

Transliteration of the Voynich MS text

René Zandbergen

Independent researcher, Lampang, Thailand

Abstract

The mysterious writing in the Voynich MS has resisted all attempts to translation or decryption, in spite of the increasing number of attempts over the last decades. What has become clear is that the text has a number of unusual properties, and increasingly advanced text and language analysis techniques have been deployed in order to get to the bottom of this mystery. The basis for all these attempts is the existence of a computer-readable version of the text – a transcription, or rather a transliteration as we don't even know the alphabet of this writing. This paper deals with a number of aspects of the transliteration of the Voynich MS text.

First, a brief historic overview introduces the various efforts and the existing transliterations resulting from them. This also serves to introduce the specific problems that exist with this text. The transliterations use different conventions for the representation of the text (usually called 'alphabets'), but their accuracy is largely unknown, and the assumptions that were made in creating these transliterations were likely sub-optimal or even incorrect. This limits or even biases all computer analyses based on them. This is also true for the most popular alphabet: "Eva", which was defined by Gabriel Landini and the author of this paper, just before the year 2000.

Nowadays, there is a multitude of users of these transliteration files, and these users are often unaware of these risks. Many prefer to use whatever transliteration file is most easily available or most easily usable. More accurate transliterations exist, but these are often considered overly complicated to use.

As a next step, this paper introduces a superset of all existing transliteration alphabets, which allows all transliterations to be represented in the same 'super alphabet'. As a result, the completeness and consistency of all existing transliterations can be verified in detail. Some key statistics about the accuracy and completeness of all existing transliterations are presented.

This 'super alphabet' also allows an easy transformation from any existing transliteration into any of the existing alphabets. Furthermore, it allows a user to easily create his own character set definition and convert any existing transliteration into it. With this, it is now possible to repeat computerised attacks on the text quickly and easily for any number of 'alphabets', allowing the analyst to tune this alphabet as he progresses. An example of such an analysis is presented.

Keywords

Text transliteration, manipulation, analysis

1. Introduction

The Voynich MS is an early 15th century manuscript written in an unknown writing system. An example of this writing is shown in Figure 1. The manuscript has text that is mostly organised in paragraphs of running text (as in Figure 1), but also includes text that is not arranged in normal paragraphs. It has numerous diagrams where text has been integrated into the drawing, for example along circumferences of circles, or along radii of such circles. In the following, we will refer to these as 'circular text' and 'radial text' respectively. The MS also includes many hundreds of stand-alone

International Conference on the Voynich Manuscript 2022, November 30--December 1, 2022, University of Malta.

EMAIL: rene.zandbergen.rz@gmail.com

ORCID: 0000-0003-3963-3713



© 2022 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

words, often near drawing elements. In the following we will refer to these as ‘labels’. See Figure 2 for an example of all three types of non-standard writing.

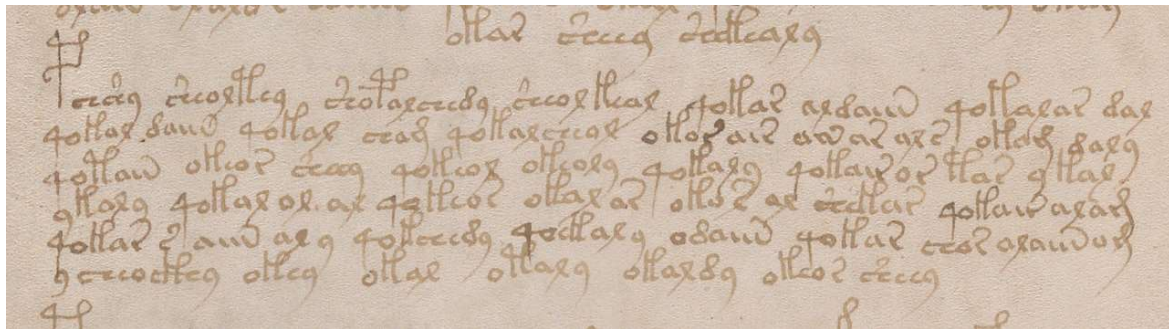


Figure 1: Voynich MS writing laid out in paragraphs



Figure 2: Non-standard Voynich MS writing: circular and radial text, and labels

The process of rendering a hand-written English text into computer-readable form is called transcription. In this case we know the alphabet and we know the meaning of each character. If we have a text written in another writing system, for example Arabic, and we convert this into the Latin alphabet, the process is called transliteration. With the Voynich MS, we can observe the various character shapes in the hand-written text, but we do not know what they mean. To render this text into computer-readable form is again a form of transliteration.

The purpose of this transliteration is to be able to make computer analyses of the text, which may range from the computation of simple statistics to complex NLP analyses. For the Voynich MS this transliteration is made difficult by several aspects:

- This is not a known alphabet. Some characters look familiar, but others don't. We do not know what the words mean, so we have no help in identifying characters from context.
- The text is hand-written, and there is a significant amount of variation in the character shapes. It is often difficult to decide if two similar, yet different shapes are different characters or just handwriting variations.
- Some characters look like composites of other characters, and one cannot be certain if these are new characters or ligatures of existing characters.
- While the text appears to consist of words separated by word spaces, in many areas the spacing is uncertain.

During the last half century, different transliteration efforts have been made, and in each case the analysts have made different assumptions on the last three points. As a result of all these uncertainties, simple questions like: “how many characters are there in the text” or “how many different characters are there in the Voynich alphabet” cannot be answered. Furthermore, in several cases, the people or teams creating these transliteration systems had their own purpose for the resulting texts and were not primarily considering more general user needs. A short overview of past transliteration efforts may

illustrate all these points.² This will also bring up the most important specific complications that the Voynich MS presents.

2. Overview of historical transliteration efforts

We may limit ourselves to significant past efforts to convert the Voynich MS text into a computer-readable form. The first of these was an effort by a team of cryptanalysts under the leadership of William F. Friedman in the 1940's, the so-called First Study Group, or FSG in the following. This activity is described in detail by Jim Reeds [1]. While the FSG managed to cover almost the entire manuscript, they decided to include only text that is arranged in normal paragraphs, and not to include the circular and radial texts, or labels. They used capital letters and numbers for the representation of the Voynich MS characters. The full alphabet definition is presented at the author's web site [2].

One of the decisions made by this group concerned a specific set of composite characters in the text. These characters still present an unresolved issue in the understanding of the script, which we need to look at in some detail, and which we may call the 'first complication' of the Voynich script. Referring to Figure 3, the character 3a, which is very frequent in the MS, was transliterated by the FSG as "T". The characters 3b, also quite frequent, were transliterated as "H" and "D" respectively. Over time, the latter characters have been nicknamed gallows characters. However, the MS text also includes the two frequent characters shown in 3c, in which the two forms are superimposed. It is not clear whether these represent new single characters, or some combination or ligature of the other two. The solution adopted by FSG was to introduce the notation "HZ" and "DZ" respectively, where the Z is only used in these combinations, and can never stand alone. These characters have been nicknamed intruding gallows or pedestalled gallows.

