

Filología digital *versus* filología clásica en el estudio de la tradición manuscrita del Fuero Juzgo

CHARLES MABILLE

Université de Lausanne

charles.mabille@unil.ch

ORCID-ID: <https://orcid.org/0000-0002-6815-7608>

RESUMEN

El objetivo de este artículo es poner a prueba cuatro herramientas de filología digital (Juxta, CollateX, Stemmaweb y TRAViz) para valorar su utilidad concreta en la *collatio codicum* y en la *constitutio stemmatis* para el filólogo que quiera estudiar una tradición manuscrita con un número elevado de testimonios conservados. Las hemos utilizado para nuestro caso de estudio concreto, la tradición manuscrita del Fuero Juzgo (versión romance del Liber Iudiciorum), comparando los resultados con las familias de manuscritos que hemos identificado con métodos de filología clásica para entender hasta qué punto los dos métodos (filología clásica y filología digital) son complementarios y en qué medida el segundo permite mejorar el primero tanto en la calidad de los resultados como en el tiempo que el filólogo puede ahorrar utilizando herramientas informáticas.

PALABRAS CLAVE: Filología digital, crítica textual, Fuero Juzgo, CollateX, Juxta, Stemmaweb, TRAViz, *collatio codicum*, *constitutio stemmatis*, colación automática, estematología, árbol filogenético.

Digital philology *versus* classical philology in the study of the manuscript tradition of the Fuero Juzgo

ABSTRACT

The aim of this article is to test four digital philology tools (Juxta, CollateX, Stemmaweb and TRAViz) to assess their concrete usefulness in *collatio codicum* and *constitutio stemmatis* for the philologist wanting to study a manuscript tradition with a large number of preserved witnesses. We have used them for our specific case study, the manuscript tradition of the Fuero Juzgo (Romance version of the Liber Iudiciorum), comparing the results with the families of manuscripts that we have identified using classical philology methods in order to understand to what extent the two methods (classical philology and digital philology) are complementary and to what extent the latter can help the former, both in terms of the quality of the results and in terms of the time that the philologist can save by using computer-based methods.

KEYWORDS: Digital philology, Textual criticism, Fuero Juzgo, CollateX, Juxta, Stemmaweb, TRAViz, *Collatio codicum*, *Constitutio stemmatis*, Automated collation, Stemmatology, Phylogenetic tree.

1. INTRODUCCIÓN

Con el fin de identificar los lugares críticos –los lugares que concentran más variación– que le permitirán clasificar los manuscritos en diferentes familias (*examinatio* y *selectio*), la primera etapa esencial a la que el investigador que tenga la intención de estudiar la tradición manuscrita de un texto medieval –ya sea literario o no– debe enfrentarse, una vez identificados los manuscritos que se conservan (*recensio*), es la *collatio codicum*, es decir, la comparación de la totalidad de los testimonios. Si el número de códices

conservados es elevado, esta tarea puede resultar ardua y compleja porque el investigador debe tener a la vista todos los testimonios simultáneamente para identificar y analizar los lugares en los que el texto presenta variación. De hecho, el esfuerzo cognitivo que requiere la colación de los manuscritos, tanto por la necesidad de acordarse de las lecciones de los diferentes testimonios como por la de concentrarse para no perder de vista el pasaje que se está analizando, es muy elevado. Además, comparar los testimonios de un texto con un manuscrito base y anotar las variantes de cada uno de ellos es una tarea que tiende a realizarse de manera mecánica y en la que una mínima falta de atención puede provocar errores.

Un método que reduzca lo más posible este esfuerzo cognitivo gracias a herramientas digitales constituye una mejora considerable y, de hecho, desde mediados del siglo XX, ha habido varios intentos de facilitar esta tarea. Pero ¿qué puede aportar realmente la tecnología para agilizar el trabajo del filólogo? ¿Hasta qué punto podemos confiar en estas herramientas? Estas son las preguntas a las que intentaremos dar respuesta en este trabajo, poniendo a prueba diferentes programas informáticos para explorar la tradición manuscrita del Fuero Juzgo, aún no estudiada en su conjunto y cuyas relaciones desconocemos hasta hoy¹. Nuestro objetivo principal es contribuir al descubrimiento de relaciones genéticas entre los testimonios conservados y, a la vez, proponer una valoración crítica de estos programas para determinar en qué medida pueden servir al filólogo que estudie una tradición manuscrita medieval. No pretendemos llegar a conclusiones definitivas sobre el primer punto, puesto que el filólogo debe estudiar una obra en su totalidad para ser capaz de definir con seguridad las filiaciones entre los testimonios (Blecuca 1983: 45) y, por el momento, nuestra *collatio* se limita a una ley (8.2.1). Se trata, más bien, de un estudio programático para investigaciones futuras y un primer acercamiento al tema para intentar despejar algunas de las incógnitas existentes y esbozar algunas pistas que, más adelante, pondremos en práctica con más profundidad.

Puesto que el objetivo principal de este experimento es comparar los métodos de filología digital y los de filología clásica y en qué medida los primeros pueden contribuir a mejorar los segundos, cabe precisar que nos centraremos sobre todo en el aspecto lingüístico y en las variantes presentes en los diferentes manuscritos para entrever las relaciones entre los testimonios, porque este es el ámbito principal en que las herramientas que ponemos a prueba pueden ayudarnos. Dejaremos de lado, por el momento, las cuestiones dialectales y codicológicas de los manuscritos, que presentan gran interés pero que abordaremos en otra etapa de esta investigación, pues somos conscientes de que la crítica textual debe apoyarse en argumentos de naturaleza variada para llegar a conclusiones empíricamente válidas. Tampoco presentaremos en este artículo resultados exhaustivos de nuestra *collatio*, entrando en el análisis detallado de las familias de manuscritos que hemos identificado, si bien el lector interesado podrá encontrar más informaciones en otros trabajos².

En este artículo, después de recorrer las principales etapas de la historia de la *collatio* asistida por ordenador y describir el actual estado de la cuestión (§2.1-2.2), presentaremos las herramientas de filología digital que hemos puesto a prueba

¹ El Fuero Juzgo es la versión romance del Liber Iudiciorum, código de leyes visigóticas traducido al romance a mediados del siglo XIII por mandato de Fernando III. Para más informaciones acerca de este texto y de su importancia histórica, *vid.* Pérez Martín 2002 y Castillo Lluch 2011, 2012 y 2016.

² *Cfr.* Mabilille 2022 (59-152) y Castillo Lluch y Mabilille 2023 (en prensa).

valorando su utilidad para nuestro caso de estudio concreto, la tradición manuscrita del Fuero Juzgo (§2.3-2.4). A continuación, comentaremos una parte de los resultados a los que hemos llegado con métodos de filología clásica (§2.5) y, finalmente, compararemos los resultados de los dos métodos para tratar de entender en qué medida pueden resultar complementarios (§3)³.

2. EXPERIMENTO: FILOLOGÍA DIGITAL *VERSUS* FILOLOGÍA CLÁSICA

2.1. Colación asistida por ordenador: panorama histórico

El primer intento de facilitar la tarea de la *collatio codicum* se debe a un investigador inglés, Charlton Hinman, que, en la década de 1940, desarrolló una técnica de colación óptica. El principio era sencillo, pero ingenioso: dos retroproyectores proyectaban en la misma pantalla el contenido de dos testimonios del mismo texto, con letras del mismo tamaño, de modo que las diferencias entre los dos testimonios pudieran identificarse fácilmente a simple vista (Nury y Spadini 2020: 3). Sin embargo, como precisan Elisa Nury y Elena Spadini (2020: 3), incluso si el proceso de colación óptica se realiza con un dispositivo electrónico, el investigador alinea los textos manualmente. En otras palabras, aún no se podía hablar de *collatio* automática.

El método de la colación óptica fue utilizado hasta los años sesenta, cuando se desarrollaron los primeros algoritmos de colación. Estos comparaban las cadenas de caracteres del manuscrito base con las de los demás testimonios, uno tras otro. Cuando encontraba una diferencia, el algoritmo debía localizar el punto en que los dos textos volvían a coincidir para identificar el tamaño de la variación antes de repetir la misma operación, y así sucesivamente hasta haber recorrido todo el texto (Nury y Spadini 2020: 3). Hasta aquí, el ordenador realiza la misma operación que el filólogo cuando coteja los manuscritos, en otras palabras, se limita a comparar dos testimonios a la vez —el manuscrito base y otro testimonio— recorriendo los dos textos en paralelo hasta encontrar una diferencia. Una vez acabado el cotejo, vuelve a realizar la misma operación cotejando otro testimonio con el manuscrito base.

A partir de la década de los noventa, la gran novedad del método consistió en poder comparar una cantidad teóricamente infinita de manuscritos en muy poco tiempo. Los investigadores empezaron a utilizar algoritmos que realizaban alineamientos por pares (*pairwise alignment*), es decir: «they compare two texts at a time and, in case more than two witnesses are to be aligned, merge the results of the pairwise comparisons to obtain the global result, that is the alignment between all the witnesses» (Nury y Spadini 2020: 4). Como precisan Nury y Spadini (2020: 4), aunque esta innovación implica un cambio metodológico considerable, el resultado de la colación sigue dependiendo del orden en el que el ordenador realiza el alineamiento y se sirve de un manuscrito base. En realidad, no es la mejor manera de proceder si lo que interesa es comprender la relación entre los testimonios, ya que el manuscrito base se elige en la casi totalidad de los casos de manera

³ Una parte de este artículo, en particular los §§ 2.1 a 2.4, proviene de mi trabajo de fin de máster titulado *La tecnología al servicio de la collatio codicum: el caso del Fuero Juzgo*, realizado en la Facultad de Letras de la Universidad de Lausana bajo la dirección de Mónica Castillo Lluch, citado en este trabajo como Mabilille 2022. La tesina completa puede encontrarse en el servidor académico de la Universidad de Lausana: https://serval.unil.ch/fr/notice/serval:BIB_S_33415

—al menos parcialmente— arbitraria (Spadini 2017: 345-346). Este tipo de algoritmo todavía se utiliza en algunos de los programas más recientes porque, a pesar de ofrecer resultados perfectibles, tiene un bajo coste computacional y permite efectuar colaciones con un número muy alto de testimonios en una duración aceptable.

Existe también otro tipo de algoritmo (*progressive multiple alignment*), desarrollado en los años ochenta en el ámbito de la bioinformática para alinear secuencias de ADN o de proteínas (Nury 2018: 54) y propuesto para la crítica textual por Matthew Spencer y Christopher Howe (2004). El alineamiento progresivo múltiple realiza un alineamiento por pares entre todos los testimonios, construye un *unrooted stemma* gracias al algoritmo Neighbor-Joining a partir de los resultados del alineamiento por pares y, finalmente, vuelve a realizar un alineamiento por pares analizando los manuscritos en un orden preciso, empezando por los testimonios que el algoritmo ha identificado como los más parecidos (Spencer y Howe 2004: 257-261; Spadini 2017: 347). Como explican Ronald H. Dekker y Gregor Middell (2011: 6), la gran ventaja de este método es que, cada vez que se añade un nuevo manuscrito a la colación, se compara con los resultados acumulados hasta el momento y no con un manuscrito base. El algoritmo acumula el resultado de cada nuevo alineamiento en un esquema —llamado *grafo de variantes* (cfr. Figura 4)— en el que se indica cada vez que el texto de un testimonio diverge de los demás. Este método «reflects the idea of dynamic programming, which takes a complicated problem and breaks it down into a repetition of the relatively easier task of two-witness alignment» (Bleeker 2017: 96). Además, permite al filólogo proceder de manera muy distinta a cuando realiza el cotejo manualmente, puesto que, mientras el *pairwise alignment* plasma el cotejo manual, el *progressive multiple alignment* realiza operaciones impensables para un cerebro humano. La desventaja de un algoritmo de alineamiento múltiple es que es mucho más costoso computacionalmente que un alineamiento por pares (Spadini 2017: 347). Según Spadini (2017: 347-348), el único programa que utiliza este tipo de algoritmo es CollateX⁴, desarrollado por Ronald H. Dekker y Gregor Middell (2011), que es el que más utilizaremos en este trabajo. Los algoritmos que realizan alineamiento progresivo múltiple, a pesar de que el resultado de la colación sigue dependiendo del orden en que se comparan los manuscritos, representan el estado actual de la cuestión (Birnbaum y Spadini 2020: 12). Como subrayan David J. Birnbaum y Elena Spadini (2020: 12), «optimal order-independent multiple-witness alignment is currently an unsolved problem in computer science», porque comparar todos los manuscritos simultáneamente es muy complejo y la duración del cotejo aumentaría exponencialmente.

La *collatio* asistida por ordenador representa sin duda un avance considerable para la investigación filológica. Helena Bermúdez Sabel (2016: 22) insiste sobre la doble ventaja de los métodos computacionales aplicados a la filología: «la informatización de este proceso [...] no solo incrementa exponencialmente la eficiencia en lo que respecta a la rapidez en la obtención de los resultados, sino que también nos proporciona la seguridad de que todas las variantes sean detectadas». Hemos visto que, actualmente, se utilizan dos tipos de algoritmos, uno que compara todos los testimonios a partir de un manuscrito base y otro que compara los testimonios entre sí. La colación sin texto base, según Bermúdez Sabel (2016: 23), constituye «la estrategia de colación más rigurosa cuando se desconoce, aunque sea parcialmente, la historia material de la tradición

⁴ Parece que TRAViz (Jänicke 2015) utiliza también un algoritmo de alineamiento múltiple. Tal vez cuando Spadini publicó el artículo este programa aún no era conocido.

textual objeto de estudio». Este es el caso de la tradición manuscrita del Fuero Juzgo, y realizar una colación sin texto base no se puede alcanzar sin recurrir a algoritmos específicos.

2.2. El modelo de Gotemburgo

En 2009, en el marco del programa de investigación *Interedition*, los desarrolladores de Juxta y CollateX –dos programas que expondremos a continuación– se reunieron en Bruselas y en Gotemburgo en un taller sobre programas de colación asistida por ordenador con el objetivo de colaborar entre ellos para poner en común sus ideas y optimizar el desarrollo de herramientas que pudieran responder a las necesidades de cada investigador (Bermúdez Sabel 2016: 23). El resultado de este taller fue el modelo de Gotemburgo (*Gothenburg model*), que describe las operaciones que un programa de colación automática debe realizar. El proceso de colación está dividido en varias etapas para que el usuario pueda intervenir en cada una de ellas y adaptar los resultados a sus necesidades. Como describe Ronald H. Dekker (2014: 3), son cuatro las etapas indispensables: (1) la tokenización de los textos objeto de estudio, (2) el alineamiento de los *tokens*, (3) el análisis del alineamiento y (4) la visualización de los resultados. A estas etapas obligatorias, se puede añadir la transcripción de los textos y la normalización de los *tokens*. Dado que cada etapa se efectúa con un algoritmo diferente, podemos considerar que un programa de colación es una combinación de algoritmos complementarios que trabajan juntos para llegar al resultado (Bleeker 2017: 95).

