A Probabilistic Approach to Syntactic Variation in Biblical Hebrew

Etienne P. van de Bijl, Cody Kingham, Wido van Peursen, and Sandjai Bhulai[1] ,

[1] *Vrije Universiteit Amsterdam, Amsterdam, Netherlands*

December 3, 2018

There is currently a disagreement in the field of Hebrew Studies on the methods used to date individual books within the Hebrew Bible. An important question in this debate is if linguistic differences between the books are significant enough to warrant a diachronic explanation. In this project, we seek to answer whether differences in syntax between the books are large enough to merit groupings into Standard or Late Biblical Hebrew. We use a statistical tool called Markov Chains, which models transition dependency in sequences. Our method takes into account word and phrase order for parts of speech, phrase functions, and phrase types. We then cluster the books based on their statistical similarities. Our results may corroborate key claims of the diachronic approach.

1 Introduction

Many biblical scholars from the past and the present are convinced that biblical texts can be dated on the basis of language (Hornkohl, 2017). Most modern scholarship acknowledges that Biblical Hebrew contains distinctive historical phases (Hornkohl, 2013). These periods are reflected not only by the content of the biblical books, but perhaps also by subtle linguistic clues. A text's preference for certain morphemes, lemmas, vocabulary, or grammatical forms are used as evidence for its date (e.g. Sáenz-Badillos, 1993). Hurvitz (1998), especially, has tried to build a scientifically sound method of classifying texts as early or late. He argues that an accumulation of late features, identified through texts which are known to be late, can indicate whether other texts contain late Hebrew.

Yet, Hurvitz's method of linguistically dating Hebrew texts is not without opponents. Young, Rezetko, and Ehrensvärd critique key assumptions in Hurvitz's methodology (Young, 2005; Ehrensvärd, 1997; Young and Rezetko, 2008). These scholars stress the relative linguistic homogeneity among the books of the Hebrew Bible: "The question that remains, however, is whether the Hebrew Bible displays adequate linguistic variety to sustain the scholarly consensus that it was composed over a period of approximately a thousand years." (Young and Rezetko, 2008, p. 46) In this project, we seek to quantify the differences in syntax between books of the Hebrew Bible, and to see whether they support Hurvitz's hypothesis of an "accumulation" of features.

Hebraists have already used syntactic tendencies as an indicator of dating. The value of using syntax over against word level features for linguistic typology is that syntax is a less conscious aspect of language (Eskhult, 2005; Chambers and Schilling, 2013). Theoretically, it is harder to modify one's use of syntax than lexicon. Givón looks at word order for verbs, subjects, and objects in Biblical Hebrew, finding a shift from the SVO found in earlier texts to VSO in later ones (1977). Eskhult argues that Biblical Hebrew gradually changed its primary narrative tense from the wayyiqtol verb to the qatal (2000). The recently completed project, "Does Syntactic Variation Reflect Language Change?" at the Vrije Universiteit Amsterdam's Eep Talstra Centre for Bible and Computer (ETCBC) has looked extensively at syntax as part of the diachronic question. This project, also co-sponsored by the ETCBC, has progressed in the same vein.

We aim to test whether syntactic information indeed provides enough data to reliably classify books of the Hebrew Bible. Traditionally two primary groups are outlined: Early Biblical Hebrew (EBH) and Late Biblical Hebrew (SBH). The primary disagreement in scholarship comes with the 'early' classifications. Thus, in the interest of objectivity, we prefer the more neutral moniker of Standard Biblical Hebrew (SBH). We seek to establish whether differences in syntax result in clusters that align with the traditional SBH or LBH groups, and to describe which features, if any, contribute most to the divisions.

A natural way to model word and phrase order is by treating grammatical units as sequences. A Markov Chain is a statistical model that can be used to model and compare sequences. The process of constructing a Markov model involves counting all observed sequences together in a table, i.e. a matrix. The raw counts are then transformed into probabilities. These two procedures allow us to easily compare similarities between books and, through the probabilities, compare books of different sizes.

In section 2, we describe the data used for this research. Section 3 explains how we find similarity between biblical texts and how we validate this similarity. The results are then presented in section 4. Section 5 describes the findings of this research in the discussion section. We also discuss avenues for future research.

2 Data

This research utilizes the linguistic annotation data published by the Eep Talstra Centre for Bible and Computer (ETCBC). The annotations are contained in a data package, Biblia Hebraica Stuttgartensia Amstelodamensis (BHSA) which is open-source Hebrew Bible data available in a Python tool, called Text-Fabric (Roorda, 2018). The annotations are stored as features on nodes within a graph structure. The nodes are linguistic units of words, phrases, and clauses for all books of the Hebrew Bible. The features mark grammatical and syntactic categories such as lexeme, morphology, part of speech, phrase type, and phrase function. We only use the syntactic annotations, though other categories are available (including discourse analysis). The list below shows the relevant features for our model.