Due to the self-imposed secrecy of this group, neither their definition of the transliteration alphabet, nor the resulting transliterated text, were known to the world at large, until they were re-discovered in the archives of the Marshall Foundation, in the 1990's, for which see [1].

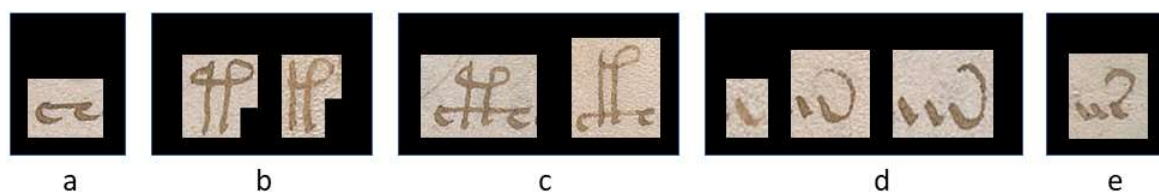


Figure 3: a-e: examples of characters in the Voynich MS referred to in Section 2.

The next major transliteration effort came in the 1970's, from Prescott Currier with contributions from Mary D'Imperio [3]. They did not have access to the FSG results and appear to have been unaware of its existence. However, both the alphabet defined by Currier and the resulting transliteration file were shared electronically and became well known in the gradually increasing Voynich MS community. Currier, like FSG, used capital letters and numbers for the representation of the Voynich MS characters, but he adopted a different approach for the 'first complication', in that he assigned to each of the four most frequent pedestalled gallows its own single character. Another important difference in Currier's character assignment concerns what we may call the 'second complication', namely the 'strings of i'.

Referring to Figure 3, the characters in 3d were transliterated as 'I', 'N' and 'M' both by FSG and Currier. It is not surprising that they came up with the same assignments, because that is what these shapes look like. However, the final minim of the N and M can also appear by itself, and here FSG and Currier assigned different characters. Furthermore, the i-shaped minim, that can appear up to three times in a row, can also be followed by other shapes, as shown in Figure 3e. For these, Currier decided to assign individual characters to all combinations, while FSG decided to 'write them out'. Both sets of transliteration rules allow for some ambiguity in representing these characters, a topic we will need to revisit later. Beside the i-shape, the only other shape that can appear up to 4 times in a row in the

² This is not meant to be a complete historical overview. The interested reader may refer to [2] for more information.

Voynich MS text is a c-shaped character, which both FSG and Currier transliterated as ‘C’. There has never been any serious consideration that this shape could also be just a minim.

The Currier transliteration was not nearly as complete as the FSG transliteration. Currier probably started his work primarily with his own use of the result in mind, but it was shared freely with all interested users, and it enjoyed great popularity in the new internet community. With so many eyes looking at the MS text, it soon became clear that there was a third complication, namely that there were a lot of other characters in the MS that were not covered by the FSG or Currier alphabets. These tended to be relatively rare, and many of them looked like ligatures of existing characters. A first inventory was prepared by Jim Reeds, who numbered them.³ In addition to ligatures, also some new shapes were identified. However, neither the FSG system nor the Currier system foresaw any way of including these additional characters in the transliteration files. For a period of time, these characters were popularly known by the Voynich community as ‘weirdoes’.

In the late 1990’s Gabriel Landini proposed to prepare a new transliteration, based on the black-and-white printed copies of a Yale microfilm and of a hand transcription made by Theodore Petersen [4]. He was immediately joined by the author of this paper. It was foreseen from the beginning that this should be a complete general-purpose transliteration, intended for many users. It was decided to properly address the three complications listed above. Of course, it remained true that there is no way of knowing whether the pedestalled gallows, the ‘strings of i’, and the weirdo ligatures represent single or multiple characters. The new alphabet simply aimed at representing in electronic form what one sees in the MS, without forcing any decision on these questions.

It was also decided to introduce a more standard file format. This format turned out to be a prototype that had to be revised in some details many years later, as will be addressed further below. To address the three complications described above, a solution was implemented that was strongly inspired by the work of Jacques Guy, who had devised an ‘analytical’ transliteration alphabet, that did not intend to represent complete characters, but rather break them down into smaller, common elements.⁴ The new alphabet, initially called ‘European Voynich Alphabet’, or Eva, consisted of lower case characters only, to distinguish it from the FSG and Currier alphabets. At the same time, the following conventions were introduced:

- Indicate ligatures of characters by surrounding them by brackets: { }
- Indicate uncertain readings of characters by listing the options between square brackets: []
- Use a numbering notation for ‘weirdoes’ as they were encountered during the transliteration effort, as: \$nnn;

Importantly, based on a suggestion from Gabriel Landini, the Voynich MS character shapes were assigned to the Latin alphabet in such a way, that the majority of the text would become more or less pronounceable. This made it very easy to learn the alphabet, and to remember transliterations of Voynich MS words. Each of us transliterated the entire manuscript individually in a first round, after which we started the complicated second round of aligning the two results to each other. At the end of the first round, well over 100 different ‘weirdoes’ had been identified, and they could be subdivided into three categories:

1. New, individual symbols (all rare)
2. Ligatures of standard symbols
3. Ligatures involving new symbols (all rare)

There was no need to maintain the above-mentioned \$nnn; notation, but instead, the new symbols, either whole or part of a ligature, received a new character code in the high-Ascii area. These could be entered in the file as high-Ascii, but alternatively as text: @nnn; in order to keep the file in plain Ascii, and not create issues during file transfers, or with tools interpreting Unicode. As a result, there was a ‘Basic Eva’ character set, consisting of lower-case characters, and an ‘Extended Eva’ character set, including all the ‘weirdo’ components. The name of the alphabet was changed into: ‘Extensible Voynich Alphabet’. Basic Eva had been defined prior to the transliteration, based on the characters of the FSG and Currier alphabet, plus three more characters, which appeared sufficiently frequently in the text. Both the FSG and Currier transliterations could be converted to Eva without loss of information. While the two transliterations were completed, and the Eva definitions including all the high-Ascii

³ See [2], Table 2.

⁴ See [2], under “Froggy”.

codes published on the internet, the alignment of the two transliterations into one was never completed. Fortunately, in parallel, the Japanese Takeshi Takahashi had started his own complete transliteration of the text, using the basic Eva alphabet, and published this towards the end of the 1990's [5]. It exists in several versions, and for the purpose of this paper the version included in the Landini-Stolfi interlinear file will be used.⁵

Due to its simplicity, the (basic) Eva alphabet acquired an enormous popularity, and completely replaced the Currier alphabet that had been in use in the community until then. Equally, the Takahashi transliteration became the standard source for most text analyses. While the Eva alphabet could represent the FSG and Currier transliterations, not all transliterations made in Eva could be converted to FSG or Currier, because the Eva alphabet can model many additional character shapes in the MS.⁶

A final complete transliteration of the Voynich MS that has been made publicly available was created in the mid-2000's by Glen Claston.⁷ He had the significant advantage of having the high-resolution digital scans from 2004 available as a source for his transliteration. His purpose was clearly stated by himself: he wished to use it to continue the research of Leonell Strong on the Voynich MS. For this purpose, he needed to know as accurately as possible which are the characters making up the alphabet used in the manuscript, and he assigned his character definitions accordingly. Like Landini and myself, Claston collected the list of symbols as he transliterated the entire manuscript and used the full (usable) Ascii range to represent them. He named the final version of the alphabet v101.