Aunque los programas de colación pueden ser muy atractivos, no debemos olvidar que implican una etapa previa que no es necesaria durante una colación manual: la **transcripción** de todos los testimonios. Como precisa Helena Bermúdez Sabel (2016: 25), al transcribir, el investigador que estudia una tradición textual compleja puede tener la tentación de neutralizar algunas variantes que considera no significativas, como las (orto)gráficas y, en ciertos casos, las morfosintácticas, para que los programas utilizados identifiquen únicamente las variantes que el mismo investigador intuye como genealógicas, es decir, en la mayoría de los casos, las variantes que Inés Fernández-Ordoñez (2019: 381) define como «textuales». Sin embargo, normalizar los textos durante la transcripción comporta una desventaja importante: todas las variantes neutralizadas se pierden y no se pueden recuperar en otra etapa de la colación o incluso en etapas posteriores al proceso de colación, como en la *constitutio textus* (Birnbaum y Spadini 2020: 4-5). De hecho, en el ecosistema digital, el investigador puede hacer la colación a partir de las variantes significativas sin necesidad de neutralizar las variantes no significativas⁵. Además, como Tara L. Andrews (2016: 538) ha podido observar en sus investigaciones, las variantes que tradicionalmente parecen carecer de valor estematológico tienden a seguir el *stemma*; de ahí que normalizar demasiado los textos durante la transcripción pueda representar una pérdida de información potencialmente significativa. Las variantes de lengua –según la terminología adoptada por Inés Fernández-Ordoñez–, especialmente las variantes gráficas, no pueden, por supuesto, utilizarse para establecer filiaciones, pero sí pueden confirmarlas si tenemos otros indicios probablemente monogénéticos o muchas variantes no significativas que van en la misma dirección, «since they are likely to follow the same genealogical patterns of a

⁵ Agradecemos a Helena Bermúdez Sabel esta preciosa observación.

textual tradition» (Nury 2018: 115). A pesar de esto, en nuestro caso hemos tenido que normalizar algunos aspectos de nuestras transcripciones para optimizar los resultados del alineamiento automático. Se trata de la separación de palabras (normalizadas según el uso actual), la puntuación y la distinción entre minúsculas y mayúsculas (eliminadas completamente). De hecho, estos tres aspectos, que no tienen ninguna relevancia estematológica, no nos interesan para estudiar las filiaciones entre los manuscritos, y normalizarlos nos ha permitido realizar el alineamiento evitando ruido, esto es, sin que el algoritmo identifique variación irrelevante.

La primera etapa del proceso de colación automática propiamente dicho es la **tokenización**, esto es, «el proceso mediante el cual [...] se divide la cadena de caracteres en unidades (=tokens) que serán posteriormente alineadas» (Bermúdez Sabel 2016: 24). La unidad que se elige es generalmente la palabra, de modo que el programa crea un nuevo *token* cada vez que identifica un espacio en blanco (Bermúdez Sabel 2016: 24; Birnbaum y Spadini 2020: 4)⁶. Cuando se estudian tradiciones medievales con mucha variación gráfica y ortográfica, es preferible tokenizar los textos antes de utilizarlos en el programa de colación para poder añadir información suplementaria a los *tokens*, como su forma normalizada o datos topográficos (el folio y la línea en la que se encuentra, por ejemplo). Las informaciones adicionales (salvo las formas normalizadas) se ignoran durante todo el proceso de colación, pero siguen disponibles en los resultados (Nury 2018: 101-102 y 152-153). Sin embargo, el procedimiento para intervenir sobre la tokenización no es intuitivo. El investigador debe crear un documento JSON (*JavaScript Object Notation*) con el texto de cada testimonio previamente tokenizado y utilizarlo como *input* en CollateX (Nury 2018: 221). Por tanto, aunque es posible intervenir manualmente en esta etapa, el usuario debe tener conocimientos más o menos avanzados en programación.

La etapa de la **normalización**, que no es necesaria para la colación, puede resultar muy útil para mejorar el alineamiento si se estudia una tradición con mucha variación ortográfica (Birnbaum y Spadini 2020: 1). El investigador puede realizar transcripciones paleográficas y, después de la tokenización, neutralizar las variantes no significativas atribuyendo un valor *n* (forma normalizada) a los *tokens* para que el programa realice el alineamiento con estas formas, y no con las formas paleográficas (Nury 2018: 119). De hecho, según Helena Bermúdez Sabel (2016: 25), «para el análisis de una tradición profundamente intrincada y con múltiples testimonios, es posible que las variantes no significativas entorpezcan el análisis». Propone entonces neutralizar los alógrafos, normalizar las mayúsculas y las minúsculas, las abreviaturas y las variantes ortográficas antes de efectuar la colación. Ya se comentó anteriormente que normalizar variantes en principio no significativas durante la transcripción podía implicar una pérdida de información, por lo que resulta oportuno llevar a cabo la normalización en esta etapa y no antes, de manera que la forma normalizada se genere como una copia de la forma original y no la sustituya por completo (Birnbaum y Spadini 2020: 5). En otras palabras, el algoritmo utilizará la forma normalizada para efectuar el alineamiento, pero la forma que aparecerá en la visualización de los resultados es la forma paleográfica. Efectivamente, Elisa Nury (2018: 81) precisa que «the advantage of tokens having both an original and a

⁶ Esto justifica nuestra elección de normalizar la separación de palabras según el uso actual. Si no hubiéramos normalizado este aspecto, los *tokens* no hubieran sido identificados correctamente y, por tanto, los resultados del alineamiento hubieran sido menos precisos.

normalised form is that it is possible to hide insignificant variants without a loss of information: the original form is always available, should it be needed».

Durante el **alineamiento**, el algoritmo alinea los *tokens* de cada testimonio, introduciendo espacios blancos (=gap tokens) cuando uno de los testimonios presenta una omisión para que los textos aparezcan perfectamente alineados (Bermúdez Sabel 2016: 24). Para los *tokens* que encuentran una correspondencia exacta en los demás testimonios, el problema se resuelve directamente. En cambio, cuando la palabra correspondiente cambia –por ejemplo, si presenta una ortografía distinta no normalizada o diferencias morfosintácticas– puede ocurrir que el algoritmo cometa errores. Estos pueden evitarse buscando, después de haber identificado todas las correspondencias exactas, *tokens* que se parecen mucho y entre los cuales cambian solo unos caracteres (método del *fuzzy matching* o *near matching*) (Birnbaum y Spadini 2020: 12-13). Otro problema puede ser el de las palabras que aparecen varias veces en el texto y que pueden falsear los resultados (por ejemplo, los artículos), pues el programa puede identificar *exact matches* con *tokens* que no están en la misma posición y considerarlos, por consiguiente, como transposiciones. Para resolver este problema, es posible atribuir a estas palabras un *token* ‘cero’, de manera que el algoritmo identifique las correspondencias entre los *tokens* de alrededor, y estas palabras, los *tokens* ‘cero’, acaben alineadas por *forced match* (Birnbaum y Spadini 2020: 11).

Encontrar el alineamiento perfecto puede resultar computacionalmente muy complejo. Por esta razón, el algoritmo «is based on a heuristic method which searches for an acceptable alignment, even if it is not the best alignment» (Nury 2018: 85 y 268). De hecho, los algoritmos basados en el método heurístico pueden resolver problemas muy complejos en muy poco tiempo, pero con resultados menos precisos (Nury 2018: 83-84). A partir de este alineamiento imperfecto, pero satisfactorio, el investigador puede modificar los resultados para corregir los errores cometidos por el algoritmo (**análisis**). Eso se puede hacer de diferentes maneras. En primer lugar, existen interfaces interactivas en las que se puede corregir el alineamiento, por ejemplo, Stemmaweb, TRAViz o PyCoviz⁷. En segundo lugar, si el investigador no tiene muchos conocimientos en programación, puede optar por corregir manualmente los documentos CSV (*comma separated values*) generados con CollateX directamente en un editor de hojas de cálculo como Microsoft Excel⁸. El problema que supone el hecho de intervenir manualmente en los resultados es que, desafortunadamente, todas las correcciones que se efectúan después del alineamiento se pierden si el investigador quiere volver a realizar la colación (Birnbaum y Spadini 2020: 12).

La última etapa del modelo de Gotemburgo es la **visualización** de los resultados obtenidos durante todo el proceso de la colación. Existen *outputs* diferentes para visualizar los resultados y cada programa tiene sus particularidades, que presentaremos más adelante.

⁷ Comentamos estas interfaces más adelante.

⁸ A pesar de sus inconvenientes, hemos corregido los resultados del alineamiento de esta manera, pues se trata del método más intuitivo y requiere menos conocimientos en programación.

2.3. Presentación de los programas utilizados

En este apartado, describiremos brevemente los programas utilizados centrándonos en la accesibilidad para un investigador que no esté acostumbrado a trabajar con herramientas informáticas, en los tipos de resultados que pueden generar y en su utilidad para nuestro caso de estudio. Cabe precisar que, antes de empezar este experimento, no teníamos prácticamente ninguna formación en programación. Por esta razón, desde nuestro punto de vista, el factor más importante a la hora de elegir un programa será el esfuerzo que su utilización exija a un principiante en relación con la utilidad de los datos que le pueda proporcionar.

2.3.1. Juxta Desktop

Juxta Desktop es un programa desarrollado en 2005 por el grupo de investigación *Applied Research in Patacriticism* (ARP) de la Universidad de Virginia bajo la dirección de Jerome McGann (Nury 2018: 63 y 311). En 2012, desarrollaron una nueva plataforma en línea, Juxta Commons, que, según Elisa Nury (2018 224-227), ofrecía más posibilidades al usuario. Por ejemplo, se podían utilizar más tipos de *input* y generar diferentes tipos de visualizaciones. Sin embargo, como hoy en día la plataforma ya no es accesible, nuestro comentario se limitará a la versión Desktop de Juxta.

La gran ventaja de Juxta es su interfaz muy intuitiva y fácil de usar. El usuario, al crear una nueva comparación, puede añadir documentos en texto simple (.txt) y el programa adapta los resultados cada vez que se añade o se quita un documento. La colación se efectúa con un manuscrito base, que se puede cambiar en cualquier momento, y los resultados se adaptan al nuevo manuscrito base en un par de segundos. Juxta ofrece algunas visualizaciones muy diferentes de las de los demás programas que hemos utilizado, pero estas soluciones sirven sobre todo para tener una visión global de la variación textual y saber qué manuscritos son más o menos diferentes del manuscrito base que se ha elegido. Nos referimos al *heatmap* (mapa de calor) y al histograma. En el primer caso, el programa propone visualizar el texto del manuscrito base con todas las lecciones que presentan variantes en otros testimonios (Bermúdez Sabel 2016: 30). Los lugares del texto que cambian en los demás testimonios aparecen subrayados en azul y cuanto más oscuro es, más testimonios presentan una solución diferente al manuscrito base (Nury y Spadini 2020: 9).

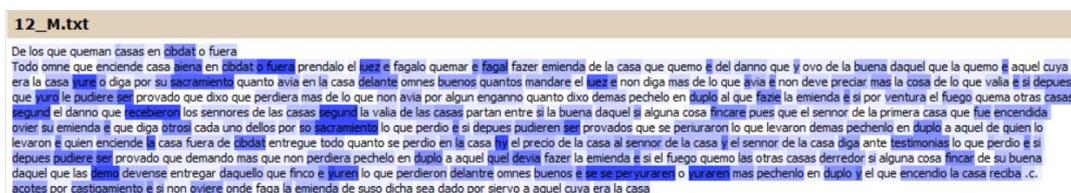


Figura 1. Ejemplo de mapa de calor en Juxta Desktop con el ms. de Murcia (M) como ms. base, comparado con todos los mss. conocidos del Fuero Juzgo.

Este tipo de visualización permite localizar las lecciones más propensas a la variación dentro de una tradición textual. Sin embargo, los resultados no son muy satisfactorios para estudiar una tradición medieval, puesto que Juxta no permite al usuario normalizar los diferentes *tokens*. Por lo tanto, aparecen tanto variaciones

significativas (variantes sustantivas o morfosintácticas, por ejemplo) como variaciones que no interesan al filólogo a la hora de establecer filiaciones entre los manuscritos (variantes gráficas y ortográficas). Concordamos con Elisa Nury (2018: 80-81) cuando afirma que «it is already difficult with only three witnesses to locate significant variants [...] where the text is not normalised. If there were many more witnesses collated, the text would become entirely blue, and the visualisation would not be helpful at all». El mismo problema aparece con el histograma:

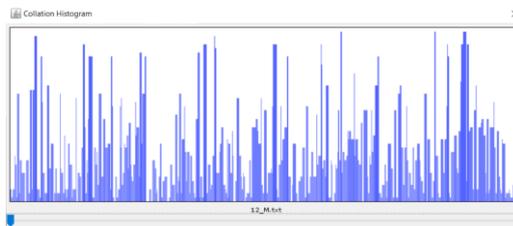


Figura 2. Ejemplo de histograma en Juxta Desktop con el ms. de Murcia (M) como ms. base, comparado con todos los mss. conocidos del Fuero Juzgo.

El histograma nos enseña, como el mapa de calor, los lugares del texto que presentan lecciones divergentes entre los diferentes testimonios y el manuscrito base. Este tipo de visualización puede resultar muy útil si el usuario está trabajando sobre un texto muy largo para identificar directamente los pasajes que varían más (Bemúdez Sabel 2016: 30). Sin embargo, como decíamos, comporta el mismo problema que el mapa de calor: el usuario no puede informar al programa de cuáles son las variantes significativas. La única solución parece ser la de normalizar los testimonios cuando se hace la transcripción, método que, como ya vimos, tiene muchas desventajas.

A pesar de los límites que tiene Juxta —la imposibilidad de normalizar los *tokens* y el hecho de utilizar un manuscrito base— el programa puede resultar muy útil a la hora de comparar dos testimonios para entender cuál de los dos ha copiado del otro, cuando el investigador ya sabe que estos pertenecen a una misma familia de la tradición textual. De hecho, la interfaz permite visualizar dos testimonios en paralelo subrayando todas las variantes.