Word: part-of-speech noun, article, preposition, etc.
Phrase: function object, subject, adjunct, etc.
Phrase: type verbal (VP), nominal (NP), etc.
Clause: type Way+X, Way+ø, W+Qtl, etc.
Clause: domain Narrative (N), Discursive (D), Quotation (Q) and Unknown (?)

Figure 1 shows the hierarchical format of each unit. Below each word the part-of-speech value is given. The green lines show the phrases within the clause and the blue border shows the clause within a sentence.

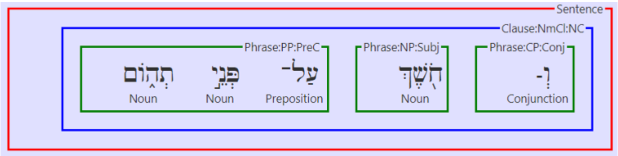

Figure 1: *Annotation format for a sentence in Genesis 1*

2.1 Preparing the Biblical Data

To prepare the data, we apply a custom Python module which loops through a list of supplied biblical books and gathers linguistic annotations per clause. The module uses data classes provided by Text-Fabric to access annotations on linguistic objects. Since the goal of this project is to test the classic divisions of Late Biblical Hebrew (LBH) and Standard Biblical Hebrew (SBH, also "Early" Biblical Hebrew), we begin with a basic list (see Young and Rezetko, 2008, pp.10-11; we, however, include Song of Songs and Ecclesiastes in our dataset.):

SBH Genesis, Exodus, Leviticus, Deuteronomy, Joshua, Judges, Kings, Samuel
LBH Song of Songs, Ecclesiastes, Esther, Daniel, Ezra-Nehemiah, Chronicles

Since the books of 1 and 2 Kings, 1 and 2 Samuel, Ezra and Nehemiah, and 1 and 2 Chronicles are traditionally understood as single compositions, we combine them under their single, respective titles. We exclude all books that have a debated categorization for this initial experiment. Because narrative and quotation material differ significantly in syntax (e.g. Niccacci, 1994), we separate clauses into collections of narrative and quotation. The result is a series of nested datasets for each syntactic feature, broken down by discourse type and then into books:

/FEATURE/DISCOURSE/BOOK/DATA

For each feature and clause the module assembles bigrams which can subsequently be counted. Figure 2 shows an example from the Hebrew text using the feature of phrase functions.

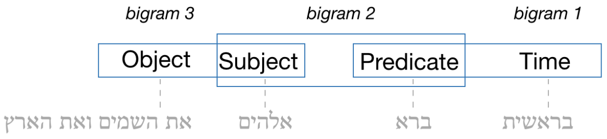

Figure 2: *Phrase Function Bigrams in Genesis 1:1 (N.B. Hebrew is right-to-left*

Each of the three larger rectangles represents a separate bigram which is counted in our dataset. In this way syntactic tendencies are simplified into tendencies of sequence. Each feature label is gathered with other labels in the clause into a Python list. A sample of the raw data for the first two quotation clauses in Genesis (Genesis 1:3 and 1:6) is provided below for three separate features. Each set of embedded brackets contains the elements from a whole clause. For each of these clauses, the algorithm iterates over the contained elements to construct the bigrams (e.g. 'verb -> subs' for word part of speech). The outer-most brackets enclose the dataset for a given feature.

word part of speech, quotation [['verb','subs'], ['verb','subs', 'prep','subs','art','subs']]
phrase types, quotation [['VP','NP'], ['VP','NP','PP']]
phrase functions, quotation [['Pred','Subj'], ['Pred','Subj','PreC']]

Since many books in the Hebrew Bible copy material from other books (e.g. Chronicles from Samuel and Kings), we have applied a filter that removes clauses with a high degree of similarity (>75%) to clauses in other books. This allows us to only compare syntactic content original to the book itself. To do this, we utilize another package, Parallels, published by researchers of the ETCBC (Roorda and Naaijer, 2018).

3 Methodology

Herein we detail the statistical approach that undergirds our model of book syntax and similarities. We also show how we have validated the similarity measurement used in this experiment by applying a clustering algorithm. Those who require a simpler explanation of the math can instead consult our Jupyter notebook (Bijl and Kingham, 2018).

