visualisations of collation output. The implications of creating and using visualisations to study textual variance are discussed in the final parts of the article. Before we go on, it is important to note that we define 'textual variance' in the broadest sense: it comprises any differences between two or more text versions, but also the revisions and other interventions within one version. Indeed, we do not make the traditional distinction between 'accidentals' and 'substantives'. This critical distinction is the editor's to make, for instance by interpreting the output of a collation software program.

2 Automated collation

Collation at its most basic level can be defined as the comparison of two or more texts to find (dis)similarities between or among them. Texts are collated for different reasons, but in general, collation is used to track the (historical) transmission of a text, to establish a critical text, or to examine an author's creative writing process. Traditionally, collation has been considered an auxiliary task: it was an elementary part of preparing the textual material in order to arrive at a critically established text and not necessarily a part of the hermeneutics of textual criticism. The reader was presented with the end-result of this endeavour (a critical text), and the variant readings were stored in appendices or footnotes, the kind of repositories that would get so few visitors that they have been bleakly referred to as cemeteries (Vanhoutte 1999; De Bruijn 2002, 114). In the environment of a digital edition, however, users can manipulate transcriptions which are prepared and annotated by editors. Many digital editions have a functionality to compare text versions and, accordingly, collation has become a scholarly primitive, like searching and annotating text. The digital representation of the result of the comparison thus brings textual variants to the forefront instead of (respectfully) entombing them.

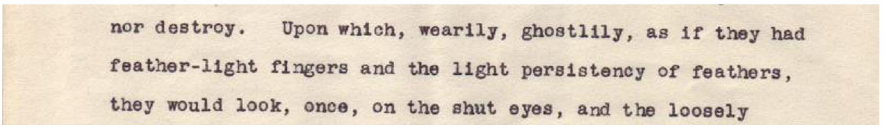
3 Properties of text

It's important to note that offering users the opportunity to explore textual variance in a digital environment is an argument *an sich*: it stresses that text is a fluid and intrinsically unstable object. And, as anyone who has worked with historical documents knows, these fluid textual objects often have complex properties, such as discontinuity, simultaneity, non-linearity, and multiple levels of revision.[1] The dynamic and temporal nature of textual objects means that they can be interpreted in more than one way but existing markup systems like TEI/XML can never fully express the range of textual and critical interpretations.[2] Nevertheless, the benefits of 'making explicit what was so often implicit … outweighed the liabilities' of the tree structure (Drucker 2012), and as it happens, the textual scholarship community has embraced TEI/XML as a means of



Witness 1 (IHD-155):

```
<text>
    <p>
      <s>to <del>obliterate, to destroy</del> touch, or to destroy, upon
      which, wearily, ghostlily, as if they had feather-light fingers, <add>&
      the light persistency of feathers</add> & <del>could phantom</del>
      could disappear & come again</s>
    </p>
</text>
```



Witness 2 (TS-4):

```
<text>
    <p>
      <s>neither touch nor destroy.</s>
      <s>Upon which, wearily, ghostlily, as if they had feather-light fingers
and the light persistency of feathers, they would look, once,</s>
    </p>
</text>
```

[1] See Haentjens Dekker and Birnbaum (2017) for an exhaustive overview of textual features and the extent to which these can be represented in a computational model.

[2] The TEI Guidelines offer the element <cert> to indicate the degree of certainty associated with some aspect of the text markup, but as Wout Dillen points out, this requires an elaborate encoding practice that is not always worth the effort (2015, 90) and furthermore the ambiguity is not always translatable to the qualifiers "low," "medium," and "high."

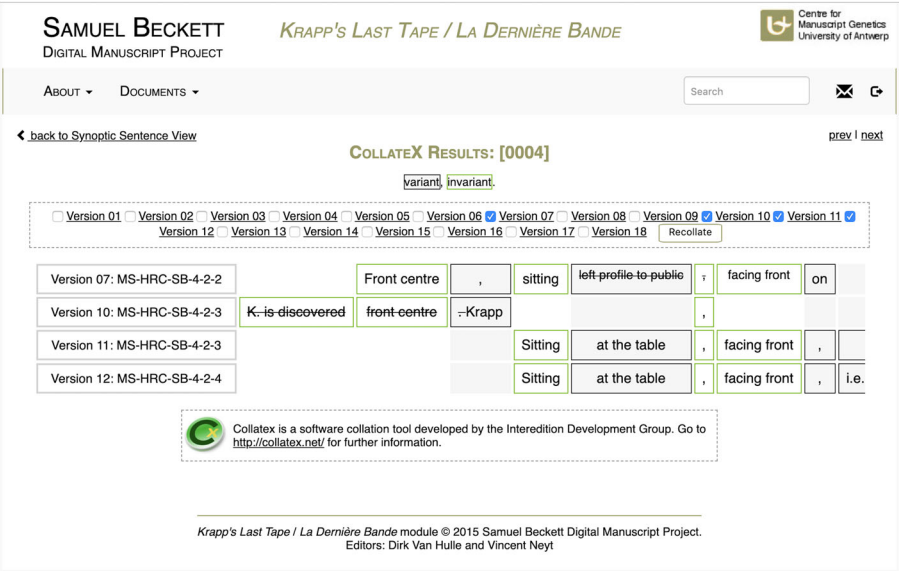


Fig. 1 Example of an alignment table visualisation of a collation of four versions of Samuel Beckett's *Krapp's Last Tape* which visualises the deleted words as strike-through. The collation was performed by CollateX

encoding literary texts. Expressing the multidimensional textual object within a tree data structure (the prevalent model for texts) requires a number of workarounds and results in an encoded XML transcription which contains neither fully ordered nor unordered information (Bleeker et al. 2018, 82). This kind of partially ordered data is challenging to process. As a result, XML files are often collated as strings of characters, inevitably leaving out aspects of the textual dynamics such as deletions, additions or substitutions. The conversion from XML to plain text implies that the multidimensional features of the text expressed by <del> and <add> tags are removed; the text is consequently flattened into a linear sequence of words. Only in the visualisation stage of the collation workflow do features like additions or deletions occur again (Fig. 1).

Although these versions of *Krapp's Last Tape* are compared on the level of plain text only, the alignment table in Fig. 1 also shows the in-text variation of witnesses 07 and 10, thus neatly illustrating the informational role of visualisations. The main objective for the development of the collation engine HyperCollate was to include textual properties like in-text variation in the alignment in order to perform a more inclusive collation and to facilitate a deeper exploration of textual variation. A look at the drafts of Virginia Woolf's *Time Passes*[3] offers a good illustration of some textual features we'd like to include in the automated collation. For reasons of clarity, we limit the collation input to two small fragments: the initial holograph

[3] Woolf, Virginia. *Time Passes*. The genetic edition of the manuscripts is edited by Peter Shillingsburg and available at www.woolfonline.com (last accessed on 2018, April 27). Excerpts from Woolf's manuscripts are reused in this contribution with special acknowledgments to the Society of Authors as the Literary Representative of the Estate of Virginia Woolf.

draft ‘IHD-155’ (witness 1) and the typescript ‘TS-4’ (witness 2). Both fragments are manually transcribed in TEI/XML. The transcriptions below are simplified for reasons of legibility.

A quick look at these fragments reveals that they contain linguistic variation between tokens with the same meaning as well as structural variation indicated by the markup. Here, the ampersand mark ‘&’ in witness 1 and the word token ‘and’ in witness 2 constitute linguistic variation: two different tokens with the same meaning. Furthermore witness 1 presents a case of in-text or intradocumentary variation: variation *within* a witness’ text (see also Schäuble and Gabler 2016; Bleeker 2017, 63). If we look at the revision site that is highlighted in the XML transcription of witness 1, we see several orders in which we can read the text: including or excluding the added text; including or excluding the deleted text. In other words, there are multiple ‘paths’ through the text,: the textualstream diverges at the point where revision occurs, indicated by the <del> element and the <add> element. When the text is parsed, the textual content of these different paths should be considered as being on the same level: they represent multiple, co-existing readings of the text. Intradocumentary variation can become highly complex, for instance in the case of a deletion inside a deletion inside a deletion, etc. The structural variation in this example becomes manifest if we compare the two witnesses: the excerpt in witness 1 is contained by one <s> element, while the phrase in witness 2 is contained by two <s> elements. However structural variation does not only occur across documents: when an author indicates the start of a new chapter or paragraph by inserting a metamark of some sorts, this is arguably a form of structural intradocumentary variation.