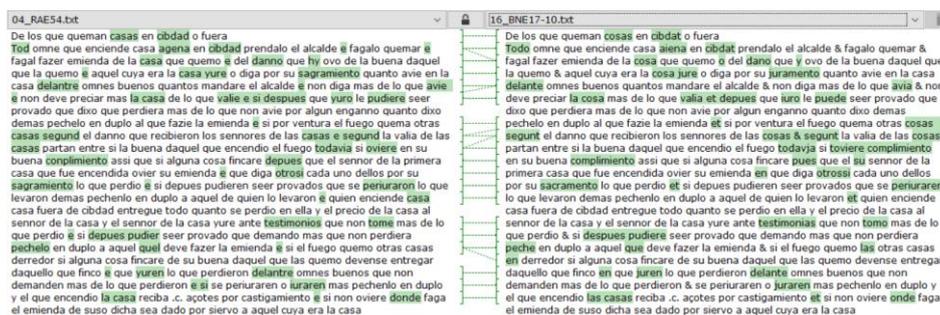


Figura 3. Comparación entre RAE54 y BNE17-10 con Juxta Desktop.

Por consiguiente, hemos utilizado Juxta Desktop solo después de haber identificado las diferentes familias dentro de la tradición textual del Fuero Juzgo para refinar los resultados y tratar de establecer filiaciones más precisas, cuando ya teníamos otros indicios que nos permitían saber con cierta seguridad que estaban relacionados y que uno copiaba, o parecía copiar, del otro.

2.3.2. CollateX

CollateX es, sin duda, el más complejo de los cuatro programas utilizados, pero también el que permite generar más tipos de resultados y modificarlos, a condición de poseer conocimientos informáticos relativamente avanzados. Hemos intentado entender su funcionamiento para ver hasta qué punto la inversión en tiempo y energía valía la pena para nuestro objeto de estudio y, en general, al filólogo que se enfrenta a una tradición manuscrita con muchos testimonios.

La ventaja principal de CollateX es que, por utilizar un *progressive multiple alignment*, la colación no depende de un manuscrito base (Spadini 2017: 346-348). El objetivo principal de CollateX es mejorar el alineamiento, centrándose en tres aspectos cruciales: (1) la detección de transposiciones; (2) la posibilidad de encontrar correspondencias entre *tokens* similares, pero con variación ortográfica; y (3) la independencia del orden en que los manuscritos son procesados (Dekker y Middell 2011: 2). Sin embargo, los resultados del alineamiento son perfectibles y parece necesario entender cómo interactuar con estos últimos para poder trabajar con resultados fiables y disfrutar plenamente de las posibilidades que ofrece este programa (Dekker *et al.* 2015: 456-457).

El problema principal que hace de CollateX un programa difícil de manejar es que no dispone de ninguna interfaz. La versión Java de CollateX –la más abaricable– funciona en línea de comandos, no solo para hacer las comparaciones, sino también para descargar el programa y las librerías necesarias para generar los resultados. Por lo tanto, el usuario debe tener al menos alguna noción de cómo funciona la línea de comandos de su computadora. A partir de un documento *input* en texto simple (.txt), la versión Java permite generar dos tipos de visualizaciones: una *alignment table* en formato CSV, esto es, una tabla de colación, y un grafo de variantes en formato DOT, que se puede visualizar en un documento PDF gracias a la librería GraphViz (Figura 4). Los resultados del alineamiento son satisfactorios, aunque puede ocurrir que el programa cometa errores. El problema de la versión Java es que el usuario no puede interactuar con los resultados del alineamiento, salvo si los vuelve a utilizar en otra interfaz que lo permita (por ejemplo, Stemmaweb). Además, aunque se pueda comparar un número teóricamente infinito de testimonios simultáneamente, el programa parece fallar cuando se trata de textos relativamente largos. En nuestro caso concreto, la versión Java no da resultados completos cuando utilizamos como *input* el texto de los cuarenta y dos testimonios utilizados durante el cotejo⁹, a pesar de que contengan (solamente) entre 1700 y 2000 caracteres.

⁹ Cuando realizamos el cotejo, sabíamos que se conservaban 42 códices. Más tarde hemos tenido conocimiento de un manuscrito medieval más, conservado en el Arnamagnænan Institute de Copenhague, gracias a la información de Rosa Rodríguez Porto, y de tres manuscritos modernos: el ms. BNE10344 (Del Camino Martínez 2021), un ms. conservado en la biblioteca Menéndez Pelayo de Santander y un ms. conservado en la Real Academia de la Historia. Los hemos llamado CAI, BNE10344, San y RAH, respectivamente. De este modo, el número de códices conservados del Fuero Juzgo asciende a 46, a los que hay que sumar 4 fragmentos que no contienen la ley 8.2.1.

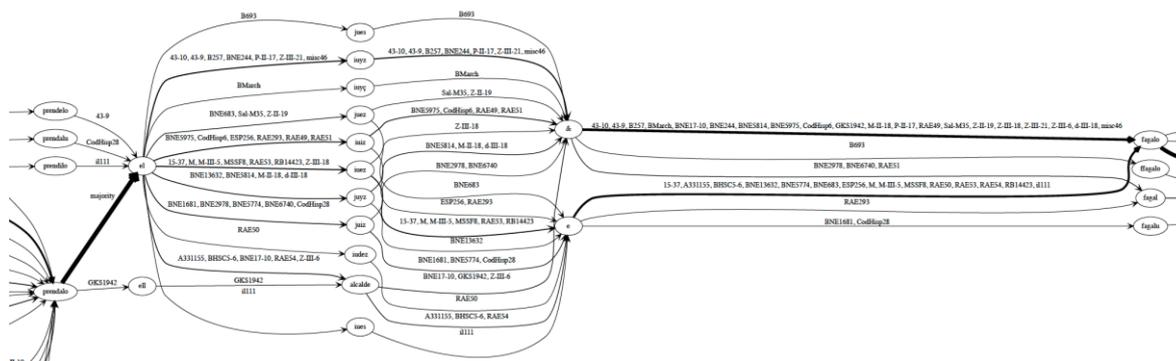


Figura 4. Grafo de variantes generado con CollateX.

En cambio, con la versión Python, los resultados obtenidos con el texto completo de los testimonios son más satisfactorios, quizás porque, desde 2014, el desarrollo y el mantenimiento del programa se hace únicamente en la versión Python (Nury 2018: 95). Sin embargo, esto requiere un paso más por parte del usuario: aprender a utilizar Python y crear un *script* específico para comparar los testimonios. De hecho, Elisa Nury (2018: 209-210) subraya que mientras «the advantages of using the Java version of CollateX in the command line environment were first that it was rather easy to install and run [...] the Python version requires a small amount of coding knowledge, which may be problematic for Humanities scholars who have not received the relevant training». Los programas escritos en el lenguaje Python se pueden utilizar de varias maneras. Existe un *shell* interactivo accesible desde el terminal (por ejemplo, IPython) y existen también los llamados Jupyter Notebooks, que permiten programar desde una interfaz web a través del navegador. Si se utiliza el *shell* interactivo, las posibilidades parecen más limitadas. El único *output* que hemos logrado visualizar en el terminal es una *alignment table* en formato ASCII que no se puede modificar si el algoritmo comete errores, de manera que, para ser capaz de generar otros tipos de visualizaciones, conviene aprender a utilizar un Jupyter Notebook. En esta interfaz se puede escribir directamente el *script* de Python – de manera más o menos intuitiva– y los resultados de las comparaciones aparecen directamente en el Notebook.

La gran ventaja de utilizar la versión Python de CollateX es que podemos atribuir a los *tokens* un valor n (forma normalizada), que se utilizará durante la etapa de alineamiento en vez del valor t (forma paleográfica), lo que contribuye significativamente a mejorar los resultados. Este aspecto es muy importante porque, como ya hemos visto, los únicos modos de interactuar con los resultados del alineamiento son o bien utilizar otra interfaz que permita hacerlo, o bien modificar manualmente los documentos CSV en un editor de hojas de cálculo y, por lo tanto, cuanto menos haya que corregir los resultados del alineamiento, mejor. El investigador puede crear otro Jupyter Notebook para generar, a partir de archivos en texto simple, un documento JSON pretokenizado en el que se podrá añadir cualquier propiedad a los diferentes *tokens* (Nury 2018: 239). Una vez creado el documento JSON pretokenizado con las formas normalizadas, este servirá como *input* en el Jupyter Notebook que se ha creado para hacer las comparaciones con CollateX. Lo que cambia respecto a la versión Java en línea de comandos es que el *input* JSON ya está tokenizado, de manera que los testimonios no van a pasar por el tokenizador automático de CollateX, sino que se procede directamente a la etapa de alineamiento. En el *script* del Jupyter Notebook el usuario puede elegir qué tipo de visualización quiere: por ejemplo, una *alignment table* HTML, dos tipos de grafos de variantes, un documento

CSV o un documento TEI/XML¹⁰. El *output* que más nos sirvió es el documento CSV que hemos corregido directamente en un editor de hojas de cálculo, y esta tabla de colación ha constituido nuestro documento de trabajo principal cuando hemos realizado la *examinatio* y la *selectio* de las variantes.

```
Entrée [1]: from collatex import *
import json
import graphviz

Entrée [2]: collation = Collation()
f = open("todos_mss_tokens.json", "r")
json_input = f.read()
print(json_input)

{
  "witnesses": [
    {
      "id": "RAE50",
      "tokens": {
        "e": "De",
        "n": "De"
      }
    }
  ]
}
```

Figura 5. Inicio del Jupyter Notebook creado para realizar comparaciones con la versión Python de CollateX.

Durante su tesis doctoral, Elisa Nury (2018) creó un Jupyter Notebook que sirve como interfaz para modificar e interpretar los resultados obtenidos por la versión Python de CollateX: PyCoviz. En primer lugar, el usuario puede modificar el resultado del alineamiento manualmente y de manera más o menos intuitiva gracias a unos *widgets* desarrollados por Nury (2018: 252-253). Además, se puede buscar en los resultados qué testimonios concuerdan entre sí en alguna lección (y qué testimonios se oponen a ellos en la misma lección) y cuáles tienen unas lecciones únicas en la tradición (Nury 2017: 320). De este modo, el investigador puede utilizar esta interfaz para aplicar el método lachmaniano a sus resultados. Sin embargo, las etapas previas necesarias para poder utilizar PyCoviz son bastante complejas: el usuario debe generar un *output* TEI/XML en Python y convertirlo en JSON gracias a una conversión XSLT precisa¹¹. El proceso para la preparación de los documentos que permiten utilizar esta interfaz es, entonces, relativamente complejo para un principiante, de modo que, por el momento, la dejaremos de lado. Sin embargo, en investigaciones futuras, podría ser muy útil –incluso necesario– aprender a manejarla.

2.3.3. Stemmaweb

Aunque no es un programa de colación propiamente dicho, la plataforma Stemmaweb, desarrollada por Tara L. Andrews y Joris van Zundert (2013), es una herramienta muy útil para nuestro caso de estudio. Permite reutilizar los resultados de CollateX y modificarlos gracias a la interfaz propuesta por Stemmaweb y, lo que es más interesante, permite generar un árbol filogenético con diferentes algoritmos que puede ayudar al filólogo a tener una idea general de la distancia entre los diferentes testimonios. Helena Bermúdez Sabel (2016: 27) define esta plataforma como «un conjunto de herramientas y recursos para una estematología empírica [que permite] la anotación y la diferenciación del tipo de variante, [además de] identificar las variantes con mayor relevancia estematológica».

Como precisa Christopher Howe (2012: 60-64), aunque el análisis filogenético no debe –y no puede– sustituir la crítica textual a la hora de establecer las relaciones entre un conjunto de testimonios, este tipo de herramienta permite al investigador tener una

¹⁰ Documentación de CollateX: <https://collatex.net/doc/> [última consulta: 18/01/2023].

¹¹ Repositorio Github del proyecto PyCoviz desarrollado por Elisa Nury: <https://github.com/enury/phd-automated-collation/tree/master/XSLT> [última consulta: 18/01/2023].

visión del conjunto de las variantes simultáneamente, puesto que el algoritmo que genera el árbol filogenético toma en consideración todas las variantes. Al igual que los algoritmos utilizados para el alineamiento, los algoritmos más utilizados para el análisis filogenético se inspiran en la bioinformática (en particular en el campo de la biología evolutiva) (Howe *et al.* 2012: 52).

La plataforma permite generar un árbol con dos algoritmos diferentes a partir de diferentes tipos de *outputs* obtenidos con CollateX. Son cuatro los algoritmos que se pueden elegir, pero dos de ellos no parecen funcionar y uno proporciona resultados inverosímiles, de manera que nos limitaremos a comentar los resultados obtenidos con el único algoritmo fiable: Neighbor-Joining. Christopher Howe describe el funcionamiento del algoritmo de este modo:

Each witness is represented as a row or column in a matrix, the elements in which show the calculated amount of difference between any pair. The computer then attempts to calculate the tree that gives the best fit between the observed amount of differences between pairs of witnesses and the values shown in the tree (with the lengths of branches in the tree representing the number of changes) (Howe *et al.* 2012: 54).

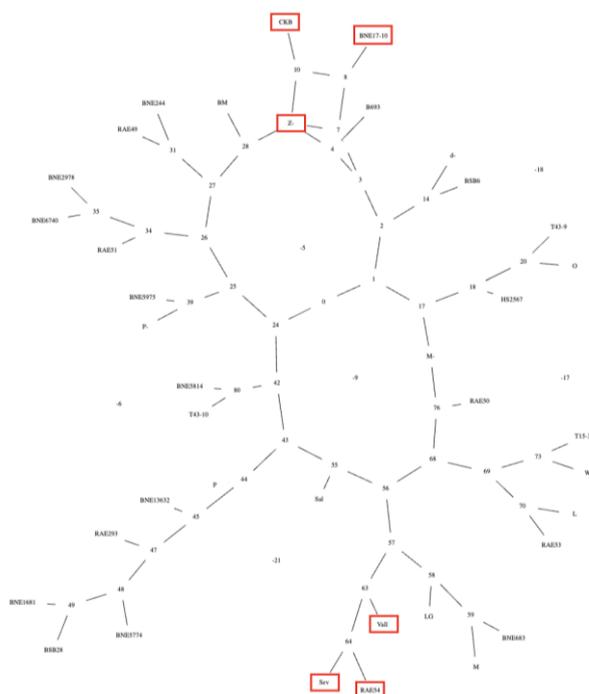


Figura 6. Árbol filogenético del Fuero Juzgo generado en Stemmaweb con el algoritmo Neighbor-Joining sin anotaciones sobre el tipo de variantes. En rojo aparecen los testimonios que pertenecen a la familia alfonsí.