This transliteration should be considered quite accurate, since it was based on high-quality input images, and Claston had made a great effort in identifying different character shapes. However, it found only moderate use in the Voynich community. This may be partly due to its complexity, with approximately 200 different characters, which in some cases only differ in minor details.

In 2017, I decided to make my transliteration, created during the collaborative effort with Landini, available to the community. Also this transliteration presents obstacles to users, because, like the v101 transliteration, it recognises approximately 200 different characters. All this means that, as from 2017, five independent transliterations of the Voynich MS text are publicly available, all with very different properties, using different character definitions, and in different file formats. They are summarized in Table 1 below.

Table 1

List of publicly available transliterations, and the codes used for them in this paper

Originator	Code	Alphabet
Friedman's First Study Group	FG	FSG
Prescott Currier (with help from Mary D'Imperio)	CD	Currier
Takeshi Takahashi	IT	Basic Eva
Glen Claston	GC	v101
René Zandbergen (with Gabriel Landini)	ZL	Extended Eva

3. Voynich MS text analysis as a project

Professionally, I have been involved in projects that include international collaboration based on processing of significant amounts of globally available data. From this perspective, the situation described above is highly unsatisfactory. Even though in these professional projects it is common that groups are working individually, there is an absolute need for the use of common conventions and standards. This allows exchange of data, exchange of results and cross-verification. It allows group or person A to use the results of group or person B, to compare, and to build on it. Only in this way, collaboration can lead to progress. Such standards are usually based on user requirements. For Voynich MS analysis there may be different types of users: some just want to look at the files and analyse

⁵ See [2], Table 12.

⁶ For some examples, see [2], Table 5.

⁷ Understood to be a pseudonym. No original publication by Claston is available. For more details, see [2] under "v101".

visually. Others want to run some quick experiments. A third group may want to do extensive numerical processing. Some users need completeness, others need accuracy. Such a requirements analysis has never been performed.

In present-day Voynich MS text analyses, most users will use the Takahashi transliteration, because it is readily available and reasonably easy to understand and use. However, it is available in several different places in different versions. As a result, two persons doing the same analysis are likely to end up with different results, without a chance of finding out where the difference comes from. Many published results, especially informal ones, do not even state which data was used as input, so they are not reproducible. The present paper describes two steps that have been made to improve on this situation. The first of these steps has already been published at the author's web site, and its results are fortunately already being used by several individuals and groups. The second step is first published here. To achieve these steps, I have not made a formal user requirements analysis, but I have considered the typical use cases listed above, and the many years of experience in running text analyses and interacting with other researchers. It is finally modelled on the principles I have encountered in my professional environment, namely:

- Input data (raw data) may be inhomogeneous
- These should be converted into standard formats that can be used by all
- There should be some standard tools that can be used by everyone to perform some of the most frequent manipulations and conversions, using these standard formats
- People/groups will tend to have their own application software, but these all use the same input and output formats

In short, there is a need for a standard format for all transliteration files, and some standard tools. As mentioned above, a first attempt for a standard format was made during the Eva transliteration, but this turned out not to be able to incorporate the v101 transliteration, due to clashes between the notation used by the format and the symbols included in the v101 transliteration alphabet.

An important point is the use of metadata. This does not refer to the transliterated text itself, but to information about this text. The first thing to know about a piece of transliterated text is where it appears in the manuscript. This location consists of two parts: the page or folio on which the text appears, and an indication where it appears on this page or folio. For the combination of the two, I will use the term *locus*. For the foliation or page naming, there is already a convention, so nothing needs to be invented here. For the indication of the location on the page, one can imagine that the coordinates of the start point of the text in some coordinate system would be most appropriate, but this information is not yet available. In all historical files, the *loci* have been numbered, starting from one on each page. It is also useful to indicate whether any piece of text is a normal line of flowing text, is a circular or radial text, or is a label.

On the one hand, the file should include metadata, but on the other hand, when performing text analyses, this information should be skipped, or previously removed from the file. Also, it will be very useful to be able to select parts of the text, based on the location (which page?) or the type of locus (for example: labels only) or other properties of the text that can be included as metadata. It is clear that a standard pre-processing tool is required, and the format of the file and the tool need to be defined in parallel.

3.1. Standard transliteration file format and standard pre-processing tool

The transliteration file format has been called "Intermediate Voynich Transliteration File Format" (IVTFF) [6]. The pre-processing tool is called "Intermediate Voynich Transliteration Tool" (ivtt), and it can be obtained from the author's web site.⁸ Some of the most important file formatting features, and the corresponding pre-processing tool options are summarised in Table 2 below. Note that the user guide ([7]) includes a great number of additional options. With respect to the word "Intermediate" that is part of both names, the reason for this is that this format should not be considered the final solution. The use of a single Ascii text file to include all data and metadata is strongly based on legacy conventions, dating back to the 1940's and the later use of punch cards. A final solution should use a

⁸ See: http://www.voynich.nu/extra/sp_transcr.html#ivtt

more modern approach, for example a relational database. Still, having all data in a single, consistent (intermediate) file format, will make the transition to such a solution much easier, and the data extracted from such a database, when used for text processing, can again use the intermediate format, assuming that by that time there are tools that are already able to read this format.

The five main transliterations described in this paper have all been converted into this format and can now be processed by the same tools, in particular by `ivttt`. All files can be accessed through the author's web site [8]. The difference between the original representation of these files, and the conversion to the standard format is illustrated in Figure 5 further below.

Table 2

Overview of some of the most important file format and tool options

File format feature	Pre-processing tool option(s)
Include the <i>locus</i> of each item	Optionally remove this information
Allow generic comments for human reading	Optionally remove these comments
Provide properties of each page, e.g.: Quire nr, type of illustration, Currier language	- Allow selection based on these properties - Optionally remove this information
Indicate the type of text for each <i>locus</i>	Allow selection of <i>loci</i> by their type
Indicate illegible characters	Allow deletion of words including these
Indicate alternatives for uncertain readings	Allow selection or convert to 'illegible'
Indicate uncertain word spaces	Optionally convert to space or 'no space'

3.2. Towards a unified transliteration alphabet

The most accurate transliterations that are publicly available (ZL and GC) use very different alphabets, both composed of well over 200 characters, and work along different principles. Due to their large character sets, and the rather inconvenient (if not outdated) use of high-Ascii codes, they are not very popular among Voynich MS researchers, who mostly prefer to use the IT file. The use of an Eva transliteration is, however, sub-optimal for most statistical analyses, due to its analytical nature. It means that symbols in the MS that are most likely meant to be individual characters may be described in Eva by groups of 2-4 symbols. The FSG, Currier and v101 alphabets each imply some decisions about which are the individual characters in the text, but these decisions are different and may all be wrong.