To summarise, we can distinguish different forms of textual variance. Variation can occur on the level of the text characters (linguistic or semantic variation) and on the structure of the text (sentences, paragraphs, etc.). Furthermore, we distinguish between intradocumentary variation (within one witness) and interdocumentary variation (across witnesses). Arguably all forms are relevant for textual scholarship, but taking them into account when processing and comparing texts has both technical and conceptual consequences. These consequences have been discussed extensively elsewhere (Bleeker et al. 2018) and will be briefly repeated in section 5 below. The main goal of the present article is to focus on the question of visualisation. Assuming we have a software program that compares texts in great detail, including structural information and in-witness revisions, how can we best visualise its ouput? first and foremost, The additional information (structural and linguistic, intradocumentary and interdocumentary) needs to be visualised in an understandable way. The visualisations can be useful for a wide range of research objectives, such as (1) finding a change in markup indicating structural revision like sentence division, (2) presenting the different paths through one witness and the possible matches between tokens from any path, (3) complex revisions, like a deletion within a deletion within an addition, (4) studying patterns of revision, and so on. This begs the question: is it even possible or desirable to decide on one visualisation? Is there one ultimate visualisation that reflects the dynamic, temporal nature of the textual object(s) by demonstrating both structural and linguistic variation on an intradocumentary and interdocumentary level? the existing field of Information Visualisation can certainly offer inspiration, but simply adopting its methods and techniques will not suffice, since it deals primarily with objects which are

'self-identical, self-evident, ahistorical, and autonomous' (Drucker 2012), adjectives which could hardly be applied to literary texts.

4 Existing Visualisations of collation results

Let us consider the various existing visualisations of collation output and explore to what extent they address the conditions outlined above. We can distinguish roughly five types of visualisation: alignment tables, parallel segmentation, synoptic viewers, variant graphs, and phylogenetic trees or 'stemmata'. A smaller example of a collation of two fragments from Woolf's *A Sketch of the Past* (holograph MS-VW-SoP and typescript TS1-VW-SoP) serves as illustration of the effect of the visualisations:

Witness 1 (MS-VW-SoP): with the boat train arriving, people talking loudly, chains being dropped, and the screws ~~the~~ beginning, and the steamer suddenly hooting

Witness 2 (TS1-VW-SoP): with the boat train arriving; with people quarrelling outside the door; chains clanking; and the steamer giving those sudden stertorous snorts

These two small fragments are transcribed in plain text format and subsequently collated with the software program CollateX. Unless indicated otherwise, the result from this collation forms the basis for the visualisation examples below.

4.1 Alignment table

An alignment table presents the text of the witnesses in linear sequence (either horizontally or vertically), making it well-suited to a study of the relationships between witnesses on a detailed level, but less so to acquire an overview of patterns in revision. Note that 'aligned tokens' are not necessarily the same as 'matching tokens': two tokens may be placed above each other because they are at the same relative position between two matches, even though they do not constitute a match. For this reason, alignment tables often have additional markup (e.g. colours) to differentiate between matches and aligned tokens. The arrangement of the tokens is also one of the advantages of an alignment table: it shows at first glance the variation between tokens at the same relative position. In other words, this representation indicates tokens which match on a semantic level, such as synonyms or fragments with similar meanings, such as 'talking loudly' and 'quarrelling outside the door' (Fig. 2).

Ongoing research into the potential of an alignment table visualisation to explore intradocumentary variation (see Bleeker et al. 2017, visualisations created by Vincent Neyt) focuses on increasing the amount of information in an alignment table by incorporating intradocumentary variation in the cells. The alignment table in Fig. 3 shows that witness 1 (Wit1) contains several paths; matching tokens are displayed in red.

Fig. 2 Example of alignment table visualisation of 'MS1-VW-SoP' (W1) collated against 'TS1-VW-SoP' (W2) which, again, shows how synonyms which do not match are aligned anyway because of the matching tokens which surround them. Table generated by CollateX

4.2 Synoptic viewers

A synoptic edition contains a visual representation of the collation results from the perspective of one witness, where the variants are indicated by means of a system of signs or diacritical marks. In contrast to an alignment table, a synoptic overview is more suitable as an overview examination of the patterns of variation. The following paragraphs discuss two ways of presenting textual variation synoptically: parallel segmentation and an inline apparatus. It may be clear that both are skeuomorphic in character, in the sense that they mimic the analogue examination and presentation of textual variants. This characteristic should not necessarily be considered negative, however, precisely because it is a tried and tested instrument for textual research.

4.2.1 Parallel segmentation

The term 'parallel segmentation' may be confusing, as it is also the name of the (TEI) encoding for a critical apparatus. In this context, parallel segmentation is used to describe the visualisation of textual variation in a side-by-side manner, often with the corresponding segments linked to one another. The quantity of online, open source tools for a parallel segmentation visualisation suggests that it is a popular way of studying textual variation (e.g. the Versioning Machine,[4] the Edition Visualisation Technology – EVT – project,[5] and the visualisation of Juxta Commons).[6] As Fig. 4 shows, parallel segmentation entails presentation of the witnesses as reading texts in separate panels which can be read vertically (per witness) or horizontally (interdocumentary variation across witnesses). Colours indicate the matching and non-matching segments.

To be clear: this parallel segmentation visualisation concerns the *presentation* of variance; it is not a collation method in and of itself. The segments are encoded by the editor, for instance using the TEI <app>/<lemm>/<rdg> construction to link matching segments. In contrast to the inline apparatus presentation (see 2b below), which uses a base text, parallel segmentation presents the witnesses are presented as variations on one another. Most tools allow for an interactive visualisation in the sense that clicking on a segment in one witness highlights the corresponding segments in the other witness(es). As represented in Fig. 4, the parallel segmentation may also visualise

[4] See http://v-machine.org/ (last accessed 2018, March 30).
[5] Downloadable on https://sourceforge.net/projects/evt-project/files/latest/download (last accessed 2018, March 30)
[6] See http://www.juxtasoftware.org/juxta-commons/ (last accessed 2018, March 30).

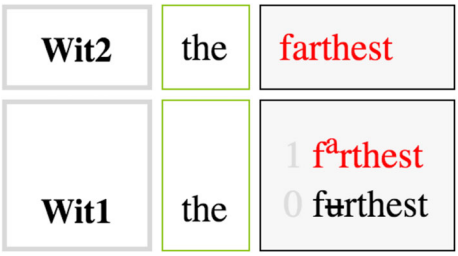


Fig. 3 Alignment table visualisation showing intradocumentary variation in witness 1. The colour red is used to draw attention to the matching tokens, which is especially useful in the case of more or longer witnesses

intradocumentary variation by rendering deletions and additions (embedded in the corresponding <rdg> by means of <del> and <add> elements).