Sin necesidad de entrar demasiado en los detalles, podemos intuir que el algoritmo Neighbor-Joining proporciona resultados interesantes, pero no muy fiables. De hecho, como Mónica Castillo Lluch ha podido demostrar en varias ocasiones (Castillo Lluch 2016; Castillo Lluch y Mabile 2021), en la tradición textual del Fuero Juzgo existe una familia de manuscritos que presenta un número muy elevado de innovaciones con respecto al resto de la tradición. Se trata de los manuscritos que aparecen en rojo en la figura 6, y podemos constatar que no aparecen todos juntos, sino en dos grupos de tres manuscritos.

En cuanto a los límites del algoritmo, no entendemos por qué aparecen solo 38 de los 42 manuscritos utilizados durante el cotejo. Además, parece que el carácter del guion

plantea problemas al algoritmo, pues las firmas de los manuscritos conservados en la Biblioteca del Monasterio de San Lorenzo de El Escorial aparecen separadas, de modo que no podemos estar seguros de qué nodo corresponde a qué manuscrito. Uno de los manuscritos que faltan en el árbol es el séptimo manuscrito de la familia alfonsí, el Z-II-9 o el Z-III-6 de El Escorial¹². Además, para obtener resultados más precisos, habría que modificar el grafo de variantes generado con CollateX y precisar, para cada lección, qué tipo de variantes aparecen y si tienen relevancia para la construcción del *stemma*. Se trata de una tarea bastante compleja que lleva mucho tiempo por dos motivos. En primer lugar, para que el algoritmo llegue a resultados satisfactorios, el usuario debe precisar el tipo de variantes para cada lección del texto. En nuestro caso, aunque nos centramos en una única ley del Fuero Juzgo, el número de variantes –sobre todo gráficas y ortográficas– es muy alto. En segundo lugar, la plataforma tiene algunos problemas de eficiencia. Cuando el usuario precisa el tipo de relación que existe entre las diferentes variantes de una lección, la plataforma tarda varios segundos en grabar las indicaciones. Si el usuario está trabajando con un par de frases, este problema de lentitud no afecta mucho a la duración del proceso; en cambio, cuando el texto es más largo, el investigador pierde un tiempo considerable esperando que sus anotaciones se graben.

A pesar de estos dos problemas, la posibilidad de interactuar de manera más o menos intuitiva con el grafo de variantes es la gran ventaja de Stemmaweb: no solo añade informaciones suplementarias en el grafo, que se tendrán en cuenta cuando el algoritmo analice las relaciones entre los testimonios, sino que también permite al usuario corregir el grafo, anotar las variantes y tener una mejor visualización de los resultados. Para limitar todo lo posible las intervenciones en cuanto al alineamiento en Stemmaweb, hemos generado con CollateX un documento en formato CSV y hemos corregido manualmente los resultados en un documento Excel. De este modo, hemos podido utilizar un documento perfectamente –o casi– alineado en Stemmaweb para no perder demasiado tiempo y centrarnos en la anotación de los diferentes tipos de variantes.

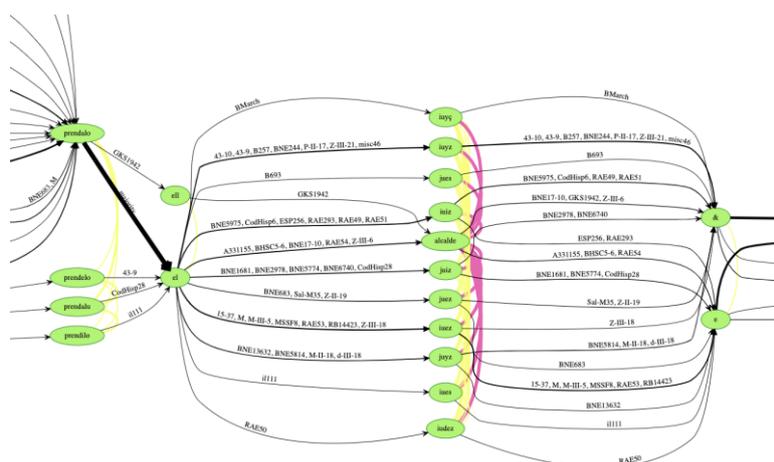


Figura 7. Grafo de variantes generado con CollateX y anotado en Stemmaweb. El amarillo corresponde a variantes ortográficas y morfológicas y el rosado a variantes léxicas (*juez* > *alcalde*). El tamaño de las líneas varía según la relevancia estematológica de la variante.

¹² Puesto que las dos firmas empiezan por la misma letra y que el resto de la firma se ha disgregado en la imagen, nos resulta imposible saber si el manuscrito que aparece en el árbol es el Z-II-9 o el Z-III-6.

Desafortunadamente, parece que la plataforma tiene fallos de funcionamiento. Después de haber anotado la totalidad del grafo precisando para cada lección el tipo de variantes que aparecían, los árboles filogenéticos obtenidos con el algoritmo Neighbor-Joining eran exactamente iguales al que aparece en la figura 6, lo que significa que el algoritmo no ha tenido en cuenta las indicaciones sobre el tipo de variantes. Tenemos entonces que contentarnos con comparar los testimonios sin distinción entre variantes ortográficas, morfológicas o sustantivas. No es tan preciso como hubiéramos esperado, pero nos da una idea general de qué manuscritos podemos comparar entre ellos para intentar establecer filiaciones. En la última parte de este trabajo, compararemos este árbol con los resultados a los que hemos llegado con métodos de filología clásica para valorar la utilidad de esta herramienta.

2.3.4. TRAViz

TRAViz es una librería JavaScript desarrollada por Stefan Jänicke (2014) en el marco del proyecto eTRACES¹³. Al igual que CollateX, no dispone de una interfaz para efectuar la colación. Sin embargo, es mucho más fácil de usar que CollateX, porque el usuario no debe utilizar el terminal o un Jupyter Notebook. El programa propone modelos que el usuario modifica integrando sus datos en el documento en formato JSON: «each item in the array is an edition, which contains both an identifier (or siglum), and a plain text version of the witness» (Nury 2018: 101).



```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-trans
2 <html>
3 <head>
4 <title>Basic Example</title>
5 <script src="../../traviz-min.js"></script>
6 <link rel="stylesheet" type="text/css" href="../../traviz.css"></link>
7 </head>
8 <body>
9 <div id="containerDiv"></div>
10 <script type="text/javascript">
11 var traviz = new TRAViz("containerDiv",{
12   lineBreaks: true,
13   connectionType: 'majority'
14 });
15 traviz.align({
16   edition: 'RAE49',
17   text: 'Todo omne que enciende casa allena en cibdat prendalo el iuiz & fagalo quemar & fagalli fazer
18 },{
19   edition: 'RAE51',
20   text: 'Todo omne que enciende casa ayena en cibdat prendalo el iuiz & fagal quemar & faga fazer emie
21 }

```

Figura 8. Ejemplo de código para hacer una colación con TRAViz.

Mientras que CollateX pone el acento sobre el alineamiento, TRAViz se centra sobre todo en proponer una visualización clara e intuitiva de los resultados, que reduce considerablemente el esfuerzo cognitivo del investigador a la hora de interpretarlos. El usuario tiene la posibilidad de personalizar la visualización, interviniendo directamente en el código JavaScript inserido dentro de un documento HTML que se puede abrir y visualizar en cualquier navegador. Como señala Stefan Jänicke (2015: 3), el objetivo de aumentar la legibilidad del grafo de variantes se alcanza mediante la utilización de colores (uno por cada testimonio), el tamaño de las palabras (cuanto más frecuente es una lección, más grande aparece en el grafo), una visualización lineal y la eliminación de elementos no necesarios (por ejemplo, los nodos y las flechas en CollateX). El usuario también puede añadir el parámetro *majority*, es decir, cuando una mayoría de testimonios presenta la misma lección, una única línea aparece para todos ellos, de manera que las variantes se destaquen más claramente y puedan identificarse más

¹³ Documentación de TRAViz: <http://www.traviz.vizcovery.org/tutorial.html> [última consulta: 18/01/2023]

2.4. Valoración crítica de los programas

Desgraciadamente, no existe un programa perfecto que disponga de todos los elementos necesarios para estudiar la tradición textual de una obra sin ninguna desventaja. En el caso de Juxta, a pesar de tener la interfaz más intuitiva y funcional, no permite comparar los testimonios sin manuscrito base. Por lo tanto, hemos utilizado este programa solo en la última fase de nuestra investigación, para comparar dos testimonios entre ellos. CollateX, en cambio, permite obtener resultados muy concretos gracias a los algoritmos que realizan alineamientos múltiples. Además, propone varios tipos de visualizaciones diferentes entre las cuales el filólogo puede elegir la que más le sirve. Por nuestra parte, hemos utilizado sobre todo el documento CSV, que puede abrirse con un editor de hojas de cálculo y, por lo tanto, permite corregir manualmente el alineamiento realizado por el programa. Se trata de un programa muy completo y sofisticado, pero es imposible disfrutar de todas sus posibilidades sin tener una formación previa en programación, puesto que carece de una interfaz intuitiva que permita al usuario efectuar la colación sin entrar en el código. Además, la documentación sobre estos programas se dirige sobre todo a personas que ya tienen conocimientos relativamente avanzados en informática. Por otra parte, además de la dificultad de manejar estos programas para realizar una *collatio* entre varios testimonios, otro problema es el de intervenir en las diferentes etapas, por ejemplo, en el alineamiento o en las visualizaciones, ya que hemos visto que se trata de una etapa compleja para un principiante. En este sentido, la plataforma Stemmaweb nos permitió anotar el grafo de variantes para refinar los resultados obtenidos con CollateX y generar otro tipo de visualización, los árboles filogenéticos, que han podido orientarnos hacia los grupos de testimonios que parecen más cercanos entre sí y que, por consiguiente, hemos podido comparar más en detalle para intentar establecer si guardan parentesco. Esta última herramienta no resulta demasiado útil para la *collatio codicum*, sino más bien para la *constitutio stemmatis*, aunque no debemos perder de vista que los resultados no deben considerarse como definitivos.

Una desventaja general de los métodos de colación asistida por ordenador es que los programas y las plataformas necesitan una infraestructura y soporte técnico –y, por lo tanto, fondos– que garantice su mantenimiento. De hecho, varios programas que hemos intentado utilizar en este trabajo ya no funcionan o presentan algunos problemas. Es el caso de la versión en línea de Juxta, Juxta Commons, a la que ya no se puede acceder, incluso si algunas referencias bibliográficas recientes a las que hemos acudido la citan como operativa. Lo mismo ocurre con Stexaminer, una herramienta de Stemmaweb mencionada por Elisa Nury (2018: 249), que parecía permitir al usuario observar las variantes que concordaban con el árbol filogenético generado con el algoritmo gracias a una tabla de colación que enseñaba el texto entero, con colores diferentes si la variante concordaba –o no– con el árbol. No es la única herramienta que falla en Stemmaweb; de hecho, dos de los cuatro algoritmos propuestos para el análisis filogenético no funcionan y solo Neighbor-Joining parece dar resultados más o menos pertinentes. Por consiguiente, aunque los programas que acabamos de describir nos han resultado muy útiles en ciertas etapas de nuestra investigación, es posible que dentro de unos años ya no sea posible utilizarlos. Además, la imposibilidad de saber de antemano si algunas herramientas funcionan o no puede tener consecuencias en la investigación que se está llevando a cabo. En nuestro caso, perdimos muchas horas en anotar el grafo de variantes

en Stemmaweb para darnos cuenta de que era inútil, ya que el algoritmo no parece tener en cuenta dichas anotaciones.

En definitiva, concordamos con Helena Bermúdez Sabel (2016: 20) cuando escribe que «a pesar de las ventajas que los métodos computacionales nos ofrecen (entre otras, mayor rapidez y, sobre todo, menos margen para el error) su implementación para [la *collatio codicum*] aún es escasa». Además, el filólogo no debe olvidar que las conclusiones a las que llegamos utilizando esos programas no pueden tomarse como verdades incontestables puesto que, como acabamos de ver, la mayoría de los algoritmos utilizados por esos programas funcionan con el método heurístico, que da resultados plausibles, pero no definitivos. Quizás sería imaginable programar un algoritmo que compare todos los testimonios simultáneamente, pero, según hemos comprobado, sería demasiado costoso computacionalmente y el proceso de colación duraría muchísimo tiempo. Por lo tanto, como precisa Elisa Nury (2018: 89), «it is important to always be critical of results obtained with computational methods, and not draw hasty conclusions based on those results, but rather compare the outputs of different methods», por lo que el investigador, con su competencia filológica, sigue desempeñando un papel importantísimo en todo el proceso que acabamos de describir en este capítulo. En este sentido, Philipp Roelli y Dieter Bachmann (2010: 322) hablan de la necesidad de un «happy marriage of our human philological judgment with the computing power of our algorithms».

Una mejora considerable de esta nueva rama de la filología que, por cierto, llega a ser cada vez más necesaria e imprescindible para estudiar la tradición textual de una obra, consistiría en el desarrollo de un programa completo que concentre todas las ventajas de los programas que acabamos de describir y que los incluya en una interfaz fácil de uso para cualquier filólogo, sin que sea necesario tener una formación previa. De hecho, todas las herramientas existentes «demandan distintos tipos de sacrificio al investigador, sea en la intensidad de la curva de aprendizaje sea en la calidad de los resultados» (Bermúdez Sabel 2016: 31). Todavía no existe la solución perfecta que combine resultados completos y cualitativos con una utilización simple e intuitiva. Comparando los mismos programas, Bermúdez Sabel (2016: 31) llega a la conclusión de que «con una tradición compleja y con objetivos muy específicos, quizás el esfuerzo que implica el dominio de una herramienta como CollateX sea recompensado [mientras que] bajo otras circunstancias, un software intuitivo que nos ofrece resultados inmediatos puede ser la mejor opción», como Juxta, por ejemplo. Elisa Nury (2018: 103) concuerda con Bermúdez Sabel en cuanto al abismo que separa estos dos programas cuando precisa que «CollateX seems to be adopted by big projects with support from professional developers», mientras que Juxta parece más adaptado para estudiantes o investigadores que no han recibido una formación técnica adecuada.

2.5. Método de filología clásica: *examinatio*, *selectio*, y *constitutio stemmatis*

Ya advertimos que los resultados de nuestra *collatio* no aparecerán de manera exhaustiva en este artículo, pues nuestro objetivo aquí consiste en comparar los resultados de los dos métodos (filología digital *versus* filología clásica) para valorar el aporte de las herramientas digitales y no tanto en justificar el esbozo (provisional) de *stemma codicum*

que hemos podido trazar tras el análisis de la ley 8.2.1¹⁶. Sin embargo, para cumplir con nuestro objetivo, es esencial describir la metodología que hemos seguido para llegar a nuestras conclusiones que, pese a su provisionalidad, nos permiten identificar las diferentes familias de manuscritos con relativa seguridad.