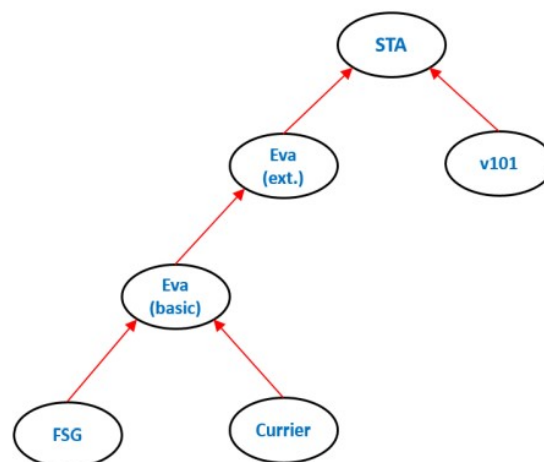


Figure 4: relation between transliteration alphabets mentioned in this paper. Arrows point up to alphabets that fully include the lower alphabet.

As mentioned above, the design of the Eva alphabet was such that it was a superset both of FSG and Currier, in such a way that transliterations in FSG or Currier could be converted to Eva without loss of information. Extended Eva adds numerous rare characters to this character set, so it is a superset of basic Eva, and therefore also of FSG and Currier. v101, on the other hand is separate from this, but both extended Eva and v101 were defined in the course of a complete transliteration of the MS, and thereby cover the same ground. It is possible in most cases to find matches between the two, and it is also possible to create a superset, that includes all characters of Eva and v101. A superset of extended Eva and v101, which will be referred to as ‘Super Transliteration Alphabet’ or ‘STA’ in the following, then also becomes a superset of FSG, Currier and Basic Eva. Once such an alphabet has been defined, it becomes possible to express all existing transliterations in the same alphabet, and to compare them directly. This work was started in the course of 2019 and completed in April 2020. The diagram in Figure 4 shows the principle.

4. Definition of a Super Transliteration Alphabet

An important caveat needs to be expressed first. The purpose of this exercise was not to define the ultimate new transliteration alphabet that shall replace all previous alphabets. The unknown aspects of the Voynich MS writing system still remain and we cannot yet predict what is the ‘correct’ alphabet or identify which are the individual characters. This is primarily a new tool that will facilitate comparison of existing files and give more easy access to the full flexibility of the ZL and GC transliterations. What is meant by that will be shown in the final section of this paper. Furthermore, undoubtedly, Eva will keep its role as an easy means of communication between users.

To set up this alphabet, two major and several minor decisions had to be made. The first major decision was whether this new alphabet should be more synthetic like Currier or v101, or more analytical, like Eva. It was decided to make the STA alphabet synthetic, as otherwise the conversion from v101 to STA would essentially be impossible. For the conversion from extended Eva to STA, this then implied the somewhat lengthy, but straightforward task of finding all possible combinations of Eva text fragments in the ZL transliteration that compose the v101 characters and any other composites.

The second major decision was how to represent the characters in the new alphabet. With Eva and v101 both having of the order of 200 characters, the combination will certainly have even more. Since we should not use high-ascii, a single character will not be enough to represent each character, so it was decided to represent all Voynich characters in STA by two Ascii characters. This has been achieved by subdividing all characters in the Voynich MS into groups, which will be called families in the following. Characters that appear similar will be part of the same family. Families are identified by a single capital letter. The specific characters in each family will be called family members and are identified by a second character. Both the Eva and v101 character sets include characters that are relatively frequent, and others that are quite rare. It was decided to identify the frequent characters by a digit (1-9) and the rare ones by a lower-case character (a-z).

Both the identification of the different families, and the list of members in each family, has been achieved through an iterative process. This would have been an impossible task without a tool that converts transliterations between different alphabets. A few words need to be spent on this tool: “bitrans”, which will also be of significant use for future users of the STA transliteration files. Its purpose is to substitute any combination of characters by any other combination of characters according to a single list of rules, while ignoring certain parts of the file that should be left unchanged. The latter are the comments and the metadata included in the transliteration file. In order to do this properly, the substitution should be ‘greedy’, which means that it should always replace the largest possible combination of characters in the source file that it can match. `bitrans` was originally developed by the French linguist Jacques Guy in Pascal and ran in MS-DOS, but the source of that tool has been lost. I have re-written this tool in C. Like `ivtt`, it is available via the author’s web site and all relevant information may be found in the `bitrans` user guide [9].

The list of families with the most important family members is shown in Table 3. Inevitably, there could be many other equivalent results, but some of the main reasons for this design stem from the Voynich MS text complications mentioned before and are discussed in the following.

Table 3

Definition of STA character families

Family code	Description	Examples	Comment
A	Circles	o a 9	These are the most frequent characters. They are combined into one family because intermediate forms exist.
B	Cross-overs	8 x 8 9	This excludes the vertical bars that also tend to have a cross-over, and the 4-shape that is very distinct.
C	Inverted S-shapes	2 2	
D	Single vertical bar	ff ff	
E	Double vertical bar	ff ff	
F	Single c	c	
G	Double c (without plume)	cc cc	May have a connecting bar at the top.
H	Double c (with plume)	22	May have a connecting bar at the top.
J	Triple c	ccc	May have a connecting bar at the top, but if so, this must connect all three. May have a plume.
K	Even more c's	cccc	Without a connecting bar at the top (these don't exist in the MS).
L	4 – shape	4	
M	Single i	i	
N	Double i	ii ii	
P	Triple i	iii iii	
Q	Even more i's	iiii	
R	Single vertical bar with pedestal (part 1)	ff ff	The ones that end with a single c (or similar shape)
S	Single vertical bar with pedestal (part 2)		The ones that end with 0 or 2+ c's (or other shapes). These are all rare.
U	Double vertical bar with pedestal (part 1)	ff ff	The ones that end with a single c (or similar shape)
V	Double vertical bar with pedestal (part 2)		The ones that end with 0 or 2+ c's (or other shapes). These are all rare.
X	Unclassifiable	x ^	
Z	-	? ???	Used for unreadable characters

An example of Voynich MS text expressed in STA is included in **Figure 5** below. This demonstrates two aspects. The first is, that this transliteration is not suitable for human reading, but only for computer

processing. The second is, that transliterations will tend to be consistent in the choice of character family, but mostly differ in the choice of family member for each character.