4.2.2 Critical or inline apparatus

Conventionally, an apparatus accompanies a critically established text which figures as a base text. The apparatus is made up of a set of notes containing variant readings, often recorded in some shorthand using diacritical signs, witness sigli, and some context. Variant readings encoded according to the TEI guidelines can be generated as said footnotes, or the reader can select certain readings to be displayed/ignored. Alternatively, an inline apparatus entails a synoptic visualisation of the variant readings in the form of diacritical marks *inside* a reading text. This kind of synoptic overview can draw the reader's attention to the places in the text that underwent heavy revisions. A classic example of a synoptic visualisation is found in the *Ulysses* edition (Joyce 1984–1986), a presentation format which Hans Walter

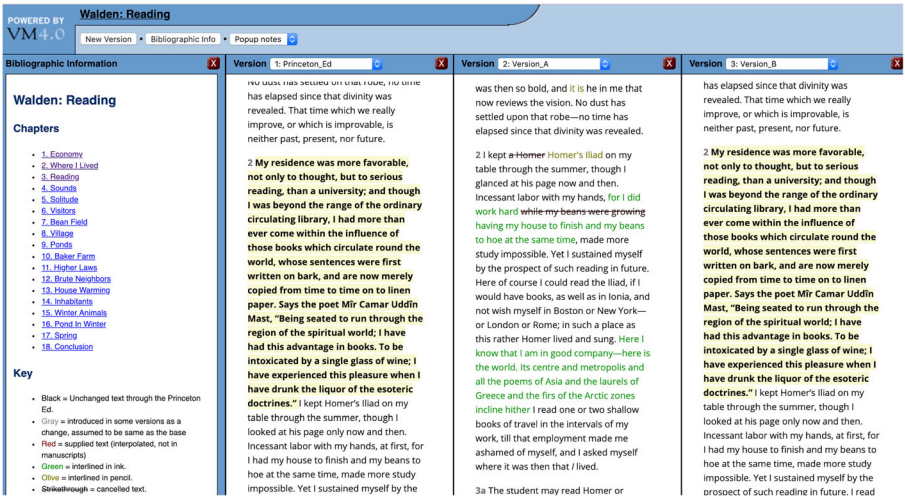


Fig. 4 Screen capture of the parallel segmentation visualisation of the Versioning Machine output of three versions of *Walden* (Henry David Thoreau): the base text of the Princeton edition, manuscript Version A, and manuscript Version B. The witnesses are displayed side-by-side, with cancelled text in witness Version A represented by strikethrough, added text by green, and matching text by highlight. In this example, the collation has been carried out manually and transcribed according to the TEI Parallel Segmentation method (Schacht 2016. 'Introduction')

Gabler and Joshua Schäuble recently repeated digitally with the Diachronic Slider (Schäuble and Gabler 2016; Fig. 5). The clear advantage of a *digital* synoptic edition is that the diacritical signs can be replaced with visual indications which have a lower readability threshold than diacritical marks, such as different colours or a darker shade behind the tokens that vary in other witnesses (cf. the *Faust* edition).

4.3 Variant graph

A variant graph is a collection of nodes and edges. It is to be read from left to right, top to bottom, following the arrows. This reading order makes it a directed acyclic graph (DAG): it can be read in one order only, without 'looping' back. In some visualisations, the text tokens are placed on the edges (e.g. Schmidt and Colomb 2009); in others, they are placed in the nodes (e.g. CollateX; Fig. 6). In contrast to the alignment table, there is no 'visual alignment' in the variant graph: matching tokens are merged. Only the variant text tokens are made explicit; witness sigla indicate which tokens belong to which witness. By following a path over nodes and edges, users can read the text of a specific witness and see where it corresponds with and diverges from other witnesses. One of the main advantages of a variant graph is that it doesn't impose one single order: in the visualisation, no path through the text is preferred over the other. The variant graph thus facilitates recording and structuring non-linear structures in manuscript texts, making it easier to visualise layers of writing without preferring one over the other. Because the variant graph is capable of including more information than for instance an alignment table, it is a useful visualisation with which to analyse the collation outcome in detail.

The vertical or horizontal direction of the variant graph depends on the tool or the preference of the user. Horizontally oriented variant graphs imitate to some extent the Western reading orientation (from left to right), while variant graphs that are vertically situated appear to anticipate the reading habits of 'homo digitalis' (from top to bottom). In both cases, longer witnesses result in endless scrolling and a loss of overview. This was reason for the TRAViz project to insert line breaks based on the assumption that

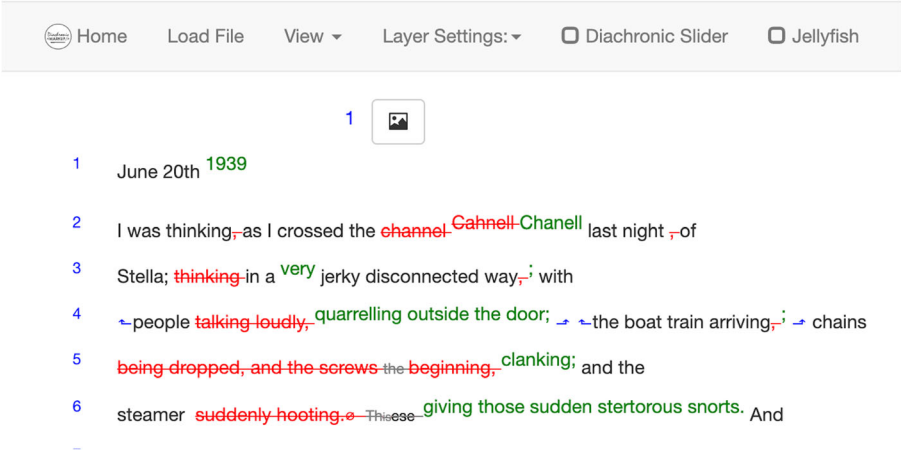


Fig. 5 Visualisation of the inline apparatus of the Diachronic Slider of 'MS1-VW-SoP' collated against 'TS1-VW-SoP'. The text from 'MS1-VW-SoP' are visualised in red; the green text is of 'TS1-VW-SoP'. The coloured visualisation replaces the traditional diacritical signs

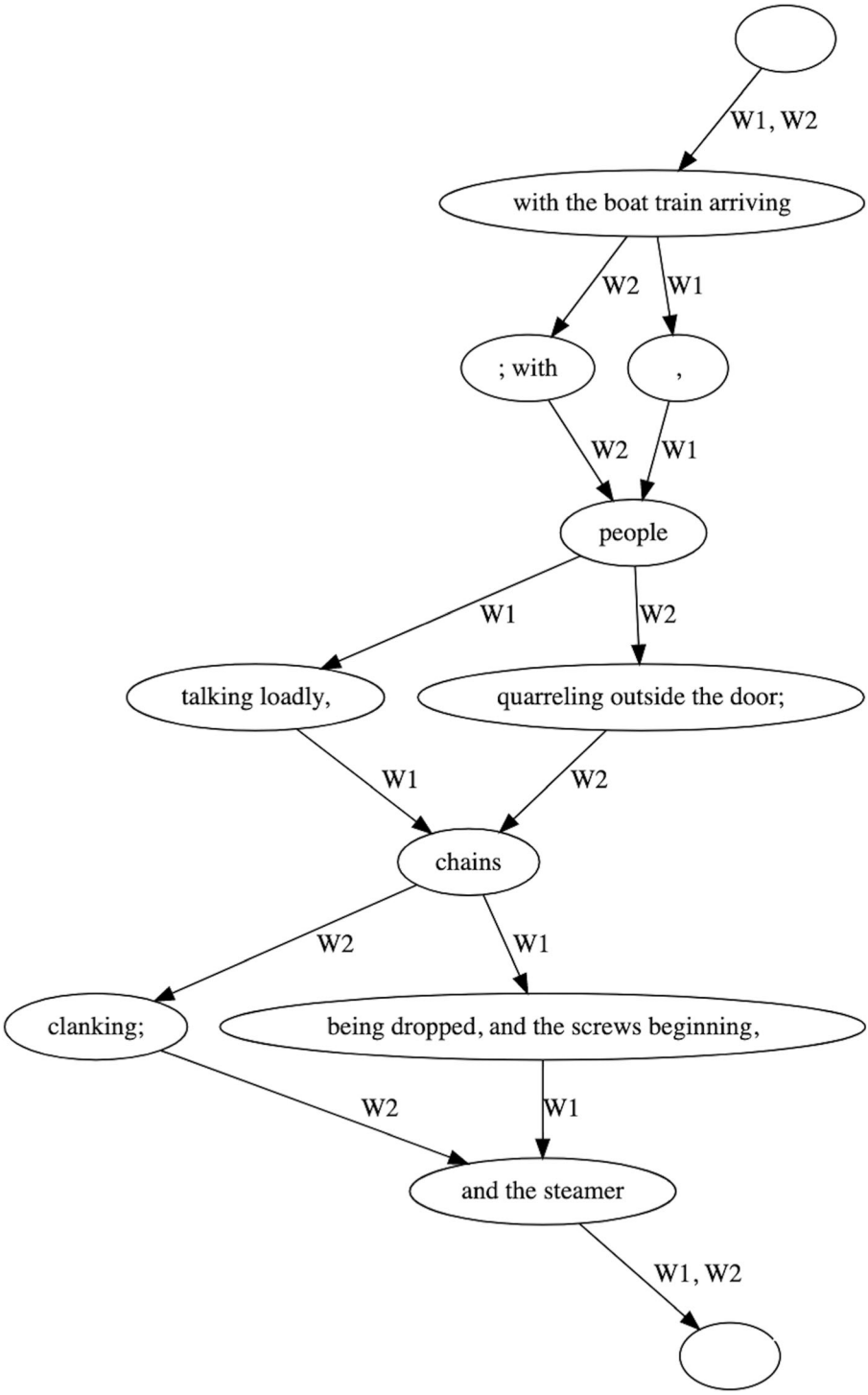


Fig. 6 Vertical variant graph visualisation of the comparison between 'MS1-VW-SoP' (W1) and 'TS-VW-SoP' (W2). Graph generated with CollateX

online readers prefer vertical scrolling but also like to be reminded that the text in the variant graph derives from a codex format (Jänicke et al. 2014; Fig. 7).