2.5.1. Marco teórico y tipos de variantes utilizadas

A diferencia del método lachmaniano, hemos estudiado el parentesco entre los testimonios basándonos no solo en los errores comunes presentes en algunos manuscritos, sino también en variantes que no son errores sino innovaciones o variantes lingüísticas. De hecho, el término *error* puede llevar a confusión puesto que, como dice Tomás Fernández (2019: 121), las variantes que el filólogo debe utilizar son «cualquier tipo de innovación frente a la lectura del ancestro, aunque dicha innovación no sea errónea en sí misma». Fernández cita a Henri Quentin, uno de los precursores del método filológico basado en las variantes –y no en el error–:

Je rejette, dès la première minute, toute visée vers la leçon primitive. Je ne connais ni erreurs, ni fautes communes, ni bonnes ni mauvaises leçons, mais seulement des formes diverses du texte, sur lesquelles, par une méthode qui s'appuie sur des statistiques rigoureuses, je délimite d'abord les familles, puis je classe les manuscrits dans chacune d'elles, et enfin les familles entre elles (Quentin 1926: 37, *op. cit.* en Fernández 2019: 122).

Y esto es aún más válido para un caso de estudio como el nuestro. De hecho, el texto del Liber Iudiciorum y, por tanto, el del Fuero Juzgo, se fue creando poco a poco, en diferentes épocas y por diferentes autores. Por consiguiente, no estamos frente a una tradición empezada por un autor único, por lo que resulta muy difícil –si no imposible– saber, en los lugares críticos donde aparecen varias lecciones aceptables (variantes adiaforas o equipolentes), cuáles son las lecciones correctas y cuáles son los errores, puesto que no podemos basarnos en argumentos lingüísticos o en el *usus scribendi* del autor de un texto ‘original’.

Además, hemos basado nuestro análisis no solo en los errores de copia evidentes y en las variantes textuales, sino también en algunas variantes de lengua (morfosintácticas y sintácticodiscursivas, según la tipología que aparece en Fernández-Ordóñez 2002: 115-128 y 2019: 380). Aunque puede parecer metodológicamente discutible, si muchas variantes de lengua –que tradicionalmente no suelen utilizarse para establecer filiaciones (Fernández-Ordóñez 2002: 108)– van en la misma dirección, nos parece sensato pensar con Andrews y Nury (*vid. supra*) que nos pueden dar informaciones importantes sobre las relaciones que existen entre los manuscritos. De hecho, en una pequeña porción de texto como de la que nos ocupamos, si dos –o más– variantes *a priori* poligenéticas aparecen en los mismos manuscritos, resulta altamente improbable que estos manuscritos concuerden en estas variantes independientemente el uno del otro. Son muy pocos los errores significativos que, tradicionalmente, permiten establecer con seguridad las filiaciones entre los manuscritos, de modo que cuando varios elementos teóricamente poligénéticos van en la misma dirección, nos parecen indicios suficientes

¹⁶ Hemos elegido la ley 8.2.1 porque hemos podido constatar, tras estudiar el libro 8 del Fuero Juzgo en el BNE683 (*Cfr.* Castillo Lluç y Mabilie 2021), que esta ley presentaba muchas variantes textuales.

para hacernos una idea de las filiaciones que existen entre los testimonios y, de hecho, para dejar de considerarlos poligenéticos. Por ejemplo, si *stricto sensu* un homoioteleuton no puede utilizarse para afirmar que dos testimonios están relacionados genéticamente, si existen otros indicios que parecen confirmar el parentesco, el homoioteleuton se convierte en una prueba relativamente sólida.

Cabe precisar que no tenemos la pretensión de llegar a conclusiones definitivas y de presentar un *stemma codicum* del Fuero Juzgo empíricamente válido, sino un esbozo de resultados provisionales que tendrán que verificarse en trabajos futuros. Se trata de un sondeo en una ley (8.2.1) para observar el comportamiento de los testimonios en el marco de ese texto para esbozar un *stemma codicum* y comenzar a describir las filiaciones entre manuscritos de los que hasta el momento se ignora casi todo de su parentesco. Por otra parte, el filólogo que pretende trazar un *stemma codicum* seguro tiene que estudiar el texto en su globalidad para localizar los casos de contaminación (Blecuá 1983: 45). Y esto es aún más verdadero en nuestro caso de estudio, puesto que los textos tenían una utilidad concreta y los copistas, a la hora de copiar un manuscrito, probablemente tenían a la vista varios testimonios para estar seguros de no equivocarse y dar lugar a incoherencias. Es, por lo tanto, más que esperable que la tradición manuscrita del Fuero Juzgo presente casos de contaminación.

En definitiva, los indicios que hemos utilizado para separar los cuarenta y dos manuscritos que hemos tenido en cuenta –incluidos los modernos– en ocho familias de manuscritos son: 1) los errores y las innovaciones sustanciales (variantes léxicas, omisiones/adiciones importantes); 2) los homoioteleuta; 3) las variantes de lengua significativas (inversión del orden sintáctico, perífrasis verbal o no, singular vs. plural, etc.); y 4) la acumulación de variantes de lengua individualmente no significativas. En cambio, las variantes que no hemos utilizado para la *constitutio stemmatis* son: 1) las variantes gráficas u ortográficas; 2) la separación de palabras; 3) las variantes dialectales (castellano *versus* asturleonés)¹⁷; y 4) la alternancia entre los sustantivos *casa* y *cosa*. En cuanto a esta última, no la hemos tenido en cuenta porque observamos una alternancia caótica entre los dos sustantivos –que presentan muchas ocurrencias en la ley 8.2.1– puesto que las gráficas <a> y <o> se parecen mucho y los copistas parecen caer muy a menudo en el error, de modo que se trata de un error que puede muy fácilmente tener carácter poligenético.

2.5.2. Las ocho familias del Fuero Juzgo

Para realizar la *examinatio* y la *selectio* de las variantes útiles para la *constitutio stemmatis*, hemos trabajado con el documento CSV generado con CollateX que contiene el texto de todos los testimonios alineados. En amarillo aparecen las variantes comunes a dos –o más– manuscritos, en rojo las variantes que aparecen en un único manuscrito (*lectiones singulares*), en azul las transposiciones y en naranja las particularidades de la rama alfonsí.

En primer lugar, hemos separado la tradición en dos ramas a partir de un lugar crítico muy llamativo, en el que la mitad de los testimonios lee *iure* y la otra *diga*. Hemos

¹⁷ No hemos utilizado las variantes dialectales para establecer filiaciones entre los manuscritos porque son variantes de lengua que de entrada consideramos poligenéticas.

llamado A a la rama que presenta la lección *diga* y B a la rama que presenta la lección *iure* (y sus alógrafos).

A	LA	LB	LC	LD	LE	LF	LG	LH	LI	LJ	LK	LL
RAMA A												
RAE50	y		el	senno	de	la	casa	diga			ante	testimonias
RAE51	&		el	senno	de	la	casa	diga			ante	testimonios
M-III-5	y		el	senno	de	la	casa	diga			ante	testimonias
Z-III-21	&		el	senno	de	la	casa	diga			ante	testimonias
T15-37	e											
RAE49	&		el	senno	de	la	casa	diga			ante	testemunias
LG	y		el	senno	de	la	casa	diga			ante	testimonias
RAE293	e		el	senno	de	la	casa	diga			ante	testimonias
RAE53	e											
Z-III-18	y		el	senno	de	la	casa	diga			ante	testimonias
L	e											
P-II-17	&		el	senno	de	la	casa	diga			ante	testimonias
E	&		el	senno	de	la	casa	diga			ante	testimonias
M	y		el	senno	de	la	casa	diga			ante	testimonias
BNE244	&		el	senno	de	la	casa	diga			ante	testemunias
W	e											
BSB28	e		el	senno	de	la	casa	diga			ante	testemunias
BSB6	e											
BNE683	y		el	senno	de	la	casa	diga			ante	testimonias
BNE1681	e		el	senno	de	la	casa	diga			ante	testemunias
BNE6740	&		el	senno	de	la	casa	diga			ante	testimonias
Sal	y		el	senno	de	la	casa	diga			ante	testimonias
BNE2978	&		el	senno	de	la	casa	diga			ante	testimonias
RAMA B												
Z-II-9	y		el	senno	de	la	casa	iure			ante	testimonias
O	&		el	senno	de	la	casa	iure			ante	testimonias
CKB	&		el	senno	de	la	casa	iure			ante	testimonias
RAE54	y		el	senno	de	la	casa	iure			ante	testimonias
HS2567	&		el	senno	de	la	casa	iure			ante	testimonias
Z-II-6	&		el	senno	de	la	casa	iure			ante	testimonias
T43-10	&		el	senno	de	la	casa	iure			ante	testimonias
P-II-18	&		el	senno	de	la	casa	iure			ante	testimonias
T43-9	&		el	senno	de	la	casa	iure			ante	testimonias
P	e		el	senno	de	la	casa	iure			ante	testimonias
BNE5975	&		el	senno	de	la	casa	iure	que	perdeo	ante	testimonias
BNE17-10	y		el	senno	de	la	casa	iure			ante	testimonias
M-II-18	&		el	senno	de	la	casa	iure			ante	testimonias
BN	et		el	senno	de	la	casa	iure			ante	testimonias
BNE13632	e	iure										
BNE5774	e		el	senno	de	la	casa	iure			ante	testimonias
Sev	y		el	senno	de	la	casa	iure			ante	testimonias
BNE5814	&		el	senno	de	la	casa	iure			ante	testimonias
Vañ	y		el	senno	de	la	casa	iure			ante	testimonias

Figura 10. Separación de la tradición en dos ramas¹⁸.

No vamos a detenernos demasiado en argumentar la validez de esta separación, porque esto supondría revisar el documento entero para constatar que no hay ninguna variante sustancial que aparezca tanto en manuscritos de la rama A como en manuscritos de la rama B. Los casos en los que aparecen variantes de lengua en manuscritos de las dos ramas pueden explicarse por poligénesis.

Sin embargo, podemos observar en este fragmento (Figura 10) dos elementos significativos que parecen confirmar la validez de la separación. En primer lugar, los siete manuscritos que presentan muchas innovaciones –que hemos llamado alfonsíes más arriba– aparecen todos en la rama B. El hecho de que estos manuscritos, emparentados con total seguridad por las innovaciones que tienen en común, aparezcan todos en la misma rama es un indicio muy potente de la validez de esta separación. En segundo lugar, otro argumento a favor de esta separación es la presencia, en cuatro manuscritos de la rama A –T15-37, RAE53, L y W¹⁹–, de un homoioteleuton muy significativo, pues el sintagma omitido es el sujeto de la proposición. En la ley 8.2.1, que trata de las personas que queman casas ajenas, se dice que la persona cuya casa ha sido quemada debe jurar (o decir) ante testigos cuánto ha perdido en el incendio. Sin este sujeto, la ley pierde totalmente su sentido: sería el pirómano quien debería decir cuánto ha perdido. No se trata entonces de un homoioteleuton cualquiera, y el hecho de que los cuatro manuscritos –también emparentados por otros indicios– aparezcan todos en la rama A constituye otra muestra de la validez de nuestra separación. En cuanto a BSB6, que presenta también una omisión, no podemos emparentarlo con los cuatro manuscritos a los que acabamos de referirnos porque se trata de un homoioteleuton más largo que,

¹⁸ El orden en que aparecen los manuscritos dentro de las dos ramas es aleatorio.

¹⁹ Hemos establecido, para cada manuscrito, una denominación más económica para evitar citar siempre su signatura completa y, por lo tanto, agilizar la argumentación. Las signaturas completas y el archivo en que están custodiados aparecen en la bibliografía.

con toda probabilidad, no puede relacionarse con la omisión del sintagma *el señor de la casa*.

Antes de seguir con la creación de las familias, cabe que nos detengamos en los dos *stemmata* que podemos construir a partir de nuestra división. De hecho, el lugar crítico en el que hemos basado la división no presenta un error separativo, sino una variante adiáfora, por lo que no sabemos cuál sería la lección original y cuál sería la lección innovadora²⁰. Por lo tanto, el inicio del *stemma* podría tener dos formas diferentes. Si consideramos que la lección original es *diga*, el *stemma* se presentaría de este modo:

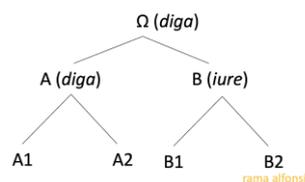


Figura 11. Separación de la tradición en dos ramas, primera posibilidad de *stemma*.

En cambio, si consideramos que la lección original es *iure*, el *stemma* podría empezar de este modo:

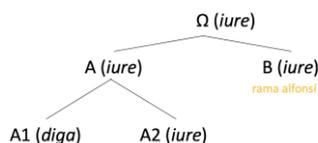


Figura 12. Separación de la tradición en dos ramas, segunda posibilidad de *stemma*.

No podemos, por el momento, saber cuál es el *stemma* correcto, porque no tenemos otra lección que emparente la rama alfonsí con el resto de la rama B. Lo ideal sería encontrar, en otra ley, un error común conjuntivo entre todos los manuscritos de la rama B, que sería separativo frente a todos los manuscritos de la rama A. Sin embargo, eso no cambia las etapas siguientes de nuestro análisis. De hecho, el análisis de las subramas sería exactamente el mismo si seguimos el primer *stemma* o si seguimos el segundo: lo único que cambiaría sería el resultado final del *stemma* en su parte superior (por ahora no tan prioritaria para nosotros y de todos modos provisional hasta explorar la tradición latina). En el segundo *stemma*, la subrama A2 corresponde a la subrama B1 del primer *stemma*, de la que forman parte los manuscritos que presentan la lección *iure* pero que no pertenecen a la rama alfonsí. La subrama A1 del segundo corresponde a la rama A del primero, es decir, todos los manuscritos que presentan la lección *diga*. Por lo tanto, si en una investigación futura nos diéramos cuenta de que la lección original es *iure*, solo tendríamos que desplazar estas subramas de un lado o del otro del *stemma* para corregir el resultado final.

Además, podríamos imaginar que esta alternancia léxica se observara también en los códices latinos, que no hemos utilizado en este trabajo. La edición del Liber Iudiciorum

²⁰ No estamos contando con la tradición latina de la que deriva el Fuero Juzgo. Por el momento nuestro razonamiento se construye independientemente de la tradición del Liber. En una segunda etapa, esta cabeza del *stemma* habrá que revisarla en función de las relaciones que descubramos con los escritos latinos.

de Karl Zeumer de 1902, que, según Antonio Pérez Martín (2002: 48), se considera como edición de referencia, no presenta ni *iurare* ni *dicere*, sino *iurisiurandi*:

perdiderunt. Et si post datum sacramentum periurasse noscuntur, hec, que amplius exegerint, in duplum ei, quem fraudulenta ambitione damnare voluerant, cogantur exolvere. Qui vero extra civitatem in domum mittit incendium, omnia, que sunt in
15 domo incensa, restituat et pretium domino domus reddere non moretur. Sed his, cuius domus incensa est, coram testibus iurisiurandi interpositione confirmet, se non amplius postulare, quam perdiderat. Et si post datum sacramentum amplius, quam perdiderat, exigisse dinoscitur, in duplum ei, qui damna sustinuit, coactus exolvat.