3.1 Markov Chain

Syntactic annotation can be observed as a sequence of outcomes of a chance experiment. It is very unlikely that these outcomes are independent because syntactic dependency exists between the elements. A model that takes into account transition dependency is a Markov Chain. Markov Chains are frequently used in linguistic applications (Al-Anzi and AbuZeina, 2016).

According to Leon-Garcia (2008): "A random process is said to be Markov if the future of the process given the present is independent of the past". Boxma (2002) describes the Markov Chain with the following considerations. Consider a finite set of states $\Omega = \{s_1, s_2, ..., s_m\}$ and a stochastic process $\{X(t), t \in T\}$ that moves along these states. We can interpret t as time and call $X(t)$ the state of the process at time t. In the linguistic setting, this process can model a sequence of annotations. If $T = 1, 2, ...,$, we call the process a discrete time stochastic process. A discrete time stochastic process with state space Ω is a Markov Chain if the successive random variables have the following dependence structure:

$$P(X_{t+1} = j | X_t = i, X_{t-1} = i_{t-1} ..., X_1 = i_1) =$$

$$P(X_{t+1} = j | X_t = i)$$

for all $t = 0, 1, 2, ...$ and all states.

The right hand side is called the stationary one-step transition probability and is given usually by p_{ij}. For these transition probabilities, the two following properties must hold:

$$0 \leq p_{ij} \leq 1 \; and \sum_{j \in \Omega} p_{ij} = 1 \; \forall i \in \Omega$$

This means that all probabilities are positive and the sum outgoing probabilities of a state must be equal to one. Obtaining the transition probabilities can be obtained by computing the Maximum Likelihood Estimator: Say $n_{ij}(t)$ are the number of transitions from state i to state j at time t and $t \in T$. Then the transition probabilities are estimated by the following formula (Anderson and Goodman, 1956):

$$\hat{p}_{ij} = \frac{\sum_{t=1}^{T} n_{ij}(t)}{\sum_{k=1}^{\Omega} \sum_{t=1}^{T} n_{ik}(t)} \; \forall i, j \in \Omega$$

This is the Maximum Likelihood Estimation of the transition probabilities. The syntactical annotations can thus be modeled by these transition dependencies on all levels.

3.2 Markov Chain Similarities

Markov Chains can model the syntactic tendencies of each biblical book. However, a method is needed to express the similarity between the syntax of two given books. Therefore, we require a measurement that indicates how similar two Markov Chains are. Measuring similarities between Markov chain transition matrices can be performed in different ways. Dyer et al. (2006) compare Markov Chains on the basis of their mixing time. We are however not interested in the steady state probabilities since the transition dependency focus would be lost. Davismoon and Eccles (2010) made a comparison between the two transition matrices by taking simply the Euclidean distance of the transition probabilities. This approach however does not take into account the probabilistic point of view that each row of a transition matrix is a conditional distribution. Therefore, using a statistical distance measure to compare the conditional distribution seems to be more appropriate to compare the outgoing probabilities of each state. If we want to define the similarity or distance between two elements, the notion of a metric is required: Say X is a set and d a function that maps $X \times X \to \mathbb{R}$. Then the pair $M = \{X, d\}$ is called a metric space if and only if d satisfies the following properties:

Non-negativeness $\forall x, y \in X : d(x, y) \geq 0$
Identification $\forall x, y \in X : d(x, y) = 0 \iff x = y$
Symmetry $\forall x, y \in X : d(x, y) = d(y, x)$
Triangle ineq $\forall x, y, z \in X : d(x, z) \leq d(x, y) + d(y, z)$

When the triangle inequality condition does not hold, this metric is a semi metric. Now, we have to choose the statistical distance function which would be most appropriate. Commonly used as a statistical distance is the Kullback-Leibler divergence. This measurement expresses the difference of one probability distribution to another one. Unfortunately, this measurement does not have the symmetry property. It would be undesirable that the similarity in syntax between two books is not symmetrical. Often called "the statistical distance" is the Total Variation distance. In this distance measure, the distance is defined by the maximum difference between the probabilities assigned to a single event by the two distributions. In mathematical notation, given two probability distributions P and Q both supported on the same space Ω:

$$D_{TV} = \max_{x \in \Omega} \mid P(x) - Q(x) \mid$$

One related distance measure to the TV distance is the Hellinger distance. Pollard (2015) states that the Hellinger distance has advantages over the Total Variation distance. According to Pollard, Hilbert spaces have nicer properties than general Banach spaces. The Hellinger distance is defined as:

$$D_H = \frac{1}{\sqrt{2}} \|\sqrt{P} - \sqrt{Q}\|_2$$

The Hellinger distance is a metric that satisfies the triangular inequality. One useful property for measuring similarity is that the maximum distance is 1. Therefore, this distance metric is a bounded metric. Based on these advantages over the Total Variation distance, we use the Hellinger distance to calculate the distance between two conditional probability rows of two transition matrices. The similarity in syntax is determined by taking the average of the Hellinger distance between the conditional probabilities of the transition matrices.