The variant graphs of CollateX in the figures directly above are non-interactive by design (since they are visual renderings of a collation output). However, the usefulness of interactive visualisations has been positively noted in several contributions (e.g., Andrews and Van Zundert) and projects. TRAViz, for instance, lets users interact with the graph and adjust it to match their needs and interests, and the variant graphs generated by the Stemmaweb tool set[7] allow for their nodes to be connected, input to be adjusted, and edges to be annotated with additional information about the type of variance. Such features emphasise the visualisation's double function as a means of communication and a scholarly instrument: on the one hand, it allows the user to clarify and communicate her argument about textual variation. On the other, the possibility of adjusting the visualisation and thus the representation of variation foregrounds the idea that the output of a tool is open to interpretation.

4.4 Phylogenetic trees or stemmata

One final type of visualisation is the phylogenetic tree (also known as 'stemma codicum' or 'stemmata'). Stemmata are not a collation method: they are created by the scholar or generated based on collation output like alignment tables or variant graphs. For that reason, stemmata do not directly concern the visualisation of collation output, primarily because the phylogenetic tree is used to store and explore the relationships between *witnesses* (and not between tokens). Nevertheless, this kind of tree provides a valuable perspective on visualising textual variation on a macro level: even at first glance, the tree conveys a good deal of information. The arrangement of the nodes within a stemma is meaningful; nodes close together in the stemma imply a high similarity between the witnesses. Each node in a tree represents a witness, and the edges which connect the nodes represent the process of copying one witness to another (a process sensitive to mistakes and thus variation). Stemmata are traditionally rooted, the witness represented as root being the 'archetype', which implies that all witnesses derive from one and the same manuscript (Fig. 8). More recently unrooted trees have

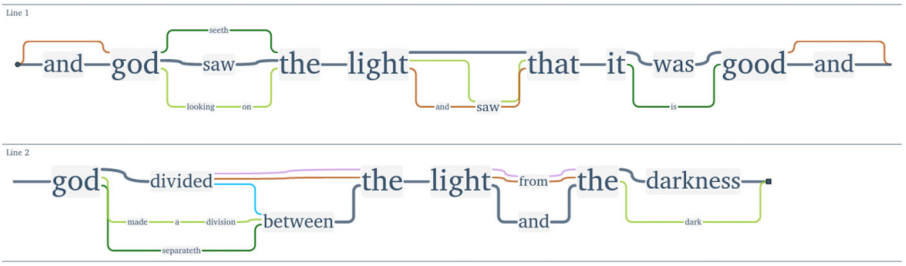


Fig. 7 Screen capture of the TRAViz variant graph visualization of a collation of Genesis 1:4. The size of the text indicates its presence in the witnesses

[7] Stemmaweb brings together several tools for stemmatology: https://stemmaweb.net/ (last accessed on 2018, April 27).

Witness 1 (IDH-157):

```
<TEI>
    <text>
        <div type="chapter" n="1">
            <p>
                <s>sliding from <del>corn</del> dresser to carpet.</s>
                <add place="margin">
                    <head>2</head>
                </add>
                <s>Further, what is one night?</s>
            </p>
        </div>
    </text>
</TEI>
```

Witness 2 (TS-5)

```
<TEI>
    <text>
        <div type="chapter" n="2">
            <p>
                <s>and washing away in water.</s>
            </p>
        </div>
        <div type="chapter" n="3">
            <head>III</head>
            <p>
                <s>But what after all, is one night?</s>
            </p>
        </div>
    </text>
</TEI>
```

been introduced that do not assume one 'ancestor' or archetype witness and simply represent relationships between witnesses (Fig. 9).[8]

A visualisation method similar to (and probably inspired by) stemmata or phylogenetic trees is the genetic graph in which the genetic relationships between documents related to a work are modelled (see Burnard et al. 2010, §4.2; Fig. 10). Nodes represent documents; the edges may be typed to indicate the exact relationship between documents (e.g. 'influence'), and they are usually directed so as to convey the chronology of the text's chronological development. A genetic graph is also not a direct visualisation of collation output, but a visual representation of the editor's argument about the text's development and her construction of the genetic dossier. With this overview representation, the editor may point to the existence of textual fragments like *paralipomena*, which were previously ignored or delegated to footnotes, critical apparatuses, or separate publications.

The kind of macrolevel visualisations provided by stemmata or genetic graphs present the necessary overview and invite more rigorous exploration. Diagrams, graphs, or coloured squares add new perspectives to the various ways in which we look at text.

[8] The Stemmaweb toolset allows users to root and reroot their stemmata to explore different outcomes, see https://stemmaweb.net/?p=27 (last accessed 2018, March 25).

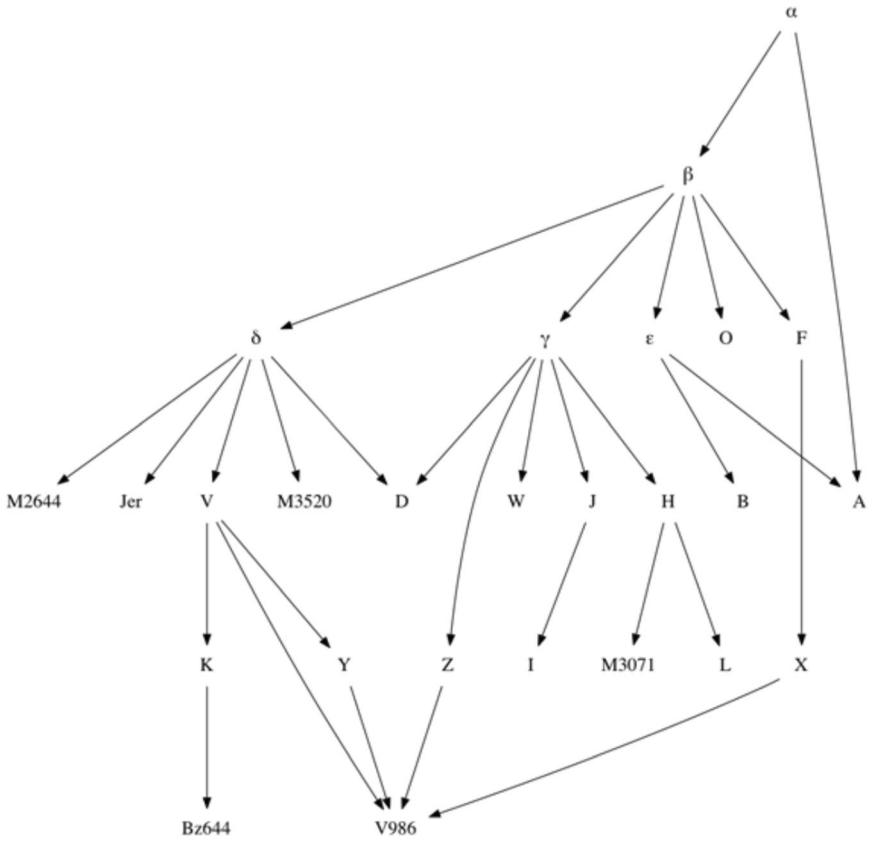


Fig. 8 A complex stemma in the form of a rooted directed acyclic graph (DAG), with the α in the top right corner representing the archetype witness from which other witnesses may derive (source: Andrews and Mace 2012)