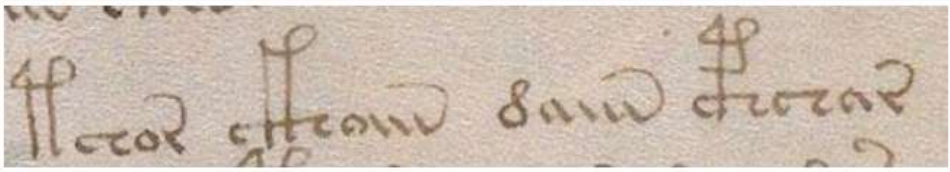
		
FG:	HTOR, DZOM, 8AM, PZTAR-	
CD:	07005A	PSOR/XAM/8AM/WSAR-
IT:	<f36v.P.5;H>	{plant}tchor.ckhoiin.daiin.cphchar-
GC:	<36v.5>	kloy.HAm.8am.Jlay-
ZL:	<f36v.5,+P1>	<%>tchor.ckhoiin.daiin.cphchar
FG:	<f36v.5,+P1>	HTOR.DZOM.8AM.PZTAR
CD:	<f36v.5,+P1>	PSOR.XAM.8AM.WSAR
IT:	<f36v.5,+P1>	tchor.ckhoiin.daiin.cphchar
GC:	<f36v.5,+P1>	kloy.HAm.8am.Jlay
ZL:	<f36v.5,+P1>	<%>tchor.ckhoiin.daiin.cphchar
FG:	<f36v.5,+P1>	E2G1A1C1.U1A1P1.B1A3P1.R1G1A3C1
CD:	<f36v.5,+P1>	E2G1A1C1.U1A3P1.B1A3P1.R1G1A3C1
IT:	<f36v.5,+P1>	E2G1A1C1.U1A1P1.B1A3P1.R1G1A3C1
GC:	<f36v.5,+P1>	E2G1A1C1.U1AaP1.B1A3P1.R1G1A3C1
ZL:	<f36v.5,+P1>	E2G1A1C1.U1A1P1.B1A3P1.R1G1A3C1

Figure 5: old vs. new transliteration systems. Top row: MS text excerpt of f36v, fifth line. Second row: five different original transliterations. Third row: the same transliterations in the IVTFF file format. Bottom row: same, converted to the STA transliteration alphabet. Grey boxes indicate remaining differences between the various transliterations.

4.1. Handling the specific complications of the Voynich writing system in STA

It is worth looking at some specific complications of the Voynich MS writing system. With respect to the intruding or pedestalled gallows, now that a synthetic approach has been decided, the only real problem is, that there are very many of them. Their families have therefore been split in the manner shown in Table 3. Not doing this would have as consequence that not two, but three characters would be needed to represent each STA character. It is stressed again, that none of these decisions are permanent or irreversible. Having the right tool (bitrans) it is easily possible to freely redesign any alphabet to each user's own taste.

Another question about the Voynich MS writing system is whether the very common combination 4o (Eva: qo) should be considered one or two characters. From the simple fact that there are quite a few occurrences of q that are not followed by o, there is a need to have a symbol that represents q by itself, so it was decided to keep the q as a separate character. The v101 alphabet also recognises a few variants of it. v101 also has two composites of q and o, numbered 185 and 186, which shows that in a few cases, a single v101 code will have to be split into two STA codes.⁹

4.2. What to do with the 'strings of i'

The most complicated decision concerns the 'strings of i'. This is not only a problem for the transliteration of the Voynich MS text, but also for its possible interpretation. The 'strings of i' tend to occur near the ends of words. When they are at the end of a word, they are almost always followed by the symbol ɔ, which may very well be nothing else than the word-final version of ɔ, but of course we

⁹ For these numbers, see [2], Table 7.

cannot be certain. In numerous cases, the last symbol is not 𐄂 but 𐄃, but also 𐄄 and 𐄅 occur in that position, though much more rarely. It should be noted that all this is true for strings of "i" of all lengths, though the long strings are rare and for the case of 3 or more "i" not all combinations exist. Table 4 shows the counts for the ZL and GC transliterations, which are reasonably similar.

Table 4

‘Strings of i’ and their frequency in the Voynich MS

Glyph					ZL count					GC count				
	𐄂					140					129			
𐄆	𐄂	𐄃	𐄄	𐄅	110	1729	610	34	46	93	1759	610	37	52
𐄆	𐄂	𐄃	𐄄	𐄅	61	4148	162	14	11	88	4113	169	13	16
𐄆	𐄂	𐄃	𐄄	𐄅	8	166	2	4		10	180	2	1	
	𐄂					1					4			

The first column shows the cases where the ‘strings of i’ are not followed by one of the four listed characters, but by some other character. These are fewer than the counts for strings ending in 𐄂 and 𐄃, but they are more numerous than strings ending in 𐄄 or 𐄅. For this reason, the decision for the STA definition is to create single codes for stand-alone ‘strings of i’ and for strings ending in 𐄂 or 𐄃, but not for the others. This means that some of the Currier and v101 characters need to be split up into two STA code. The complete list of all STA codes is too long to include in this paper, and it may be found at a web page that includes additional material related to this paper [10]. Table 5 shows the part for character family J as an example.

Table 5

STA codes for the relatively small STA family J, including their counts in the ZL and GC files

STA	Combi	Eva	v101	Eva	v101	# ZL	# GC	Comment
J1		eee	ccc	𐄆	𐄆	406	27	
J2		eeb	D	𐄆	𐄆	6	10	
J3		{eee}	d	𐄆	𐄆		332	
J4		e{ee}	cC	𐄆	𐄆		20	
J5		{ee}e	Cc	𐄆	𐄆		39	
Ja		{chh}		𐄆		5		
Jb		{c'hh}	@222;	𐄆	𐄆	4	3	
Jc		{ch'h}		𐄆		2		
Jd		{c'ha}		𐄆		1		
Je			@209;		𐄆		1	

Of the four alphabets, only Eva allows an unambiguous representation of these characters. The problem with the other three alphabets may be illustrated using the character in Figure 3d, which looks like “m” and occurs slightly over 4000 times in the MS (see also Table 4). The standard representation in Currier is “M”, but “IN” would be equivalent and so would “IID”. FSG and v101 allow similar alternatives. While running checks on the correct conversions of the original transliteration files to STA and back, it was discovered that the CD file includes, in the transliteration of *locus* f23v.1, the word: 08AIIIE9 (Eva: odaiiily). Converting this representation to STA, and then back to Currier, we obtain 08A19, which is how this should have been transliterated by Currier in the first place. It is likely that this was a mistake by Currier, and it would in principle be possible to record this as a mistake and modify the original file, but similar issues related to ‘strings of i’ appear numerous times in the GC file. In this case, it is not possible to assume that these are mistakes, and they are likely intentional. The way in which this has been resolved can be kept outside the scope of this paper and will be described at the author’s web site in the near future.

5. Analysis and statistics of the five main transliterations

We are now able to present some figures related to the completeness and accuracy of the existing transliterations. For their completeness, see Table 6. Note that more recent versions of the transliteration files exist, but these only differ in the contents of the metadata.¹⁰ The number of characters is likely to be less than the actual number of characters in the MS, because some of the STA codes likely represent more than one character (as in family J, see Table 5).¹¹

Table 6

Number of loci and number of STA characters included in each of the main transliterations.

File Code	Version	Nr. of <i>loci</i>	Nr. of characters (STA)
FG	1c	4060	135,604
CD	0d	2196	65,168
IT	0d	5216	155,340
GC	0c	5367	157,095
ZL	1r	5389	157,264

In terms of accuracy, what we can show is the consistency between files. Table 7 shows that correspondence between characters (white cells) is generally higher than 95%, with the exception of the GC file, where it is slightly over 90%. Correspondence between character families (light blue cells) is up to 99%. The GC transliteration differs most from the others in terms of characters, as it recognizes numerous subtle variants of main characters. However, the very different ZL and GC transliterations turn out to be the most similar pair in terms of character families. It is, of course, not known whether v101 recognises ‘too many’ different characters, or Eva (and the others) ‘too few’.