Figura 13. Fragmento de la ley 8.2.1 en la edición de Karl Zeumer (1902: 319).

Aunque *iurare* y la lección elegida por Zeumer provienen del mismo étimo, el significado de los dos verbos es sensiblemente diferente. *iurisiurandi* corresponde en español a *prestar juramento*, y no a *jurar*. Tendríamos que comprobar en los manuscritos latinos si este lugar presenta variación, pero si el manuscrito –o los manuscritos– que sirvieron para la traducción –o las traducciones– leía(n) *iurisiurandi*, podríamos imaginar que hubo una persona que tradujo con el verbo *dezir* y otra persona que tradujo con el verbo *iurar*, que corresponderían a los arquetipos de las dos ramas. Zeumer (1902: 319), que utiliza varios códices para la *constitutio textus* de su edición, no indica ninguna variante para este lugar crítico, salvo un códice que presenta la variante de lengua *iurisiurandi*, lo que significa que ninguno de los códices utilizados por Zeumer presenta una lección basada en el verbo *dicere*. Por lo tanto, no podemos saber si la lección *diga* es el resultado de una traducción diferente a partir del códice latino (*stemma* 1, Figura 11) o si se trata de un desvío de la lección *iure* presente en los demás códices romances (*stemma* 2, Figura 12).

Una vez separada la tradición en dos ramas, hemos podido separar cada rama en dos subramas (A1, A2 y B1, B2). En cuanto a la rama A, la hemos separado a partir de un lugar crítico en el que aparece una variante de lengua significativa. Hemos llamado A1 a la subrama que presenta la lección con el sintagma en singular –*la casa*– y A2 a la subrama que presenta la lección con el sintagma en plural –*las casas*–.

A	PZ	QA	QB	QC	QD	QE	QF	QG
A1								
Z-III-21	el	que	encendio	la	casa	reciba	.c.	azotes
T15-37	el	que	encendio	la	casa	reciba	.c.	açotes
RAE49	el	que	encendio	la	casa	reciba	.c.	azotes
LG	el	que	encendio	la	casa	reçiba	.c.	açotes
RAE293	el	que	encendio	ela	casa	reciba	.c.	azotes
RAE53	el	que	encendio	la	casa	reciba	.c.	açotes
L	el	que	encendio	la	casa	reciba	.c.	azotes
M	el	que	encendio	la	casa	reciba	.c.	açotes
W	el	que	encendio	la	casa	reciba	.c.	açotes
BNE683	el	que	encendio	la	casa	reciba	.c.	azotes
BNE13632	el	que	ençendio	la	casa	reçiba	cient	açotes
A2								
RAE50	el	que	encendio	las	casas	reciba	.c.	azotes
RAE51	el	que	encendio	las	casas	reciba	.c.	azotes
M-III-5	el	que	encendio	las	casas	reciba	ciento	açotes
Z-III-18	el	que	encendio	las	casas	reciba	.c.	açotes
P-II-17	el	que	ençendio	las	casas	reçiba	.c.	açotes
E	el	que	ençendio	las	casas	rreçiba	cient	açotes
BNE244	el	que	encendio	las	casas	reciba	.c.	azotes
BSB28	el	que	ençendio	las	casas	reciba	.c.	azotes
B5B6	el	que	ençendio	las	casas	reçiba	.c.	açotes
BNE1681	el	que	ençendio	las	casas	reciba	.c.	azotes
BNE6740	el	que	encendio	las	casas	reciba	.c.	azotes
Sal	el	que	ençendio	las	cassas	reçiva	cient	açotes
BNE2978	el	que	encendio	las	casas	reciba	.c.	azotes

Figura 14. Separación de la rama A en dos subramas.

La subrama A1 puede a su vez separarse en dos familias gracias al homoioteleuton que ya hemos mencionado (Figura 15) y la adición de una conjunción copulativa en los mismos cuatro manuscritos (Figura 16). La segunda es una variante que, en otro contexto, no tendría mucho peso estematológico, pero, puesto que concuerda con el homoioteleuton común a los cuatro manuscritos, nos puede servir para confirmar nuestra división.

A	KV	KW	KX	KY	KZ	LA	LB	LC	LD	LE	LF	LG	LH
Grupo 1													
T15-37	al	sennor	de	la	casa	e							diga
RAE53	al	sennor	de	la	casa	e							diga
L	al	sennor	de	la	casa	e							diga
W	al	sennor	de	la	casa	e							diga
Grupo 2													
Z-III-21	al	sennor	de	la	casa	&		el	sennor	de	la	casa	diga
RAE49	al	sennor	de	la	casa	&		el	sennor	de	la	casa	diga
LG	al	sennor	de	la	casa	y		el	sennor	de	la	casa	diga
RAE293						e		el	sennor	de	la	casa	diga
M	al	sennor	de	la	casa	y		el	sennor	de	la	casa	diga
BNE683													
BNE13632						e	jure	el	sennor	de	la	casa	diga

Figura 15. Separación de la rama A1 en dos grupos (1)²¹.

A	NN	NO	NP	NQ	NR
Grupo 1					
T15-37	derredor	e	sí	alguna	cosa
RAE53	derredor	e	sí	alguna	cosa
L	derredor	e	sí	alguna	cosa
W	derredor	e	sí	alguna	cosa
Grupo 2					
Z-III-21	aderredor		se	alguna	cosa
RAE49	derredor		si	alguna	cosa
LG	derredor		si	alguna	cosa
RAE293	derredor		si	alguna	cosa
M	derredor		si	alguna	cosa
BNE683					
BNE13632	derredor		si	alguna	cosa
BNE13632					
	redor		se	alguna	cosa

Figura 16. Separación de la rama A1 en dos grupos (2).

La subrama A2 también puede separarse en dos familias a partir de un lugar crítico que presenta una variante de lengua significativa que no aparece en ningún otro manuscrito en toda la tradición: la construcción perifrástica *devia fazer* –y sus alógrafos– se sustituye por una solución en pretérito imperfecto –*fazia* y sus alomorfos– sin el verbo modal.

²¹ A pesar de presentar la lección *jure* (columna LB), hemos integrado BNE13632 en la rama A. En efecto, por su estrecha cercanía con RAE293 (varias lecciones en común que no aparecen en ningún otro manuscrito), nos parece improbable que esos dos manuscritos pertenezcan a dos familias distintas. BNE13632 es un manuscrito moderno (finales del XVI – inicios del XVII), de modo que podríamos imaginar que su copista tuviera a la vista varios códices: un descendiente perdido de RAE293 y otro de la rama B (Mabille 2022: 68, 77-79 y 97-99).

A	MU	MV	MW	MX	MY	MZ	NA	NB
Grupo 1								
RAE50	aquel		quel	devia		fazer	el	emienda
M-III-5	aquel		quel	devia		fazer	el	emienda
Z-III-18	aquel		quel	deve		fazer	el	emjenda
E	aquel		quel	deve		ffazer	la	emjenda
BSB28	aquel		que	devia		fazer	la	emenda
BNE1681	aquel		que	debia		facer	la	emienda
Grupo 2								
RAE51	aquel		que			fazie	la	emienda
P-II-17	aquel		que			fazia	la	emenda
BNE244	aquel		que			faz	la	emenda
BSB6	aquel	a	quien			façie	la	emienda
BNE6740	aquel		que			facie	la	emienda
Sal	aquel		quien			fazia	la	hemienda
BNE2978	aquel		que			facie	la	emienda

Figura 17. División de la rama A2 en dos grupos.

En cuanto a la rama B, no la hemos separado en dos subramas en base a un único lugar crítico, sino a todas las innovaciones de los manuscritos alfonsíes. Hemos llamado B1 a la subrama que contiene los manuscritos que presentan la lección *iure*, pero no las innovaciones sustanciales presentes en los manuscritos alfonsíes, y B2 a la subrama que contiene los manuscritos alfonsíes. A continuación, aparece una de estas innovaciones.

A	LH	LI	LJ	LK	LL	LM	LN	LO	LP	LQ	LR	LS	LT	LU	LV	LW
B1																
O	iure			ante	testimonias	lo		que								perdeo
HS2567	iure			ante	testimonias			que	lo							perdio
T43-10	iure			ante	testimonias	lo		que								perdio
d-III-18	iure			ante	testimonias	lo		que								perdio
T43-9	iure			ante	timonias	lo		que								perdeo
P	iure			ante	testimonias	lo		que								perdio
M-II-18	iure			ante	testimonias	lo		que								perdio
BM	iure			ante	testimonias	lo		que								perdio
BNE5975	iure	que	perdeo	ante	testimonias											
BNE5814	iure			ante	testimonias	lo		que								perdio
BNE5774	iure			ante	testimonias	lo		que								perdiu
B2																
Z-II-9	iure			ante	testimonias			que	non	tome	mas	de		quanto		perdio
CKB	iure			ante	testimonias			que	non	tome	mas	de	lo	que		perdio
RAE54	yure			ante	testimonias			que	non	tome	mas	de	lo	que		perdio
Z-III-6	iure			ante	testimonias			que	non	tomo	mas	de	lo	que		perdio
BNE17-10	yure			ante	testimonias			que	non	tomo	mas	de	lo	que		perdio
Sev	yure			ante	testimonias			que	non	tome	mas	de	lo	que		perdio
Vall	iure			ante	testimonias			que	non	tome	mas	de	lo	que		perdio

Figura 18. División de la rama B en dos ramas (B1 y B2), con una de las innovaciones de la rama B2.

En cuanto a la subrama B1, puede separarse en dos familias a partir de la adición del verbo *dar* en imperativo con el pronombre enclítico de objeto directo en cuatro manuscritos: T43-9 y T43-10, d-III-18 y O (Figura 19). Esta adición –que no aparece en ningún otro manuscrito en toda la tradición– no cambia mucho el sentido de la oración, porque el complemento indirecto *al sennor de la casa* está regido, en los demás manuscritos, por el verbo *entregar* en imperativo que aparece antes en la oración. El verbo *entregar* aparece también en los cuatro manuscritos que contienen la adición: se trata entonces de una repetición que sirve para aclarar el sentido de la oración porque el verbo *entregar* rige varios complementos y el lector desatento podría tener dificultades para entender la frase²².

²² Texto de O: [...] *quien enciende casa fuera de cibdat entregue todo quanto se perdeo en la casa & el precio de la casa delo al sennor de la casa* [...].

A	KP	KQ	KR	KS	KT	KU	KV	KW	KX	KY	KZ
Grupo 1											
O	el	precio	de	la	casa	delo	al	senhor	de	la	casa
T43-10	el	precio	de	la	casa	dela	al	senhor	de	la	casa
d-III-18	el	prescio	de	la	casa	dela	al	senhor	de	la	casa
T43-9	el	precio	de	la	cosa	delo	al	senhor	de	la	casa
Grupo 2											
HS2567	el	precio	de	la	casa		al	senhor	de	la	casa
P	el	prezio	de	la	casa		al	senhor	de	la	casa
M-II-18	el	precio	de	la	casa		al	senhor	de	la	casa
BM	el	precio	de	la	casa		al	senhor	de	la	casa
BNE5975											
BNE5814	el	precio	de	la	casa		al	senor	de	la	casa
BNE5774											

Figura 19. Separación de la rama B1 en dos grupos.

La división de B1 en dos familias está confirmada por la adición, en los mismos manuscritos, de una preposición (*a*) o de un pronombre adverbial (*ende*) (Figura 20):

A	OC	OD	OE	OF	OG	OH
Grupo 1						
O	quemo	devese	a	entregar	de	aquello
T43-10	quemo	devese	a	entregar		daquello
d-III-18	quemo	devense	ende	entregar	de	aquello
T43-9	quemo	devese	a	entregar	de	aquello
Grupo 2						
HS2567	quemo	devense		entregar	de	aquello
P	quemo	devensse		entregar		daquello
M-II-18	quemo	devese		entregar		daquello
BM	quemo	deve		entregar		daquello
BNE5975	quymo	devese		entregar		daquello
BNE5814	quemo	devense		entregar		daquello
BNE5774	quimo	devense		entregar		daquello

Figura 20. Prueba 2 de la validez de la separación de B1 en dos grupos.

En cuanto a la subrama B2, podemos separarla en dos familias gracias a un homoioteleuton (Figura 21) –así como tres variantes de lengua individualmente no significativas– comunes a Z-III-6, Z-II-9 y CKB. Se trata de la adición del adverbio de lugar *y* (Figura 22), del pronombre de objeto indirecto *le* (Figura 23) y la presencia del artículo definido *la*, mientras que los demás manuscritos de B2 presentan su alomorfo *e/* (Figura 24).

A	FL	FM	FN	FO	FP	FQ	FR	FS	FT	FU	FV
Grupo 1											
Z-II-9	de	las	casas								partan
CKB	de	las	casas								partan
Z-III-6	de	las	casas	&							partan
Grupo 2											
RAE54	de	las	casas	e	segund	la	valia	de	las	casas	partan
BNE17-10	de	las	cosas	&	segunt	la	valia	de	las	casas	partan
Sev	de	las	casas	e	segund	la	valia	de	las	casas	partan
Vall	de	las	casas	e	segun	la	valia	de	las	casas	partan

Figura 21. División de la rama B2 en dos grupos.

A	CG	CH	CI	CJ	CK	CL	CM	CN
Grupo 1								
Z-II-9	non	diga	mas	de	lo	que	y	avia
CKB	non	diga	mas	de	lo	quanto	y	avia
Z-III-6	non	diga	mas	de	lo	que	y	avia
Grupo 2								
RAE54	non	diga	mas	de	lo	que		avie
BNE17-10	non	diga	mas	de	lo	que		avia
Sev	non	diga	mas	de	lo	que		avie
Vall	non	diga	mas	de	lo	que		avie

Figura 22. Prueba 1 de la validez de la división de B2 en dos grupos.

A	LY	LZ	MA	MB	MC	MD	ME	MF	MG
Grupo 1									
Z-II-9	sy	despues	le	puhier	sseer	provado	que	demando	mas
CKB	ssi	depues	le	puhiere	ser	provado	que	demando	mas
Z-III-6	si	despues	le	puhiere	seer	provado	que	demando	mas
Grupo 2									
RAE54	si	depues		puhier	seer	provado	que	demando	mas
BNE17-10	si	depues		puhiere	seer	provado	que	demando	mas
Sev	si	depues		puhier	seer	provado	que	demando	mas
Vall	si	depues		puhier	seer	provado	que	demando	mas

Figura 23. Prueba 2 de la validez de la división de B2 en dos grupos.