3.3 Clustering

Using the similarity value between biblical books, we can group books which are similar to each other. The rational here is to find out whether the similarity measure between books is coherent with the classic typological divisions in biblical research. In order to validate the distance measure, we use a hierarchical clustering algorithm and a non-hierarchical clustering algorithm named k-medoids.

3.3.1 Hierarchical Clustering

In hierarchical clustering, clusters can overlap and the number of clusters is not pre-defined. The end product of such a method is a dendrogram, which resembles a family tree. In a dendrogram, the resulting clusters are given on the defined level. There are two methods to perform hierarchical clustering: divisive clustering and agglomerative clustering. Since there are Python packages for agglomerative clustering and divisive hierarchical clustering is more complex than agglomerative, we use this method to cluster the biblical books. The agglomerative algorithm considers each element as a cluster. Iteratively, the algorithm tries to merge two clusters with the least distance. When merging two clusters, new distances between the merged cluster and all other clusters must be defined. The linkage function is the function that redefines the new distances between the merged cluster A and B and cluster C. There are several linkage functions possible: single, complete, average, weighted, centroid and Ward. In the single linkage, the new distance between two merged clusters A and B and another cluster C is defined by the least distance between an element in A or B and an element in C. The complete linkage takes the maximum distance. These two linkages however are very extreme. A less extreme linkage would be to use the average linkage or the Ward linkage. In the Ward linkage, the distance is defined by the increase in standard deviation within the clusters when clusters are merged. Thus we select Ward linkage for our analysis.

3.3.2 K-medoids clustering

A non-hierarchical clustering algorithm groups objects into a pre-defined number of clusters based on optimal distances between the clustered elements. In contrast to hierarchical clusters, these clusters do not overlap. The algorithm can be used for separating the biblical texts into a predefined number of clusters which do not overlap. By this method, it is possible to find the distinguished biblical texts groups (SBH and LBH) on certain annotation levels. In comparison to k-means clustering, where points in a certain space are considered to be centers of clusters, K-medoids selects data-points (in this case biblical books) as centers, and tries to label the data-points to the center which is closest. In the context of the biblical books, the clustering algorithm randomly initializes some, two or more, books as centers. Other books are then linked to the center which is nearest. The end result is a set of clusters. Within each cluster, a new center is searched for in that group of objects which results in the least sum of distances. After finding new centers, the books are again linked to the nearest center. This procedure repeats for a certain number of iterations. The algorithm is performed multiple times to find how often books fall into the same cluster.

4 Results

In this section we present the project results.

4.1 Constructing Markov Chains

In order to construct Markov Chains, the number of average transitions per biblical text must be examined. When the number of transitions is too low and the number of states is too large, then the transition probabilities are not reliable enough.