Table 7

Correspondence between Voynich MS transliterations, expressed in % of common symbols (white boxes) or % of common STA families (light blue boxes).

	FG	CD	IT	GC	ZL
FG		95.4	97.0	91.4	97.4
CD	96.9		96.8	90.5	96.5
IT	98.2	97.8		91.4	97.3
GC	98.2	97.6	98.5		92.0
ZL	98.4	97.8	98.5	99.0	

¹⁰ These source files have been added to [10]

¹¹ Note that each case of ‘unknown number of unreadable characters’ is counted as 2 STA codes.

5.1. Hapax legomena

Probably the statistic that is most sensitive to transliteration errors or differences is the number of *hapax legomena* in the text. These are words that occur only once in a text. This quantity is especially sensitive to the identification of word spaces, something that is not reflected in Table 7. Both the absolute and relative number of *hapax legomena* depend strongly on the length of the text, so we should only compare this result for transliterations of similar length, namely the IT, GC and ZL files (see Table 6). We use the terms word tokens, or just tokens, for the count of words in the text, and word types, or just types, for the number of different words. Note that both the GC and ZL files indicate uncertain spaces, so we can compute this statistic twice for each of these two files: once for all spaces (including the uncertain spaces) and once for certain spaces only. In the former case we observe more word tokens and in the latter case fewer word tokens in the same text. The result for these three files (five cases) is shown in Table 8 below and visualized in Figure 6, showing the percentage of *hapax* per word token.

Table 8

Counts of words, *hapax legomena* and ratios in % for five transliteration versions. Only standard paragraph text has been used in this analysis

File	Spaces	Tokens	Types	Hapax	Hapax/Token	Hapax/Type
IT	All	33641	7464	5232	15.6%	70.1%
ZL	All	34974	7178	4935	14.1%	68.8%
ZL	Certain	32509	8133	5859	18.0%	72.0%
GC	All	36658	8602	6100	16.6%	70.9%
GC	Certain	34415	9264	6753	19.6%	72.9%

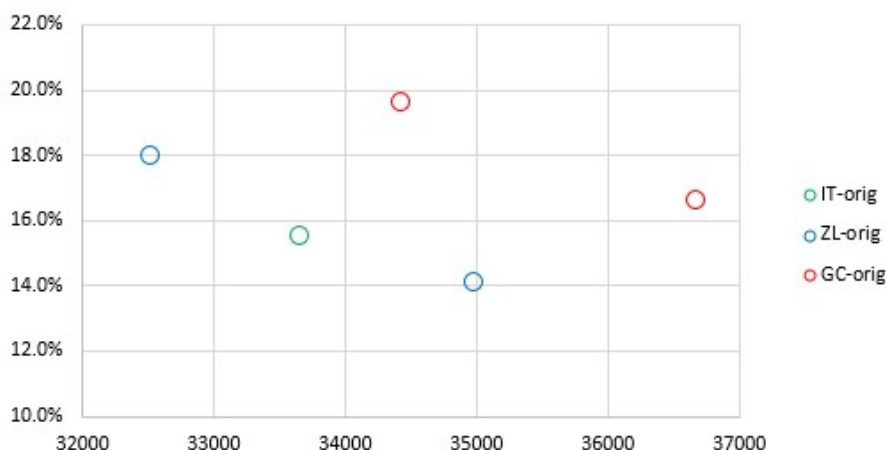


Figure 6: The percentage of *hapax legomena* per word token for the Voynich MS text, according to different versions of the transliteration files. The leftmost symbols for the ZL and GC files are for certain spaces only. The rightmost symbols are in case uncertain spaces are also counted as spaces.

The numbers have a very wide range, from 14% to 20%, showing that the result is completely dominated by the accuracy of the transliteration.

5.2. Use of bitrans

The tool *bitrans* has already been introduced before. It has been used to convert the original transliterations to STA and back. This type of conversion is bi-directional because it represents a one-to-one mapping. The same conversion table can be used in both directions. However, the tool can also

be used to create many-to-one mappings, and in this case the conversion table only works in one direction. This important capability allows users to easily create their own (shorter) transliteration or analysis alphabets from the long STA, v101 or extended Eva alphabets. Since Eva is not suitable for making most types of statistical analysis, I have long used a mixture of Eva and Currier, which I have called “Cuva”.¹² This was never meant to be a particularly important or accurate alphabet. Many other versions could have been defined. An example of its use is the analysis presented in [11]. In the following I will use a similar *ad hoc* transliteration alphabet called “Stava”, as it is a simplification of the full STA alphabet. The `bitrans` table to convert STA to Stava may be found in [10]. All transliterations can be converted to Stava by first converting them to STA. The analysis in Table 8 and Figure 6 can now be repeated after converting the three transliteration files to Stava. The result is shown in Figure 7.

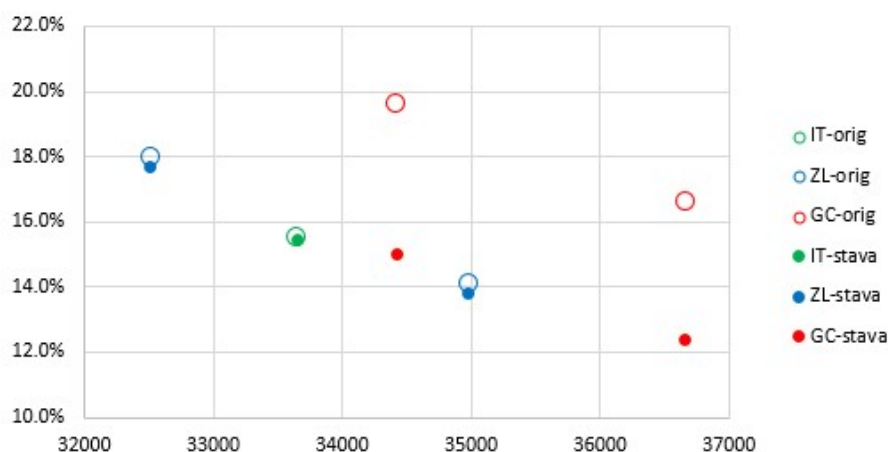


Figure 7: ratio of hapax legomena and word types for several different transliteration files and two different transliteration alphabets

This shows that the original alphabet of v101 creates more *hapax legomena* than the two other alphabets, while after conversion, the only remaining impact is from the number of observed word spaces. The main purpose of this example is to show the way to use the tools, and the impact of converting alphabets. Any linguistic consequences of such an analysis need to remain outside the scope of this paper.

As a final example of the use of the transliteration files and tools, let us look again at the analysis described in [11], which was originally performed more than 20 years ago, using prototype versions of the ZL file and `ivtt`, and using the Cuva alphabet. The latter should be considered more an analysis alphabet than a transliteration alphabet, and we can now convert every transliteration file to this alphabet using `bitrans`. This is most conveniently done by first converting to STA, and then use the approximative conversion from STA to ‘nearest basic Eva’. This approximation finds a close equivalent in basic Eva of the full STA alphabet. Using this table on any file that was originally in FSG, Currier or Basic Eva corresponds to an exact conversion. For extended Eva and v101 it represents a simplification, by which a great amount of detail (such as all rare characters) in the original transliteration file is lost. The STA to basic Eva table is quite long and may be found in [10]. It can be used as a template by users who wish to create their own transliteration alphabets. The conversion from basic Eva to Cuva requires a second, shorter `bitrans` file, which is also provided in [10].