A	QL	QM	QN	QO	QP	QQ	QR	QS	QT	QU
Grupo 1										
Z-II-9	sse	non	oviere	onde	faga	la	emienda	de	suso	dicha
CKB	si	non	oviere	onde	faga	la	emienda	de	suso	dicha
Z-III-6	si	non	oviere	donde	faga	la	emienda	de	suso	dicha
Grupo 2										
RAE54	si	non	oviere	donde	faga	el	emienda	de	suso	dicha
BNE17-10	si	non	oviere	onde	faga	el	emienda	de	suso	dicha
Sev	si	non	oviere	onde	faga	el	emienda	de	suso	dicha
Vall	si	non	oviere	donde	faga	el	emienda	de	suso	dicha

Figura 24. Prueba 3 de la validez de la división de B2 en dos grupos.

Estamos, por consiguiente, frente a un ejemplo perfecto de una acumulación de variantes que individualmente no tienen mucha relevancia estematológica, pero, puesto que las cuatro van en la misma dirección, constituyen un indicio de parentesco relativamente seguro.

Finalmente, tras exponer los lugares críticos que nos han permitido identificar ocho familias de manuscritos, en el esquema que sigue, aparece un resumen de las separaciones con las variantes que hemos utilizado:

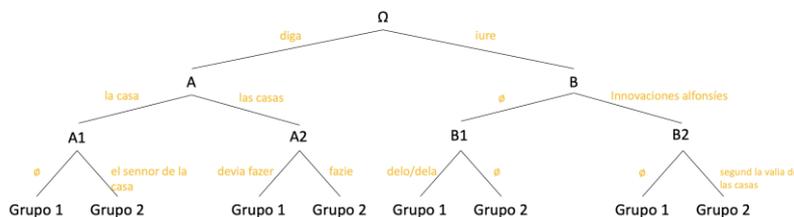


Figura 25. Estructura del *stemma codicum* del Fuero Juzgo.

No vamos a entrar en el análisis detallado de las diferentes familias y de las relaciones de parentesco entre los miembros de cada familia, pero a continuación aparece el esbozo provisional de *stemma codicum* que hemos podido trazar a partir del análisis de la ley 8.2.1²³. Hemos podido integrar un poco más del ochenta por ciento de los manuscritos, pero no disponíamos de indicios suficientemente sólidos para trazar el *stemma codicum* completo de la segunda familia de B1²⁴. Para completar, confirmar y refinar estos resultados, tendremos que explorar otras leyes y aportar argumentos de naturaleza variada.

²³ Para más detalles sobre el análisis de las relaciones entre los manuscritos de cada familia, *cfr.* Mabile 2022.

²⁴ Los manuscritos que no se han podido integrar en el *stemma* aparecen en gris, al lado de las familias a las que probablemente pertenecen.

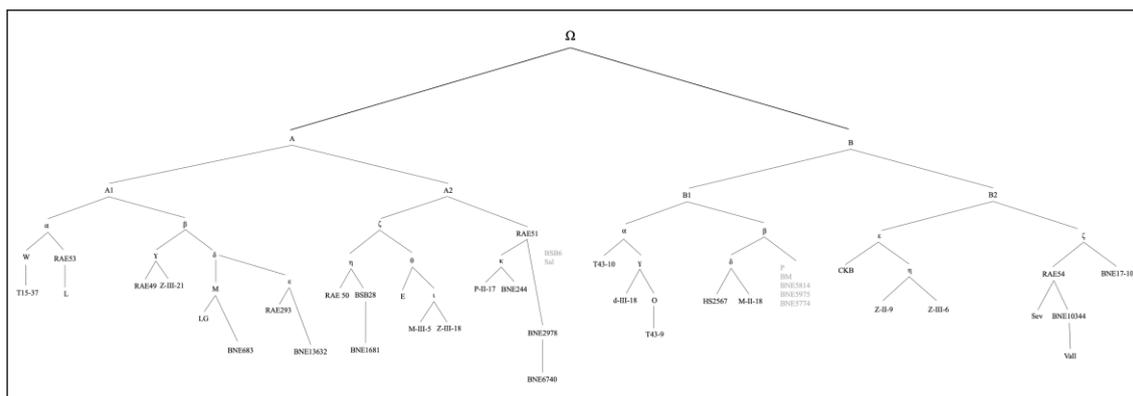


Figura 26. Esbozo de *stemma codicum* del Fuero Juzgo.

3. COMPARACIÓN ENTRE LOS RESULTADOS DE LOS DOS MÉTODOS

Si volvemos al árbol filogenético que hemos generado con el algoritmo Neighbor-Joining en la plataforma Stemmaweb (Figura 6), nos damos cuenta de que los resultados coinciden en parte con los resultados a los que hemos llegado con métodos de filología clásica. De hecho, son siete los grupos de manuscritos que el algoritmo había identificado como emparentados y que, efectivamente, pertenecen a la misma familia. Se trata de la primera familia de A1 –W, T15-37, RAE53 y L– que aparece en rojo (*cfr.* Figura 27 y 28); dos grupos de la segunda familia de A1 –M, LG y BNE683²⁵ (en morado), y RAE293 y BNE13632 (en naranja)²⁶–; una parte de la primera familia de A2 –BSB28 y BNE1681 (en azul oscuro)–; una parte de la segunda familia de A2 –RAE51, BNE2978 y BNE6740 (en azul claro)–; dos manuscritos de la primera familia de B1 –O y T43-9– (en amarillo); y, finalmente, la familia alfonsí, aunque aparece separada en dos grupos en el árbol filogenético (en verde).

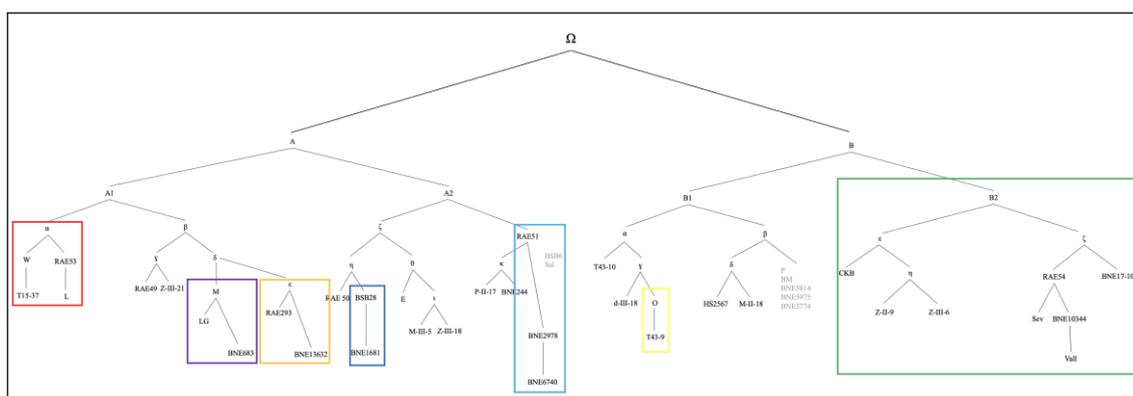


Figura 27. Esbozo de *stemma codicum* del Fuero Juzgo. Los colores indican los grupos de manuscritos que el algoritmo ya había identificado como emparentados (BSB6 y Sal, que aparecen en gris, no forman parte del grupo azul claro).

²⁵ Analizamos en detalle el libro VIII en este manuscrito en Castillo Lluch y Mabile 2021.

²⁶ Aunque el algoritmo haya identificado estos dos grupos de manuscritos, no los indica como emparentados entre ellos. En cambio, el análisis de las variantes nos revela que forman parte de la misma familia: A1(β).

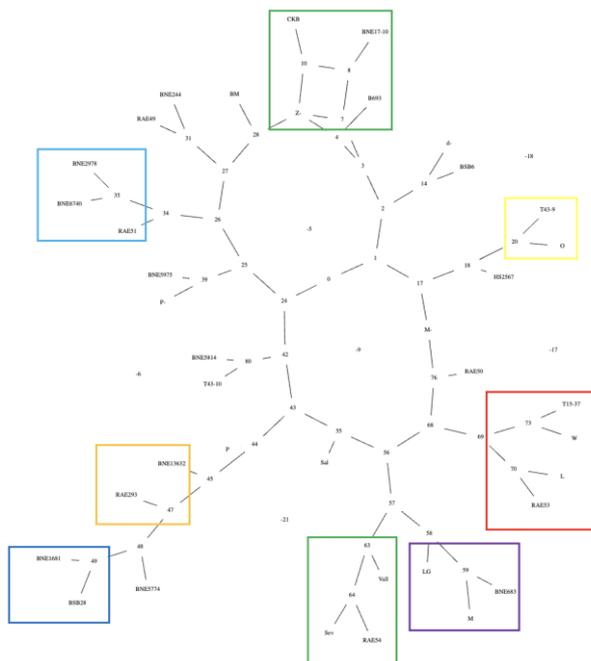


Figura 28. Árbol filogenético generado con el algoritmo Neighbor-Joining en la plataforma Stemmaweb. Los colores corresponden a los grupos de manuscritos que los métodos clásicos identificaron también como emparentados.

Observamos que, para los demás manuscritos, los resultados son diferentes. Por consiguiente, llegamos a la conclusión de que el algoritmo puede proporcionarnos una visión global de la distancia entre los testimonios de un mismo texto antes de empezar a explorar los lugares críticos que nos permiten identificar las diferentes familias, pero no nos ofrece resultados del todo fiables. De hecho, para los manuscritos no marcados en colores en el árbol filogenético, los resultados a los que llegamos con métodos clásicos no convergen y, por lo tanto, a pesar de las coincidencias mencionadas en el párrafo precedente, constatamos que hay también muchos fallos. Aunque no podemos comprobar si los errores están en el árbol, en nuestro esbozo de *stemma codicum* o incluso en ambos lugares antes de explorar otras leyes, nos parece más plausible que el problema provenga del algoritmo, puesto que hemos intentado ser muy sistemáticos en nuestro análisis de las diferentes familias. Nos parece entonces razonable afirmar que los algoritmos pueden ayudar al filólogo al inicio del trabajo, pero por el momento no pueden sustituirlo en el proceso de la *constitutio stemmatis*. A pesar de esto, los resultados del algoritmo son mucho más precisos de lo que imaginábamos al comenzar nuestra investigación. De hecho, cuando vimos el árbol filogenético por primera vez, no pensábamos que algunas de las familias habían sido identificadas correctamente, por lo que dejamos de lado esta herramienta. Fue solo en un segundo momento, después de haber trazado el esbozo de *stemma codicum* siguiendo el método tradicional, cuando nos dimos cuenta de que aparecían correspondencias. Por lo tanto, con algunas mejoras en el algoritmo y la integración del sistema de anotación de variantes según su relevancia estematológica, podríamos sin duda llegar a resultados más precisos.

4. CONCLUSIÓN

El objetivo principal de este trabajo era poner a prueba algunas de las herramientas de filología digital existentes y evaluar cómo puede ayudar la tecnología al filólogo en la tarea

interesantísima pero muy costosa, tanto en tiempo como en energía, que constituye la *collatio codicum*. Para contestar a esta pregunta, hemos recorrido el estado de la cuestión de la *collatio* asistida por ordenador intentando entender cómo funcionan los diferentes programas que se han desarrollado en los últimos años y en qué medida permiten ahorrar tiempo a la hora de alinear los diferentes testimonios, concentrándonos sobre todo en su accesibilidad y funcionalidad. En segundo lugar, hemos tratado de entender cómo el resultado de este alineamiento podía facilitar la tarea de la *examinatio* y de la *selectio* de los lugares críticos significativos que pueden servir al filólogo para estudiar las relaciones de parentesco entre los diferentes manuscritos. Sin embargo, los resultados de nuestras pruebas no son tan concluyentes como habíamos imaginado.

Al fin y al cabo, incluso si hemos aprendido a utilizar cuatro programas diferentes, el único que nos sirvió realmente para el análisis de las relaciones entre los testimonios del Fuero Juzgo es la versión Python de CollateX, que nos permitió generar un documento CSV que contiene los diferentes testimonios tokenizados y alineados (tabla de colación). Hemos corregido manualmente el alineamiento automático en Microsoft Excel y hemos añadido colores para poner en evidencia los errores, las innovaciones y las transposiciones, y hemos basado casi la totalidad de nuestro análisis en este alineamiento. Un programa como TRAViz, por ejemplo, resulta muy útil para visualizar el texto de los diferentes testimonios en una eventual edición numérica que pretenda mostrar todas las variantes simultáneamente, pero no sirve mucho para analizar las relaciones entre los diferentes testimonios de manera concreta con el fin de identificar las diferentes familias de manuscritos. Una plataforma como Stemmaweb podría ofrecer muchas ventajas al filólogo proporcionando árboles filogenéticos precisos que serían muy útiles para tener una idea global de la distancia entre los testimonios, pero esto requeriría solucionar los fallos de funcionamiento de la plataforma. Aunque hemos podido observar correspondencias entre los resultados obtenidos con el algoritmo y los obtenidos con métodos de filología clásica, el árbol filogenético podría mejorarse mucho, puesto que no tiene en cuenta las intervenciones manuales que el investigador añade con respecto al tipo de variantes, de modo que, con una tradición medieval que presenta mucha variación ortográfica y morfológica los resultados no son muy precisos. En cuanto a Juxta, incluso si dispone de una interfaz intuitiva, nos parece útil solamente cuando el filólogo quiere comparar dos testimonios a la vez. De hecho, el usuario no puede neutralizar las variantes no significativas y resulta complicado interpretar los resultados del alineamiento, pues, en una tradición que presenta mucha variación, el testimonio base aparece repleto de variación.

Existen muchas posibilidades y la informática puede ofrecer herramientas con un potencial enorme para mejorar no solo las condiciones de trabajo del investigador, sino también la calidad de los resultados obtenidos después de la *collatio codicum*. No obstante, estas herramientas son muy complejas y necesitan una formación previa para que el investigador pueda aprovechar al cien por ciento lo que pueden aportar. La inversión en tiempo y en energía que un principiante debe proporcionar para llegar a resultados plenamente satisfactorios es considerable. En definitiva, para un filólogo que no dispone de una formación previa, el tiempo ahorrado con el alineamiento automático se ve aniquilado por el tiempo necesario para aprender a utilizar los programas. Desgraciadamente, todavía no existe un programa fácil de uso que agrupe las ventajas y las funcionalidades de los diferentes programas existentes en el que el usuario pueda realizar la *collatio* de manera intuitiva. Para que la *collatio* asistida por ordenador sea

accesible a los filólogos, necesitaríamos una interfaz más amigable que permita intervenir en las diferentes etapas de la colación y adaptarlas a su caso de estudio concreto, sin que sea necesario entrar en el código y aprender lenguajes informáticos relativamente complejos para un principiante.