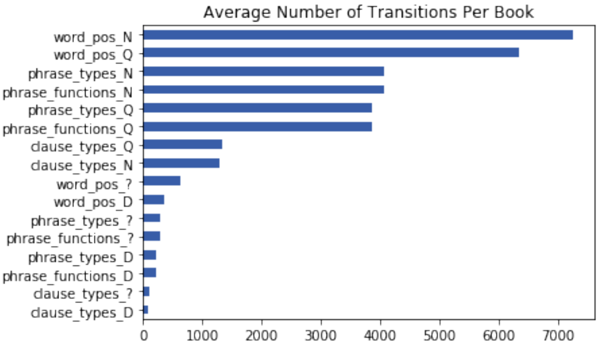

Figure 3: *Average number of annotations per biblical book*

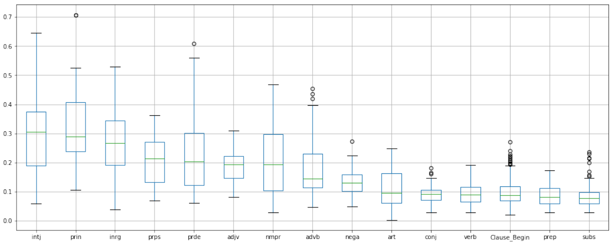

Figure 4: *Statistical Distances for Part of Speech in Q*

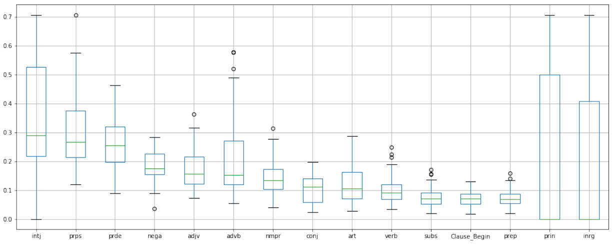

Figure 5: *Statistical Distances for Part of Speech in N*

Figure 3 shows the average number of annotations for all biblical book at each level. On the left axis, the annotation level is given with the according domain. As can be observed, the number of annotations for the ? and D domain are below 1000. Given the number of different states, constructing Markov Chain transition matrices for these domains is not usefull since the transition probabilities are not reliable. The Markov Transition Matrices are constructed for the domains N and Q on part-of-speech and phrase function/type level.

4.2 Statistical Distances

After constructing the Markov Chain matrices with the transition probabilities on N and Q domain and on the part-of-speech, phrase function and phrase type levels, the statistical distances are calculated for each of the conditional probabilities. Figures 4, 5, 6, and 7 show box plots of the distances between all books on four different levels. It can be observed that the largest average conditional distance is around 0.30. On the phrase function level, there are more outliers but the differences are smaller. Furthermore, the boxplots of the phrase function levels (figures 6 and 7) are more spread compared to the part of speech level. This means that the distances on the phrase function level are more diverse. It can be observed that on the part of speech level the boxplots between Q and N levels are also different. This indicates that the transition probabilities between Q and N level are indeed very different.

Figure 8 shows the average distance between different levels. It can be observed that the two domains N and Q are distinguishable as the average distance is least intern. Interesting is that the Q domain phrase types and part-of-speech distance are approximately equivalent. As the distances itself are not self-explanatory, the relative distance between the levels shows how the levels are related.

4.3 Hierarchical Clusters

Using this data, we create hierarchical clusters. Table 1 shows a dendrogram for each feature and domain in our model (six in total). Each linked branch represents a discrete cluster. The lower the bracket, the more similar the pair is statistically. Groups of links that receive a unique color are sufficiently distinct to constitute their own cluster.

Two important tendencies emerge in the dendrograms. First, phrase types in Q and word part of speech in Q displays, by far, the greatest distance between clusters, with a distance of around 0.70 between the two identified clusters. Second, throughout all the plots we can observe a mixture of SBH and LBH books. Using these features alone, there is no clear corroboration of the classic two-part division between SBH and LBH. For instance, in phrase types of narrative clauses (phrase_types N) Joshua and Daniel are clustered together. But we can see some familiar tendencies for individual books. For instance, with Ezra-Nehemiah, we see links with Daniel (phrase_functions N, phrase_types Q), Esther (word_pos N), and Ecclesiastes (phrase_functions Q, word_pos Q). Likewise, Esther often falls together with the LBH books of Chronicles, Ezra-Nehemiah, and Chronicles. Thus, we see that while there is no corroboration of the classic division per feature, there does seem to be patterns of LBH and SBH books favoring each other within the individual plots.

This observation led us to wonder what clusters might be observed if the distances for all six datasets were averaged together. Table 2 shows the generated

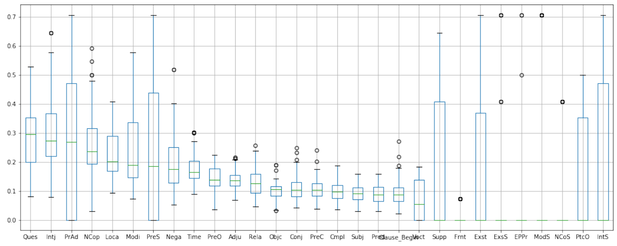

Figure 6: *Statistical Distances for Phrase Functions in Q*

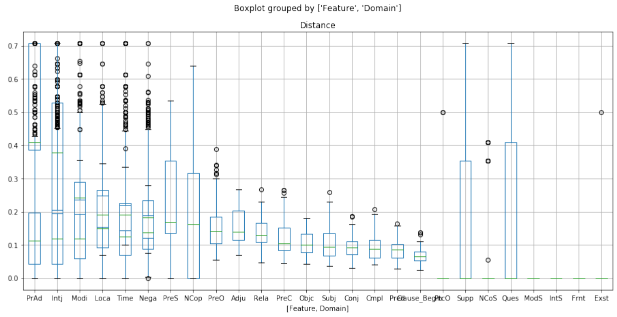

Figure 7: *Statistical Distances for Phrase Functions in N*

clusters when book distances were averaged for narrative and quotation for word part of speech, phrase type, and phrase functions. The results reveal a striking similarity to the classic two-part SBH and LBH divisions with a few surprises. The clusters suggest the following groupings:

group 1 Ezra-Nehemiah, Esther, Daniel, Song of Songs, Leviticus
group 2a Exodus, Deuteronomy, Ecclesiastes, Chronicles
group 2b Judges, Genesis, Samuel, Kings, Joshua