5 HyperCollate

HyperCollate, a newly developed collation tool at the R&D department of the Humanities Cluster of the Dutch Royal Academy of Science, examines textual variation in an inclusive way using a hypergraph model for textual variation. HyperCollate is an implementation of TAG, the data model also developed at the R&D department (Haentjens Dekker and Birnbaum 2017). A discussion of the collation tool's technical specifications is not within the scope of the present article (see Bleeker et al. 2018); for now, it suffices to know that a hypergraph differs from traditional graphs, the edges of which can connect only two nodes with each other, because the edges in a hypergraph can connect more than two nodes with one another. These 'hyperedges' connect an arbitrary set of nodes, and the nodes in turn can have multiple hyperedges. Conceptually, then, the hyperedges in the TAG model can be considered as multiple layers of markup/information on a text. The hypergraph for variation used by HyperCollate is an evolved model based on the variant graph. By treating texts as a network, HyperCollate is able to process intradocumentary variation and store multiple hierarchies in an idiomatic manner. In other words, because HyperCollate doesn't require TEI/XML

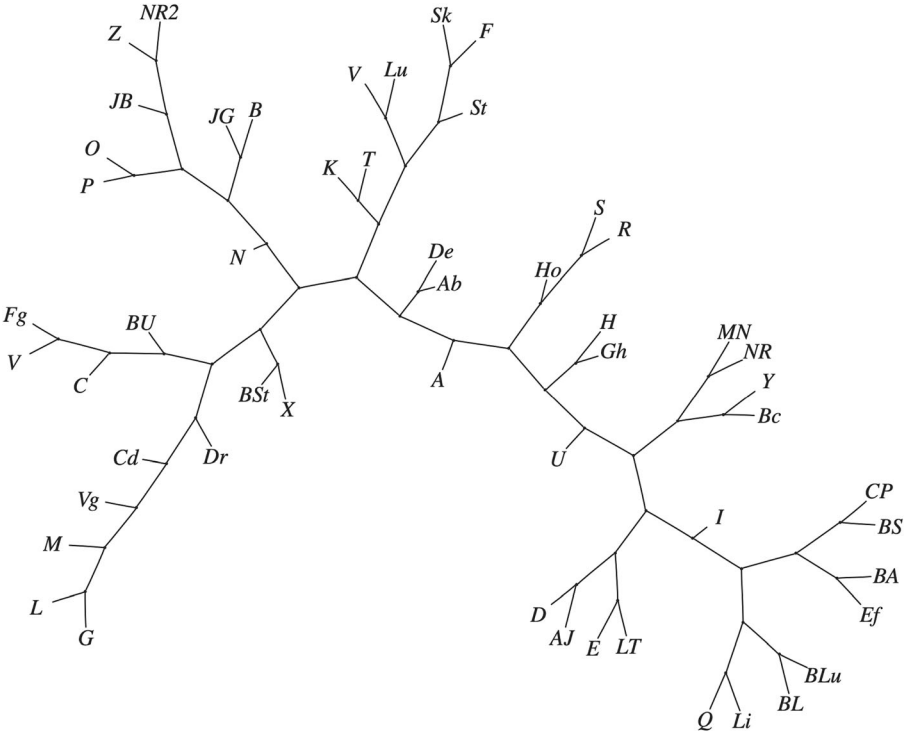


Fig. 9 Example of an unrooted phylogenetic tree (source: Roos and Heikkilä 2009)

transcriptions to be transformed into plain text files, TEI tags indicating revision like <del> and <add> can be used to improve the collation result. HyperCollate accordingly uses valuable intelligence of the editor expressed by markup to improve the alignment of witnesses.

Since the internal data model of HyperCollate is a hypergraph, the input text can be an XML file and doesn't need to be transformed into plain text. The comparison of two data-centric XML files is relatively simple, and it is even a built-in of the oXygen XML

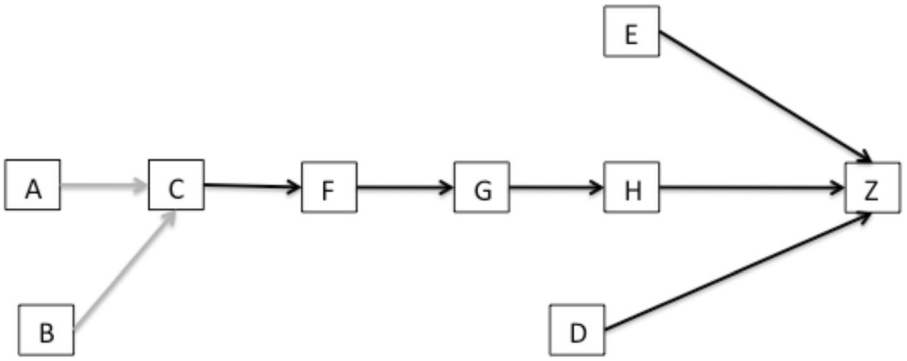


Fig. 10 Possible genetic graph visualisation proposed by the TEI Workgroup on Genetic Editions (Burnard et al. 2010), with the nodes A to Z representing different documents in the genetic dossier of a hypothetical work

editor, but as explained above, a typical TEI-XML transcription of a *literary* text with intradocumentary variation constitutes partially ordered information. In order to process this kind of information, HyperCollate first transforms the TEI-XML witnesses into separate hypergraphs and then collates the hypergraphs. Graph-to-graph collation ensures that the input text can be processed taking into account both the textual content and the structure of the text. For each witness, HyperCollate looks at the witness' text, the different paths through the witness' text, and the structure of the witness, and subsequently compares the witnesses on all these levels. Accordingly, the output of HyperCollate contains a plethora of information. Similar to CollateX,[9] a widely used text collation tool, the output of HyperCollate could be visualised in different ways (e.g., an alignment table or a variant graph). By default, HyperCollate's output is visualised as a variant graph, primarily because a variant graph does not have a single order so it is relatively straightforward to represent the different orders of the tokens as individual paths. The question is, how (and where) to include the additional information in the visualisations? A variant graph may be more flexible regarding the token order, but the nodes and edges can only contain so much extra information, as Fig. 12 below shows.

A favourable consequence of HyperCollate is that, in case of intradocumentary variation, each path through a witness is considered equally important. This feature is in stark contrast with current approaches to intradocumentary variation, which usually entail a manual selection of one revision stage per witness (see Bleeker 2017, 110–113). By means of illustration, let us take a look at another collation of two small fragments from Woolf's *Time Passes* containing intradocumentary structural variation. The fragments are manually transcribed in TEI/XML and simplified for reasons of clarity. The XML files form the input of HyperCollate.

Witness 1 contains an interesting addition (highlighted): Woolf added a metamark and the number '2' in the margin. The transcriber interpreted the added number as an indication that the running text should be split up and a new chapter should be started, so she tagged the number with the <head> element.[10] This means that the tokens of this witness can be ordered in two ways: excluding the addition and including the addition. Furthermore, the <head> element in witness 1 is at the same relative position as the <head> element in witness 2, so that the two headers are a match (even though their content is not).

Figure 11 shows the variant graph visualisation of the output. Note that the paths through the witnesses can be read by following the witness sigli on the edges (w1, w1:add, w2); the markup <head> is represented as a 'hyperedge'[11] on the text nodes:

An alternative way of representing HyperCollate's output in a variant graph is by enclosing both linguistic and structural information within the text nodes (Fig. 12).

The visualisations of the collation hypergraph in Figs. 11 and 12 represent the collation output of two small and simplified witnesses. It may be clear that collating two larger TEI/XML transcriptions of literary text, each containing several stages of revisions and multiple layers of markup, results in a collation hypergraph that, in its entirety, cannot

[9] Haentjens Dekker, Ronald and Gregor Middell. CollateX. https://collatex.net/.