En cuanto a los métodos de filología clásica, hemos realizado la *selectio* y la *examinatio* de los lugares críticos más significativos apoyándonos en el alineamiento generado con CollateX. De hecho, nos parecía la manera cognitivamente más intuitiva de localizar los lugares críticos, añadiendo colores manualmente para clasificar los diferentes tipos de variantes y organizarlas jerárquicamente según su relevancia estematológica. Este método nos ha permitido alcanzar la meta principal de nuestra investigación: estudiar las relaciones de parentesco existentes entre las diferentes familias de manuscritos para llevar a cabo la *constitutio stemmatis* del texto del Fuero Juzgo y proponer un esbozo de *stemma codicum*. En primer lugar, a partir de una variante sustantiva muy significativa, hemos separado el conjunto de manuscritos en dos ramas. En segundo lugar, a partir de otros lugares críticos significativos, hemos separado cada una de las ramas en dos subramas, divididas sucesivamente en familias de manuscritos. Para cada una de las familias identificadas gracias a las diferentes separaciones, hemos puesto en práctica los métodos de filología clásica para identificar, en la medida de lo posible, las relaciones de parentesco entre los manuscritos.

A partir de la ventana textual que hemos abierto, hemos logrado trazar un esbozo de *stemma codicum* de más del ochenta por ciento de la tradición manuscrita del Fuero Juzgo con argumentos relativamente sólidos. Será necesario, en el futuro, abrir otras ventanas sobre este texto para, por un lado, entender las relaciones entre los manuscritos que no hemos podido colocar en el *stemma* y, por otro, confirmar y refinar los resultados que hemos obtenido²⁷. Además, los indicios que hemos considerado para estudiar las relaciones de parentesco entre los diferentes manuscritos son casi exclusivamente textuales, pero el filólogo que pretende construir el *stemma codicum* de una tradición debe basar sus conclusiones en argumentos de variada naturaleza, tanto (con)textuales como lingüísticos y materiales, aspectos todos ellos previstos en la próxima etapa de nuestra investigación.

BIBLIOGRAFÍA

Fuentes bibliográficas

- [RAE49] Madrid, Real Academia Española, ms. 49.
- [RAE50] Madrid, Real Academia Española, ms. 50.
- [RAE51] Madrid, Real Academia Española, ms. 51.
- [RAE53] Madrid, Real Academia Española, ms. 53.
- [RAE54] Madrid, Real Academia Española, ms. 54.
- [RAE293] Madrid, Real Academia Española, ms. 293.
- [BNE17-10] Madrid, Biblioteca Nacional de España, ms. Vitr. 17-10.
- [BNE244] Madrid, Biblioteca Nacional de España, ms. 244.

²⁷ En un trabajo reciente (Castillo Lluch y Mabile 2023, en prensa), hemos cotejado otras cuatro leyes (1.1.2, 3.5.7, 8.4.1 y 11.1.2) que han permitido validar más del 50 % del esbozo de *stemma*. Serán necesarios otros cotejos para observar si las relaciones restantes se confirman o si el *stemma* requiere algunos ajustes.

- [BNE683] Madrid, Biblioteca Nacional de España, ms. 683.
 [BNE1681] Madrid, Biblioteca Nacional de España, ms. 1681.
 [BNE2978] Madrid, Biblioteca Nacional de España, ms. 2978.
 [BNE5774] Madrid, Biblioteca Nacional de España, ms. 5774.
 [BNE5814] Madrid, Biblioteca Nacional de España, ms. 5814.
 [BNE5975] Madrid, Biblioteca Nacional de España, ms. 5975.
 [BNE6740] Madrid, Biblioteca Nacional de España, ms. 6740.
 [BNE10344] Madrid, Biblioteca Nacional de España, ms. 10344.
 [BNE13632] Madrid, Biblioteca Nacional de España, ms. 13632.
 [d-III-18] San Lorenzo de El Escorial, Real Biblioteca del Monasterio, ms. d-III-18.
 [M-III-5] San Lorenzo de El Escorial, Real Biblioteca del Monasterio, ms. M-III-5.
 [M-II-18] San Lorenzo de El Escorial, Real Biblioteca del Monasterio, ms. M-II-18.
 [P-II-17] San Lorenzo de El Escorial, Real Biblioteca del Monasterio, ms. P-II-17.
 [Z-II-9] San Lorenzo de El Escorial, Real Biblioteca del Monasterio, ms. Z-II-9.
 [Z-III-6] San Lorenzo de El Escorial, Real Biblioteca del Monasterio, ms. Z-III-6.
 [Z-III-18] San Lorenzo de El Escorial, Real Biblioteca del Monasterio, ms. Z-III-18.
 [Z-III-21] San Lorenzo de El Escorial, Real Biblioteca del Monasterio, ms. Z-III-21.
 [T15-37] Toledo, Archivo y Biblioteca Capitulares, ms. 15-37.
 [T43-9] Toledo, Archivo y Biblioteca Capitulares, ms. 43-9.
 [T43-10] Toledo, Archivo y Biblioteca Capitulares, ms. 43-10.
 [M] Murcia, Archivo Municipal, Serie 3, ms. 53.
 [LG] Madrid, Fundación Lázaro Galdiano, Registro 14423, ms. 286.
 [BM] Mallorca, Fundación Bartolomé March, ms. 20/5/4.
 [Sal] Salamanca, Biblioteca Universitaria, ms. M 35.
 [San] Santander, Biblioteca Menéndez Pelayo, ms. M-557.
 [Sev] Sevilla, Biblioteca Provincial de la Universidad, ms. A 331/155.
 [Val] Valladolid, Biblioteca Histórica de Santa Cruz, ms. 005-006.
 [RAH] Madrid, Real Academia de la Historia, ms. 9/1783.
 [L] Lisboa, Biblioteca Nacional de Portugal, ms. IL 111.
 [HS2567] New-York, Hispanic Society of America, ms. B 2567.
 [P] Paris, Bibliothèque Nationale de France, ms. Esp. 256.
 [BSB6] München, Bayerische Staatsbibliothek, ms. Cod. Hisp. 6.
 [BSB28] München, Bayerische Staatsbibliothek, ms. Cod. Hisp. 28.
 [W] Washington, Library of Congress Law, ms. F8.
 [O] Oxford, Bodleian Library, ms. Holkham misc. 46.
 [E] Stockholm, Kungliga Biblioteket, ms. B 693.
 [CKB] København, Det Kongelige Bibliotek, ms. G.K.S. 1942.
 [CAI] København, Arnamagnæan Institute, ms. AM806.
 [LF] Santiago de Compostela, Museo do Pobo galego, fondo Blanco Cicerón, ms. RBC/P5-03//03
(Fragmento).
 [HS2713] New-York, Hispanic Society of America, ms. B2713 *(Fragmento)*.
 [ARCV1] Valladolid, Archivo de la Real Chancillería, Pergaminos, carpeta 116, 8 *(Fragmento)*.
 [ARCV2] Valladolid, Archivo de la Real Chancillería, Pergaminos, carpeta 118, 2 *(Fragmento)*.

Referencias bibliográficas

- ANDREWS, Tara L. (2016): «Analysis of Variation Significance in Artificial Traditions Using Stemmaweb», *Digital Scholarship in the Humanities*, 31, 3, pp 523-539.

- ANDREWS, Tara L. y van Zundert, Joris (2013): «An Interactive Interface for Text Variant Graph Models», *Digital Humanities Conference DH2013*, University of Nebraska-Lincoln, 16-19 de Julio de 2013.
- BERMÚDEZ SABEL, Helena (2016): «Colación asistida por ordenador: estadio de la cuestión y retos», *Revista de Humanidades Digitales*, 1, pp. 20-34.
- BIRNBAUM, David J. y Elena SPADINI (2020): «Reassessing the Locus of Normalization in Machine-Assisted Collation», *Digital Humanities Quarterly*, 14, 3, 21 p.
- BLECUA, Alberto (1983): *Manual de crítica textual*. Madrid: Castalia.
- BLEEKER, Elli (2017): *Mapping Invention in Writing: Digital Infrastructure and the Role of the Genetic Editor*. Antwerp: Antwerp University. Tesis doctoral dirigida por Dirk Van Hulle.
- CASTILLO LLUCH, Mónica (2011): *Tel fils, tel père: Ferdinand III dans le processus de planification du castillan (étude linguistique du Fuero Juzgo)*, estudio inédito para obtener la Habilitation à diriger des recherches, presentado en la Universidad París-Sorbona, Paris IV el 26/11/2021. https://serval.unil.ch/resource/serval:BIB_5E3C78DF48FB.P001/REF
- CASTILLO LLUCH, Mónica (2012): «Las lenguas del *Fuero Juzgo*: avatares históricos e historiográficos de las versiones romances de la Ley visigótica (I)», *e-Spania* [en línea]. <http://journals.openedition.org/e-spania/20994>
- CASTILLO LLUCH, Mónica (2016): «Las fechas del *Fuero juzgo*: avatares históricos e historiográficos de la versión romance de la ley visigótica (II)», en Antonio Narbona Jiménez y Araceli López Serena (coords.), *El español a través del tiempo. Estudios de lingüística histórica ofrecidos a Rafael Cano Aguilar*. Sevilla: Editorial de la Universidad de Sevilla, pp. 47-68.
- CASTILLO LLUCH, Mónica y Charles MABILLE (2021): «El *Fuero Juzgo* en el ms. BNE 683 (1755) de Andrés Marcos Burriel», *Scriptum Digital*, 10, pp. 75-107.
- CASTILLO LLUCH, Mónica y Charles MABILLE (2023): «Hacia un esbozo de *stemma codicum* del *Fuero Juzgo* desde el Humanismo hasta hoy», *Actas del XII Congreso de Historia de la Lengua Española (16-20 de mayo de 2022)*, León: Universidad de León, en prensa.
- DEKKER, Ronald H. (2014): «Computer-Supported Collation of Modern Manuscripts: CollateX and the Beckett Digital Manuscript Project», *Digital Scholarship in the Humanities*, 30, 3, pp. 452-470.
- DEKKER, Ronald H. y Gregor MIDDELL (2011): «Computer-Supported Collation with CollateX: Managing Textual Variance in an Environment with Varying Requirements», *Supporting Digital Humanities 2011*. University of Copenhagen, Denmark, 17-18 November 2011.
- FERNÁNDEZ, Tomás (2019): «Lectiones adiaphorae, lectiones equipolentes y construcción en espiral de stemmata», *CIRCE de clásicos y modernos*, 23, 2, pp. 121-128.
- FERNÁNDEZ-ORDÓÑEZ, Inés (2002): «Tras la *collatio* o cómo establecer correctamente el error textual», *La corónica: A Journal of Medieval Hispanic Languages, Literatures and Cultures*, 30, 2, pp. 105-180.
- FERNÁNDEZ-ORDÓÑEZ, Inés (2019): «Las variantes de lengua: un concepto tan necesario como necesitado de formalización», en Enrico Malato y Andrea Mazzucchi (eds), *La critica del testo. Problemi di método ed esperienze di lavoro. Trent'anni dopo in vista del settecentenario della morte di Dante*. Roma: Salerno Editrice, pp. 375-403.
- HOWE, Christopher J. et al. (2012): «Responding to Criticisms of Phylogenetic Methods in Stemmatology», *Studies in English Literature. 1500-1900*, 52, 1, pp. 51-67.
- JÄNICKE, Stefan et al. (2015): «TRAVIz: A Visualization for Variant Graphs», *Digital Scholarship in the Humanities*, 30, 1, pp. 83-99.
- MABILLE, Charles (2022): *La tecnología al servicio de la collatio codicum: el caso del Fuero Juzgo*, Mémoire de Maîtrise universitaire ès lettres sous la direction de Prof. Dr. Mónica Castillo Lluch, Université de Lausanne, Faculté des Lettres. https://serval.unil.ch/fr/notice/serval:BIB_S_33415
- NURY, Elisa (2017): «Visualizing Collation Results», en Peter Boot, Anna Cappellotto, Wout Dillen, Franz Fischer, Kelly Aodhán, Andreas Mertgens, Anna-Maria Sichani, Elena Spadini y Dirk

- van Hulle (eds), *Advances in Digital Scholarly Editing. Papers Presented at the DiXiT Conferences in the Hague, Cologne, and Antwerp*. Leiden: Sidestone Press, pp. 317-321.
- NURY, Elisa (2018): *Automated Collation and Digital Editions: From Theory to Practice*. Londres: King's College. Tesis doctoral dirigida por Elena Pierazzo y Victoria Moul.
- NURY, Elisa y Elena SPADINI (2020): «From Giant Despair to a New Heaven: the Early Years of Automatic Collation», *Information Technology*, 62, 2, p. 61-73.
- PÉREZ MARTÍN, Antonio (2002): «El Fuero Juzgo, código de leyes del Reino de Murcia», en José Perona (ed.), *El Fuero Juzgo. Estudios críticos y transcripción*. Murcia: Consejería de Educación y Cultura de la Región de Murcia y Fundación Séneca, p. 41-74.
- QUENTIN, Henri (1926): *Essais de critique textuelle (ecdotique)*. Paris: A. Picard.
- ROBINSON, Peter M. W. (1989): «The Collation and Textual Criticism of Icelandic Manuscripts (1): Collation», *Literary and Linguistic Computing*, 4, 2, pp. 99-105.
- ROELLI, Philipp y Dieter BACHMANN (2010): «Towards Generating a Stemma of Complicated Manuscript Traditions: Petrus Alfonsi's Dialogus», *Revue d'histoire des Textes*, n.s., 5, 307–321.
- SPADINI, Elena (2017): «The Role of the Base Manuscript in the Collation of Medieval Texts», en Peter Boot, Anna Cappellotto, Wout Dillen, Franz Fischer, Kelly Aodhán, Andreas Mertgens, Anna-Maria Sichani, Elena Spadini y Dirk van Hulle (eds), *Advances in Digital Scholarly Editing. Papers Presented at the DiXiT Conferences in the Hague, Cologne, and Antwerp*. Leiden: Sidestone Press, pp. 345–49
- SPENCER, Matthew y Christopher J. HOWE (2004): «Collating Texts Using Progressive Multiple Alignment», *Computer and the Humanities*, 38, pp. 253-270.
- ZEUMER, Karl (1902), *Leges Visigothorum*, in *Monumenta Germaniae historica*, I. Hannover – Leipzig: Hahn.