This correspondence to the classical SBH and LBH division, especially seen between group 1 and group 2b, only emerges after combining book distances across all six of our datasets. This seems to suggest that no single feature within the books is by itself indicative of either group. Rather, it is the combination of features which yields the familiar clusters. We interpret this as a potential corroboration for the diachronic method of seeking an "accumulation" of characteristic linguistic features (e.g. Hornkohl 2013). These results, while tantalizing, still require confirmation. Specifically, while the resulting groups appear to suggest the validity of the classic divisions, the clusters we see here remain connected to the clustering method itself, which only seeks to reach an optimal clustering based on the data it compares. Perhaps different combinations of books, such as the exclusion of somewhat debated books such as the Song of Songs, would produce different clusters. These kinds of tests, due to the limits of this project, must be left to future investigation.

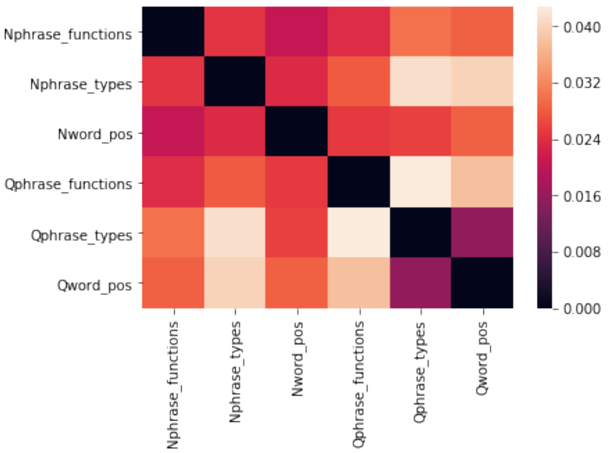

Figure 8: *Average statistical distance between annotation levels*

4.4 Future Research

In our project, we used Markov Chains to model language. This model is simplistic in the sense that it only considers the linear transitions of linguistic units. Yet, language is complex and hierarchical. This suggests that simplistic, linear sequences may not be sophisticated enough. Therefore, we encourage future research to explore a language model that can take more dependencies within sentences into account.

Finally, future research should seek to falsify or confirm our results by looking at other possible combinations of books. This includes processing multiple iterations of book groupings. For instance, an algorithm might first cluster Genesis, Exodus, Chronicles, and Ezra-Nehemiah to see whether Chronicles and Ezra-Nehemiah indeed cluster together, as would be expected with the classical divisions. We believe applying this method, especially with multiple iterations, could help clarify the robustness of the clusters we obtained.

References

Al-Anzi, F.S. and D. M. AbuZeina (2016). “A Survey of Markov Chain Models in Linguistics Applications”. In: *Computer Science and Information Technology*, pp. 53–62.

Anderson, T.W. and L.A. Goodman (1956). *Statistical Inference About Markov Chains*, p. 92.

Bijl, Etienne Pieter van de and Cody Kingham (2018). *Analysis Notebook: A Probabilistic Approach to Linguistic Variation and Change in Biblical Hebrew*. `https://nbviewer.jupyter.org/github/ETCBC/Probabilistic_Language_Change/blob/master/analysis.ipynb`.

Boxma, O.J. (2002). *Stochastic Performance Modelling*.