[10] Arguably the transcriber could have added a , but the TEI Guidelines do not allow for a to be placed within an <add>. Nevertheless, contrasting the structure of witness 1 with the structure of witness 2 already alerts the reader to structural revisions and invites a closer inspection.

[11] The edges in a hypergraph are called hyperedges. In contrast to edges in a DAG, hyperedges can connect a set of nodes.

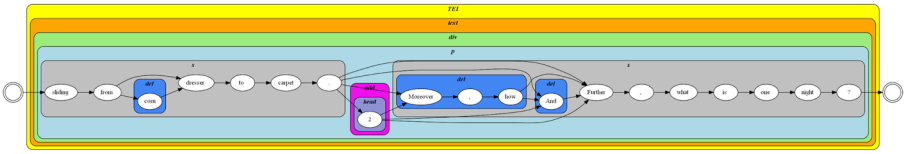


Fig. 11 Alternative, black-and-white visualisation of HyperCollate output, with the <head> markup represented as hyperedge on the nodes. Other markup is not visualised

be visualised in any meaningful way. At the same time, the various types of information contained by the collation hypergraph are of instrumental value to a deeper study of the textual objects. For that reason, HyperCollate offers not one specific type but rather lets the user select from a wide variety of visualisations, ranging from alignment tables to variant graphs. In selecting the output visualisation, the user decides which information she prefers to see and which information can be ignored. She may consider an alignment table if she's primarily interested in the relationships between witnesses on a microlevel, or a variant graph if an insightful overview of the various token orders is more relevant to her research. Furthermore, she may decide what markup layers she want to see: arguably knowing that every token is part of the root element 'text' is of less concern than detecting changes in the structure of sentences. Making such decisions does require the user to have a basic knowledge of the underlying dataset and a clear idea of what she's looking for.

6 Requirements for visualising textual variance

This overview allows us to draw a number of conclusions regarding the visualisation of textual variation and to what extent each visualisation considers the various dimensions of the textual object. We have seen that intradocumentary variation is as of yet not represented by default; the editor is required to make certain adjustments to the visualisation. Alignment tables and parallel segmentation can be extended to some extent, for instance by using colours and visualising deletions and additions. Regular variant graphs may include intradocumentary variation if the different paths through the texts are collated as separate witnesses[12]; only HyperCollate's variant graph output includes both intra- and interdocumentary variation. Structural variation, is currently only taken into account by HyperCollate and consequently only visualised in HyperCollate's variant graph. While the added value of studying this type of variation may be clear, it remains a challenge to visualise both linguistic/semantic and structural variation in an informative and clear manner. Fig. 11 may clearly convey the structural difference between witness 1 and witness 2 (i.e., the <head> element), but the raw collation output contains much more information which, if included, would probably overburden the user. A promising feature of visualisations intended to further explorations of textual variation is interactivity. One can imagine, for instance, the added value of discovering promising sites of revision through a graph representation, zooming in, and annotating the relationships between the witness nodes.

Acknowledging the various strengths and shortcomings of existing visualisations, we propose that there is not one, all-encompassing visualisation that pays head to all properties

[12] This practice leads to some problematic issues in case of complex revisions, see De Bruijn et al. 2007; Bleeker 2017, 111–114.

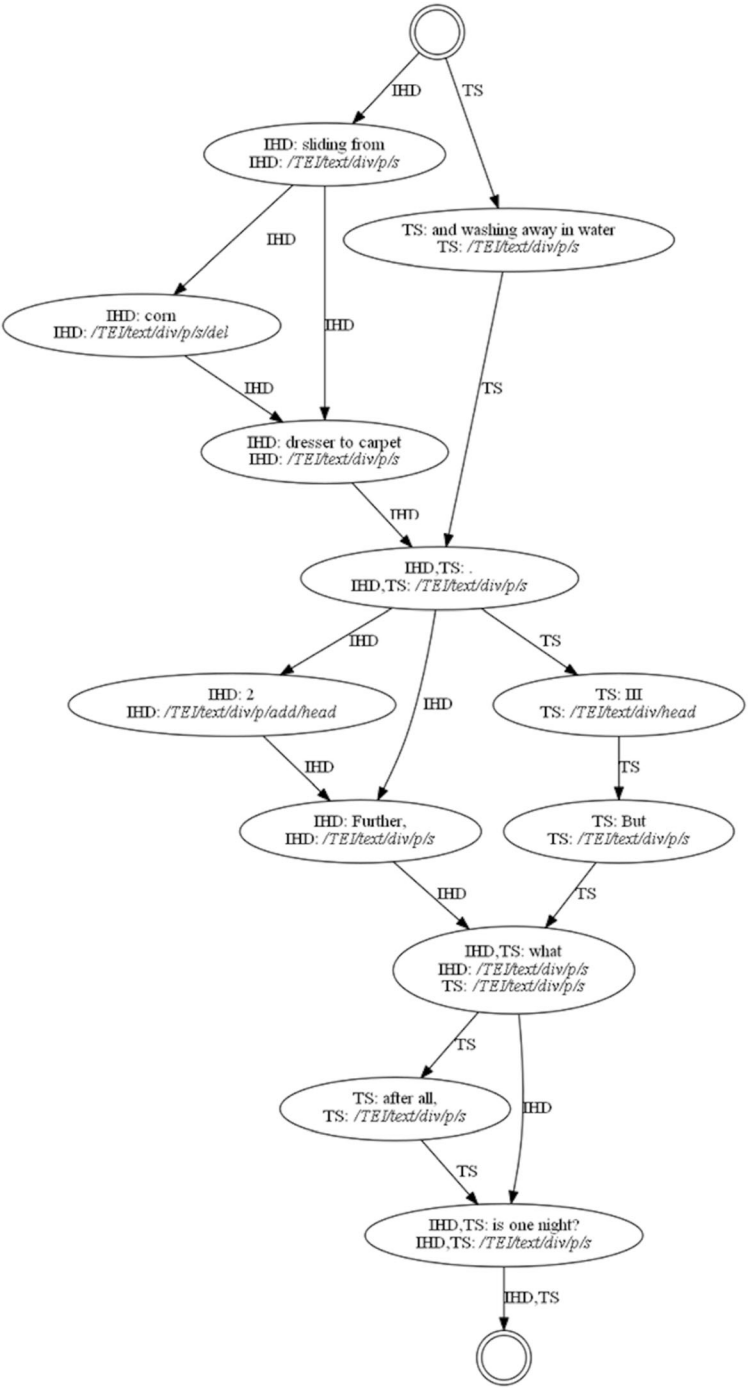


Fig. 12 Alternative visualisation of HyperCollate output, with each node containing the Xpath-like information about the place of the text in the XML tree (e.g. the path */TEI/text/div/p/s/* indicates that the ancestors of a text node are, bottom up, an <s> element, a <p> element, a element, the <text> element and the <TEI> element)

of text. Instead, each visualisation highlights a different aspect of textual variance or provides another perspective on text. Each perspective puts another textual characteristic before the footlights, while (ideally) making users aware of the fact that there is much more happing behind the familiar scenes. As Tanya Clement argues, focusing on one aspect can be instrumental in our understanding of text, helping the user 'get a better look at a small part of the text to learn something about the workings of the whole' (Clement 2013, §3). Indeed it seems that multiple and interactive representations (cf. Andrews and Van Zundert 2013; Jänicke et al. 2014; Sinclair et al. 2013) are a promising direction.

7 Visual literacy and code criticism

The process of visualising data is a scholarly activity in line with the process of modelling, hence the resulting visualisation influences the ways in which a text can be studied Collation output can be visualised in different ways, which raises essential questions regarding the assessment and evaluation of visualisations. The function of a digital visualisation is two-fold: on the one hand, it serves as a means of communication and on the other hand it provides an instrument of research. The communicative aspect implies that visualisation is first and foremost an affair of the scholar(s) who creating visualisations. The diversity of visualisations, each of which highlights different aspects of the text, reflects the hermeneutic aspect inherent to humanist textual research. Thus, by using visualisation to foreground textual variation, editors are able to better represent the multifocal nature of text. In order to choose an appropriate representation of collation output, then, scholars need to know what argument they want to make about their data set, and how the visualisation can support that argument by presenting and omitting certain information. Accordingly, they can estimate the value of a visualisation for a specific scholarly task and expose the inevitable bias embedded in technology.