The analysis in [11] looks at the bigram distribution of every single page in the MS. A simple plot near the bottom of the page shows the frequency of some of the most frequent bigrams over all pages. The more interesting plot is based on a PCA analysis of vectors of all bigrams on each page.¹³ This will now be repeated for both the ZL and GC files, while using different analysis alphabets. After conversion of both files to basic Eva, extracts for a single page are most conveniently made using a script that

¹² The Cuva definition is presented here: http://www.voyrich.nu/extra/sp_analysis.html#cuva

¹³ Principle Component Analysis, a standard way to show the most important features of a multi-dimensional cloud of points. The method implemented by the author in 1998 was not a standard PCA, but a personal development which leads to very similar results.

repeatedly invokes `ivttt`. The PCA plots have been made for both files, and the results are virtually identical. They are also very similar to the result of the original analysis. Figure 8 shows the original result from before 2000 and the new result based on the GC file side by side, to demonstrate this. Similarly, the bigram count plots for all three cases are virtually identical and do not need to be shown here. The complete set of figures for all analyses discussed here is provided in [10] for reference.

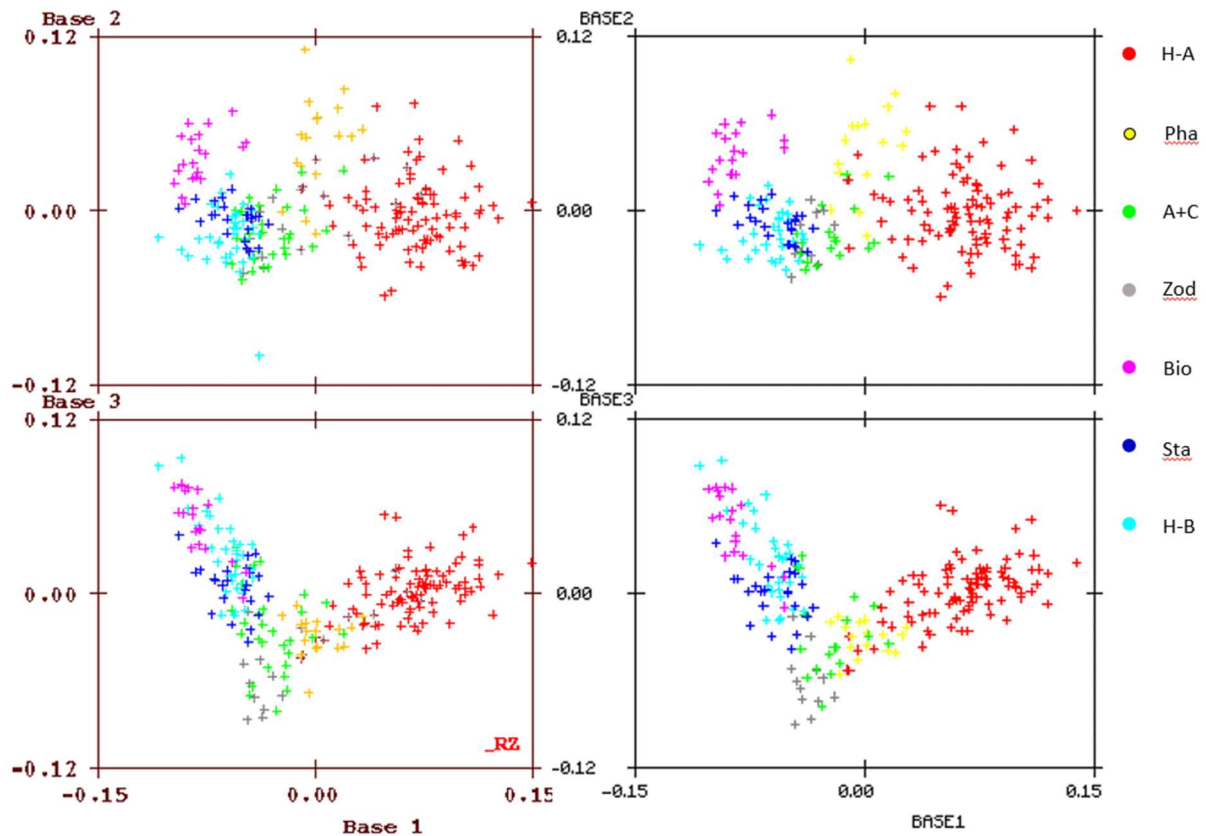


Figure 8: PCA plots for each page in the MS made before 2000 (left) and at present based on the GC file (right). Both cases use the Cuva alphabet. Pages are colour-coded depending on the type of illustration: H-A/B = herbal A/B, Pha=pharmaceutical, A+C=astronomical+cosmological, Zod=zodiac, Bio=biological, Sta=stars/recipes.

To demonstrate the ease of changing transliteration alphabets and repeating analyses, two more alphabets will be defined in the following. The first is very similar to Cuva and will be called Cuva-1. It consists of only very few changes to Eva, resulting in a mixed case alphabet. The second is a small set of changes to Cuva-1, resulting in Cuva-2. Details for both are provided in Table 9. This is part of the author's on-going research, and the reasoning behind this is not relevant for the present paper. Setting up the related `bitrans` tables and re-running the analysis tool(s) was done in a matter of minutes.

Using Cuva-1, the PCA plots hardly change from Figure 8, and need not be shown here. Figure 9 further below shows the impact of changing from Cuva-1 to Cuva-2 on the bigram frequency plots, both based on the GC transliteration. The left part of this Figure (for Cuva-1, based on the GC file) is very similar to the Figure shown in [11].

Table 9Definition of the (*ad hoc*) analysis alphabets: Cuva-1 (left) and Cuva-2 (right)

Char	Eva	Cuva-1	Char Cuva-1	Char Cuva-2	Bitrans syntax	
α	ch	S	.ʔ	.α	%q	%S
ʒ	sh	Z	α	α	Se	S
ɰ	in	N	ʒ	ʒ	Ze	Z
ɰɰ	iin	M	ʔʒ	ʔʔ	te	t
ʔʔʒ	cth	ote	(Similarly for three more)			
(Similarly for three more)						