When a visualisation is used as an instrument of study and exploration, it is vital to be critical about its workings and its (implicit) bias. This includes an awareness of which elements the visualisation highlights and, just as important, which elements are ignored. As Martyn Jessop has pointed out, humanist education often overlooks training in 'visual literacy', which can be defined as the effective use of images to explore and communicate ideas (Jessop 2008, 282). Visual literacy, then, denotes an understanding of the fact that a visualisation represents a scholarly argument. Jessop identifies four principles that facilitate the understanding of a visualisation: aims and methods, sources, transparency requirements, and documentation (Jessop 2008 290). The documentation of a visualisation of collation output then, could describe what research objective(s) it aims to achieve, on what witnesses it is based, and how these witnesses have been transcribed, tokenized, and aligned.[13] Another suitable rationale for critically evaluating the visualisation process is offered by the domains of 'tool criticism' or 'code criticism' (Traub and van Ossenbruggen 2015; Van Zundert and Dekker 2017, 125). Tool criticism assumes that the code base of scholarly tools reflects certain scholarly decisions and assumptions, and it raises critical questions in order to further awareness of the

[13] Although the value of documenting a tool's operations is uncontested, making use of documentation is not yet part of digital humanities' best practice. In that respect, it is worthwhile to keep in mind the RTFM-mantra of software development ('Read the F-ing Manual').

relationships between code and scholarly intentions. Questions include (but are not limited to) 'is documentation on the precision, recall, biases and pitfalls of the tool available', or 'is provenance data available on the way the tool manipulates the data set?' (Traub and van Ossenbruggen 2015).

Indeed, when it comes to evaluating the visualisation of automated collation results, one may well ask to what extent these witnesses and the ways in which they have been processed by the collation tool are subject to bias and interpretation. Like transcription (and any operation on text for that matter), collation is not a neutral process: it is subject to the influence of the editor. This becomes clear if we look at the different steps in the collation workflow as identified by the Gothenburg model (GM; 2009). The GM consists of five steps: tokenisation, normalisation, alignment, analysis, and visualisation. For each step, the editor is required to make decisions, e.g. 'what constitutes a token', 'do I normalise the tokens and, if so, do I present the original and the normalised tokens', or 'what is my definition of a match and how do I want to align the tokens?' As Joris Van Zundert and Ronald Haentjens Dekker emphasise, not all decisions made by collation software are easily accessible to the user, simply because they are the result of 'incredibly complex heuristics and algorithms' (Van Zundert and Dekker 2017, 123). To illustrate this, we can look at the decision tree used by HyperCollate to calculate the alignment of two simple sentences.

Witness A: `The dog's big eyes.`

Witness 2: `The dog's brown eyes.`

The graph in Figs. 13 and 14 are complementary and show all possible decisions the alignment algorithm of Hypercollate can take in order to align the tokens of witness A and witness B and the likely outcomes of each decision. An evident downside of such trees is that they become very large very quickly. For that reason, we see them as primarily useful for editors keen to find out more about the alignment of their complex text.

The GM pipeline is not strictly chronological or linear. Although automated collation does start with tokenization, not every user insists on normalising the tokens, and a step can be revisited if the outcome is considered unsatisfactory or not in line with the user's expectations. Though visualisation comes last in the GM model, this article has argued that it is surely not an afterthought to collation. In fact, the visual representation of textual variance entails an additional form of information modelling: editors are compelled to give physical form to an abstract idea of textual variation which exists at that point only in the transcription and (partly) in the collation result. Using the markup to obtain a more optimal alignment, as HyperCollate does, only emphasises this point: marking up texts

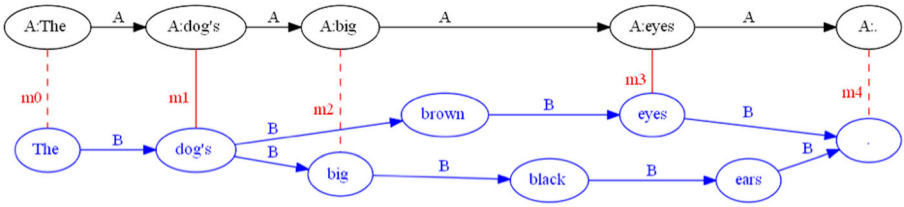


Fig. 13 The collation of witness B against witness A, with potential matches indicated in red

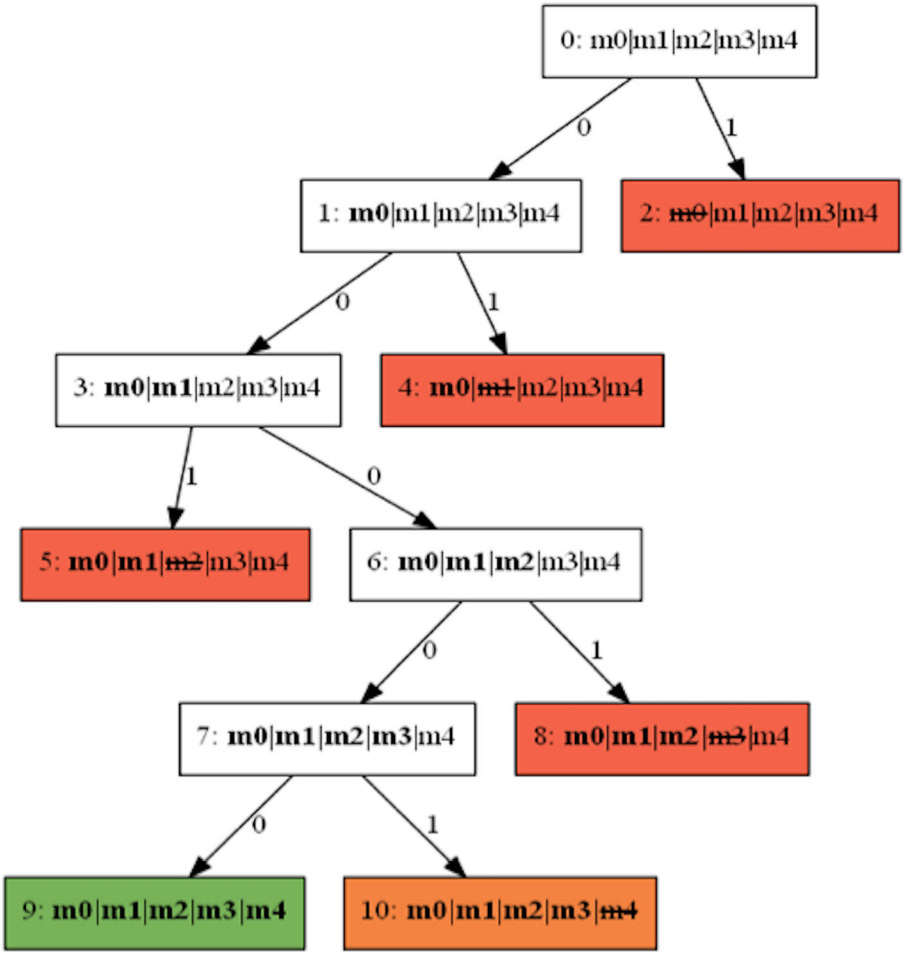


Fig. 14 The decision tree for collating witness B against witness A. Chosen matches indicated in bold, discarded matches rendered as strike-through; others are potential matches. Arrow numbers indicate the number of matches discarded since the root node (this number should be as low as possible). Red leaf nodes indicate a dead end, orange leaf nodes a 'sub-optimal' match, and green leaf nodes indicate an optimal set of matches

entails making explicit the knowledge and assumptions that would otherwise have been left implicit. Visualising the markup elements, then, implies that these assumptions and thus a particular scholarly orientation to text is foregrounded.

8 Conclusions

The present article investigated several methods of representing textual variation: alignment tables, synoptic viewers, and graphs. Two small textual fragments containing in-text variation and structural variation formed the example input for the alignment table and the variant graph visualisation. The fragments were transcribed in TEI/XML and subsequently collated with CollateX and

HyperCollate respectively. In addition, we looked at existing visualisations of the Versioning Machine and the Diachronic Slider. These visualisations were judged on their potential to represent different types of variance in addition to the regular interdocumentary variation: intradocumentary, linguistic, and structural. Visualising these aspects of text paves the way for a deeper, more thorough, and more inclusive study of the text's dimensions. We concluded that there is currently no ideal visualisation, and that the focus should not be on creating an ideal visualisation. Instead, we propose appreciating the multitude of possible visualisations which, individually, amplify a different textual property. This requires us to appreciate what a visualisation can do for our research goals and, furthermore, to evaluate its effectiveness. To this end, methods from code criticism and visual literacy can be of aid in furthering an understanding of the digital representations of collation output as rhetorical devices. We propose evaluating the usefulness of a visualisation on the basis of the following principles:

1) **Interactivity**. This may range from annotating the edges of a graph, adjusting the alignment by (re)moving nodes, to alternating between macro- and micro level explorations of variance.
2) **Readability and scalability**. Especially in a case of many and/or long witnesses, alignment tables and variant graphs become too intricate to read: their function becomes primarily to indicate complex revision sites.
3) **Transparency of the textual model**. The visualisation not only represents textual variance, but simultaneously makes clear what scholarly model is intrinsic to the collation. It needs to be clear which scholarly perspective serves as a model for transcription and representation.
4) **Transparency of the code**. Visualisations represent the outcome of an internal collation process which is usually not available to the general user audience. A clear, step-by-step documentation of the algorithmic process helps users understand what scholarly assumptions are present in the code, what decisions have been made, what parameters have been used, and how these assumptions, decisions, and parameters may have influenced the outcome. Decision trees may be of additional use. This applies particularly to interactive visualisations: if it's possible to adjust parameters or filters, these adjustments need to be made explicit.

Digital visualisation is sometimes regarded as an afterthought in humanities research, or even considered with a certain degree of suspicion. Some consider it a mere technical undertaking, an irksome habit of some digital humanists who recently learned to work with a flashy tool. Yet if used correctly, these flashy tools may also function as instruments of study and research, which means they should be evaluated accordingly. Within the framework of visualising collation output, visual literacy is key. Having a critical understanding of the research potential of visualisations facilitates our research into textual variance. After all, these representational systems produce an object which we use for research purposes; we need to take seriously the ways in which they do this. In addition to communicating a scholarly argument, digital visualisations of collation output foreground textual variation. The collation tool HyperCollate facilitates the examination of a text from multiple perspectives (some unfamiliar, some inspiring, some contrasting, but all of them highlighting a particular element of interest). This

freedom of choice invites scholars to reappraise prevalent notions and continue exploring the dynamic nature of text in dialogue with other disciplines. Digital visualisations, then, give us a means to take variants out of the graveyard and into an environment in which they can be fully appreciated.

References

Andrews, T., & Mace, C. (2012). *Trees of Texts: Models and Methods for an Updated Theory of Medieval Text Stemmatology*. Paper presented at the digital humanities conference, 2012, July 16–20, University of Hamburg. Abstract available at http://www.dh2012.uni-hamburg.de/conference/programme/abstracts/trees-of-texts-models-and-methods-for-an-updated-theory-of-medieval-text-stemmatology.1.html. Accessed 23 Dec 2018.

Andrews, T., & Van Zundert, J. (2013). *An Interactive Interface for Text Variant Graph Models*. Paper presented at the Digital Humanities Conference, 2013, July 16–19, University of Lincoln, Nebraska. Abstract available at http://dh2013.unl.edu/abstracts/ab-379.html. Accessed 23 Dec 2018.

Bleeker, E. (2017). *Mapping invention in writing: Digital infrastructure and the role of the genetic editor*. Ph.D. Dissertation, University of Antwerp.

Bleeker, E., Buitendijk, B., Dekker, R. H., Neyt, V., & van Hulle D. (2017). The challenges of automated collation of manuscripts. In *Advanced in digital scholarly editing*, Leiden: Sidestone Press, pp. 241–249.

Bleeker, E., Buitendijk, B., Dekker, R. H., & Kulsdom, A. (2018). *Including XML Markup in the Automated Collation of Literary Texts*. Proceedings of the XML Prague conference 2018, February 9–11, pp. 77–95.

Burnard, L., Jannidis, F., Middell, G., Pierazzo, E., & Rehbein, M. (2010). An encoding model for genetic editions, accessible at http://www.tei-c.org/Activities/Council/Working/tcw19.html (last accessed 2018, March 30).

Clement, T. (2013). Text analysis, data mining, and visualizations in literary scholarship. In *Literary studies in the digital age: An evolving anthology*. https://doi.org/10.1632/lsda.2013.0.

De Bruijn, P. (2002). Dancing around the grave. A history of historical-critical editing in the Netherlands. In Plachta, B. & Van Vliet, H.T.M. (red.), *Perspectives of scholarly editing/perspektiven der textedition* (pp. 113–124). Berlin: Weidler Buchverlag.

Dillen, W. (2015). *Digital scholarly editing for the genetic orientation: The making of a genetic edition of Samuel Beckett's works*. Ph.D. thesis, University of Antwerp.

Drucker, J. (2012). Humanistic theory and digital scholarship. In M. Gold (Ed.), *Debates in the digital humanities* (pp. 85–96). Minneapolis: University of Minnesota Press.

Haentjens Dekker, R., & Birnbaum, D. J. (2017). It's more than just overlap: Text as graph. Presented at Balisage: The Markup Conference 2017, Washington, DC, August 1 - 4, 2017. In *Proceedings of Balisage: The Markup Conference 2017*. Balisage Series on Markup Technologies, vol. 19. https://doi.org/10.4242/BalisageVol19.Dekker01.

Jänicke, Stefan, Gessner, Annette, Büchler, Marco, & Scheuermann Gerik (2014). Design rules for visualizing text variant graphs. In *Proceedings of the digital humanities 2014,* edited by Clare Mills, Michael Pidd and Jessica Williams.

Joyce, J. (1984-1986). *Ulysses: A critical and synoptic edition, prepared by Hans Walter Gabler with Wolfhard Steppe and Claus Melchior*, 3 vols. New York & London: Garland Publishing Inc.

Jessop, M. (2008). Digital visualization as a scholarly activity. *Literary and Linguistic Computing, 23*(3), 281–293.

Roos, T., & Heikkilä, T. (2009). Evaluating methods for computer-assisted stemmatology using artificial benchmark data sets. *Literary and Linguistic Computing, 24*(4), 417–433.

Schacht, P. (2016). 'Introduction' in: Thoreau, Henry David. Walden: A fluid-text edition. Digital Thoreau. http://digitalthoreau.org/fluid-text-toc. Accessed 27 May 2019.

Schäuble, J., & Gabler, H. W. (2016). *Visualising processes of text composition and revision across document Borders*. Paper presented at the symposium Digital Scholarly Editions as Interfaces, Graz, Austria, September 22–23.

Schmidt, D., & Colomb, R. (2009). A data structure for representing multi-version texts online. *International Journal of Human-Computer Studies, 67*(6), 497–514.

Sinclair, S., Ruecker, S., & Radzikowska, M. (2013). Information visualization for humanities scholars. In *Literary studies in the digital age, an evolving anthology*, edited by Kenneth Price and Ray Siemens. Available at https://dlsanthology.mla.hcommons.org/information-visualization-for-humanities-scholars. Accessed 23 Dec 2018

Traub, M., & van Ossenbruggen, J. (2015). Workshop on tool criticism in the digital humanities. *CWI Techreport July 1, 2015*. Available at https://pdfs.semanticscholar.org/d337/ce558c2fd1d8be793786c9cfc3fab6512dea.pdf. Accessed 27 May 2019.

Vanhoutte, E. (1999). Where is the editor? *Human IT, 3.1*, 197–214.

Van Zundert, J., & Dekker, R. H. (2017). Code, scholarship, and criticism: When is code scholarship and when is it not? *Digital Scholarship in the Humanities, 32*, 121–133.