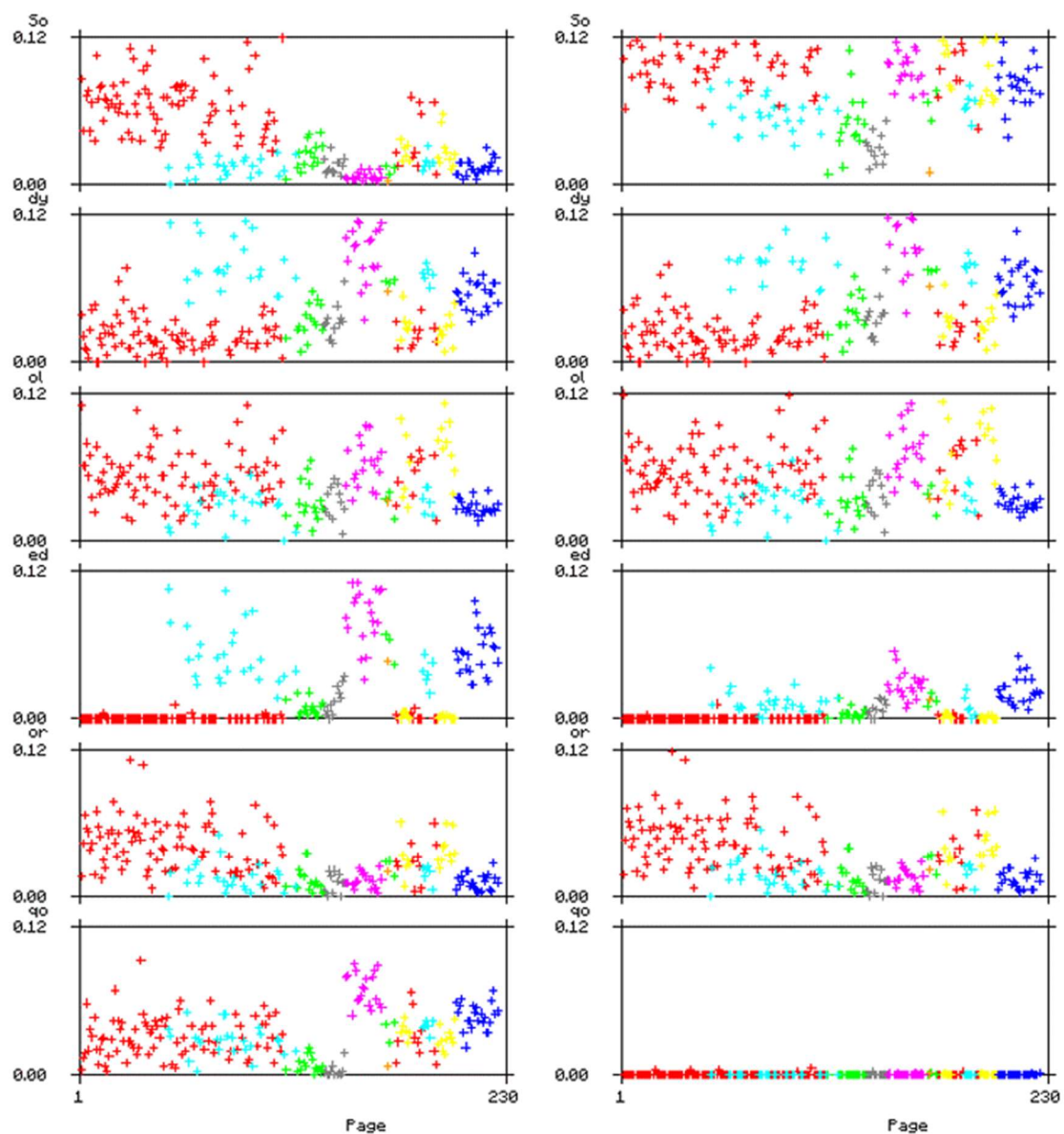


Figure 9: frequency of some of the most common bigrams, for each page in the MS. Pages are colour-coded depending on the type of illustration. Both figures are based on the GC transliteration, converted to Cuva-1 (left) and Cuva-2 (right). The change of alphabet has a major impact on the result. See also a similar result in [11]. For the colour legend see Figure 8.

6. Summary and conclusions

Several electronic versions of the Voynich MS text have been created by different people over the last decades and have been used in all statistical analyses of the MS text by the Voynich community. The five most important files are described in this paper. By introducing a common file format for these transliterations, it has now become possible to process all five files with the same tools. A standard tool for pre-processing these files, including selection of text based on a large set of criteria, has been presented.

By introducing a superset of all transliteration alphabets (named STA in this paper), it has furthermore become possible to compare all transliterations directly. This allows us to obtain a first insight into the accuracy of these transliterations, which is presented in this paper.

Finally, this paper demonstrates the advantages of using standard a standard file format, a single common alphabet and dedicated tools based on both, in performing Voynich MS text analysis. It gives all users the capability to create their own transliteration without having to physically transliterate all pages of the MS, a painstaking process taking several weeks of continuous work. Instead, they just need to set up their own alphabet as a conversion of the STA alphabet and run a simple tool to convert any of the existing transliteration to the user's own alphabet.

7. Acknowledgements

I wish to acknowledge the leading examples of William Friedman and his study group, and of Prescott Currier and Mary D'Imperio, in preparing the first computer-readable transliterations of the Voynich MS text. They prepared the path for all future text analyses of the Voynich MS. Following in their footsteps are Jim Reeds and Jacques Guy for typing in the FSG printout, Takeshi Takahashi and Glen Claston for making more recent publicly available transliterations, and Gabriel Landini for his support to the definition of Eva and the creation of the ZL transliteration. I thank the organisers of the 2022 Voynich conference at the university of Malta for the opportunity to present this work.

8. References

- [1] J. Reeds: William F. Friedman's Transcription of the Voynich Manuscript, Cryptologia XIX, 1995, pp.1-23.
- [2] R. Zandbergen: Text Analysis – Transliteration of the text: <http://www.voynich.nu/transcr.html>, accessed 6/11/2022
- [3] M. D'Imperio (ed.): New Research on the Voynich Manuscript: Proceedings of a seminar, 30 November 1976. Privately printed pamphlet, Washington D.C., 1976.
- [4] Th. Petersen: hand transcription of the Voynich MS, items 1615.1 and 1615.2 of the George C. Marshall Museum & Library, Lexington (Va), USA
- [5] T. Takahashi: Voynich Manuscript – transcription, <http://www.voynich.com/pages/>, accessed 6/11/2022
- [6] R. Zandbergen: IVTFF – Intermediate Voynich MS Transliteration File Format, version 1.7, 10/04/2020, http://www.voynich.nu/software/ivtt/IVTFF_format.pdf, accessed 6/11/2022.
- [7] R. Zandbergen: Intermediate Voynich MS Transliteration Tool – User Manual, Issue 1.1, 10/4/2020, http://www.voynich.nu/software/ivtt/IVTT_manual.pdf, accessed 6/11/2022
- [8] R. Zandbergen: location of main transliteration resources, <http://www.voynich.nu/transcr.html#links>, accessed 6/11/2022
- [9] R. Zandbergen: Bitrans – bi-directional Translation / Substitution tool – User Manual, Issue 1.4, 6/11/2021, http://www.voynich.nu/software/bitrans/Bitrans_manual.pdf, accessed 6/11/2022
- [10] R. Zandbergen: Transliteration of the Voynich MS text – extra material, http://www.voynich.nu/extra/conf2022_extra.html, accessed 15/11/2022
- [11] R. Zandbergen: the Currier languages revisited, <http://www.voynich.nu/extra/curabcd.html>, accessed 6/11/2022