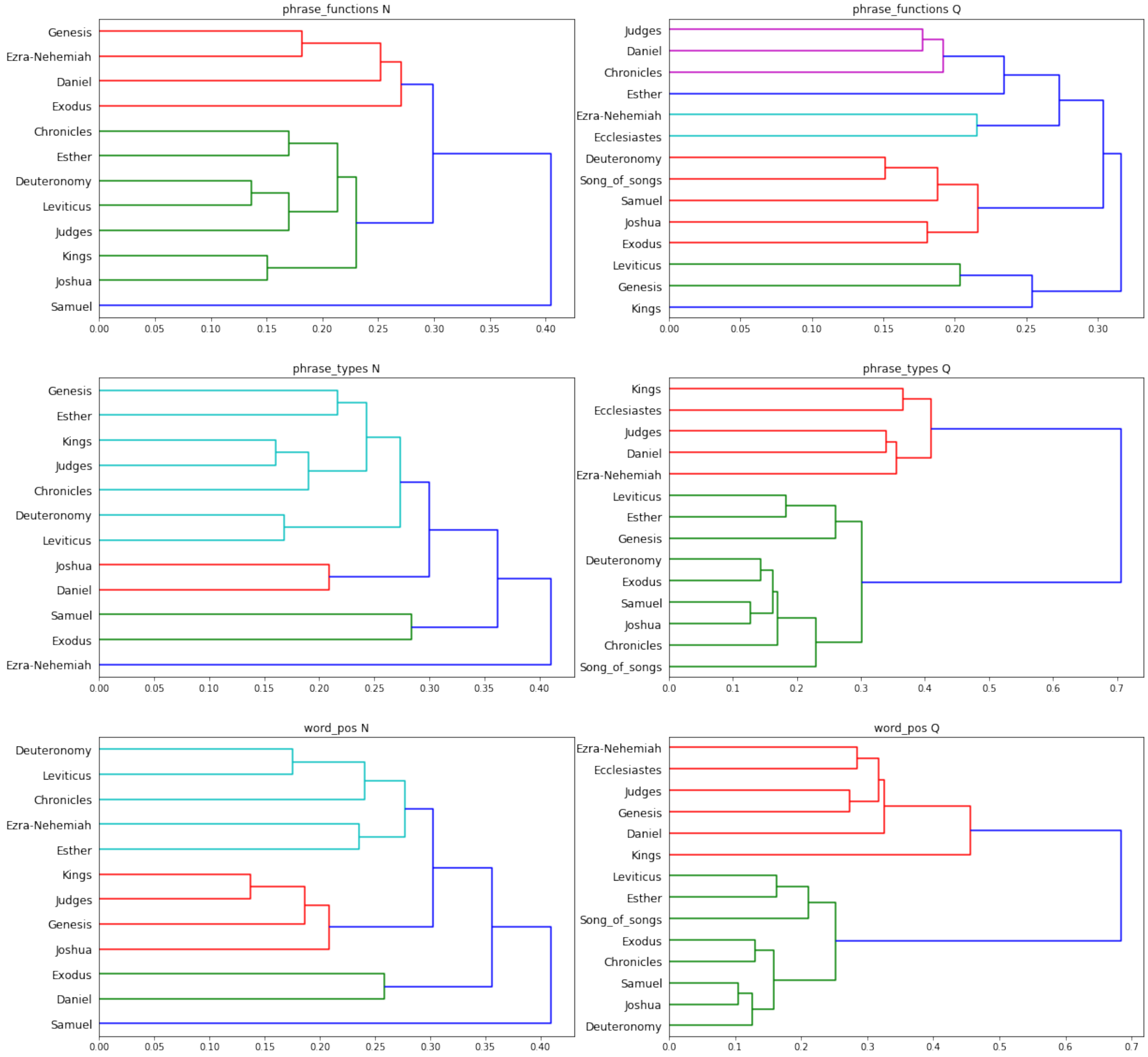

Table 1: *Book Clusters By Feature and Domain*

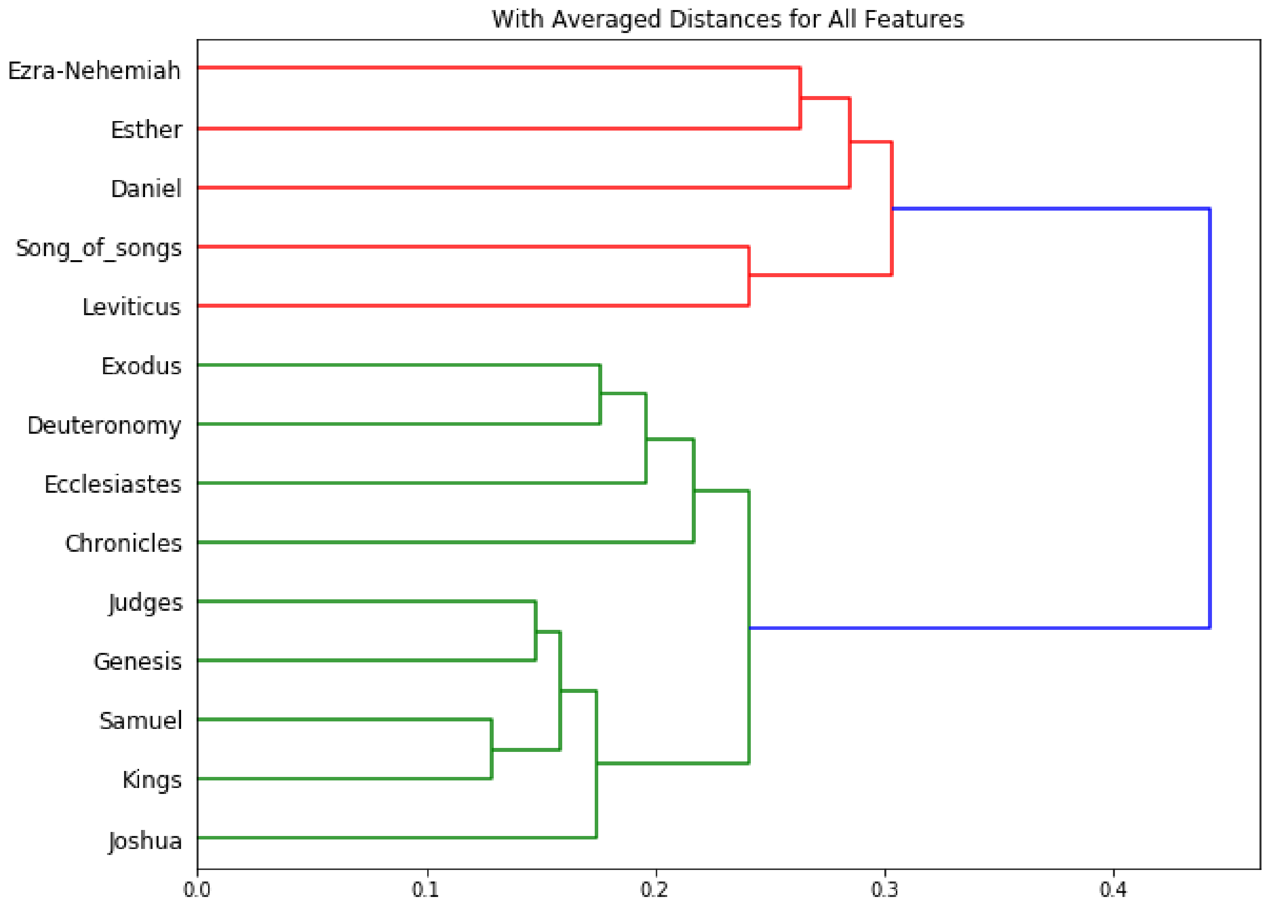

Table 2: *Book Clusters with Averaged Distances*

Chambers, J.K. and Natalie Schilling, eds. (2013). *The handbook of language variation and change*. eng. Second Edition. ISBN: 9780470659946.

Davismoon, S. and J. Eccles (2010). "Combining Musical Constraints with Markov Transition Probabilities to Improve the Generation of Creative Musical Structures". In: *Applications of Evolutionary Computation*, pp. 361–370.

Dyer, M. et al. (2006). "Markov Chain comparison". In: *Probability Surveys* 3, pp. 89–111.

Ehrensvärd, Ian (1997). "Once again: The problem of dating biblical Hebrew". In: *Scandinavian Journal of the Old Testament: An International Journal of Nordic Theology* 11, pp. 29–40.

Eskhult, Mats (2005). "Traces of Linguistic Development in Biblical Hebrew". In: *Hebrew Studies* 46, pp. 353–370.

Hornkohl, Aaron (2013). "Biblical Hewbrew: Periodization". In: *Encyclopedia of Hebrew Language and Linguistics* 1, pp. 315–325.

– (2017). "All Is Not Lost: Linguistic Periodization in the Face of Textual and Literary Pluriformity". In: *Advances in Biblical Hebrew Linguistics*, p. 53.

Hurvitz, Avi (1998). "Can Biblical texts be dated linguistically? Chronological perspectives in the historical study of Biblical Hebrew". In: *Congress Volume Oslo 1998* 80, pp. 143–160.

Leon-Garcia, A. (2008). *Probability, Statistics, and Random Processes for Electrical Engineering*. 3rd ed. Pearson, p. 700.

Niccacci, Alviero (1994). "On the Hebrew Verbal System". In: ed. by Robert D. Bergen, pp. 117–137.

Pollard, D. (2015). "Total variation distance between measures". In: URL: `http://www.stat.yale.edu/~pollard/Courses/607.spring05/handouts/Totalvar.pdf`.

Roorda, Dirk (2018). "Coding the Hebrew Bible". In: *Research Data Journal for the Humanities and Social Sciences* July.

Roorda, Dirk and Martijn Naaijer (2018). *Parallels*. `https://doi.org/10.5281/zenodo.1305272`.

Sáenz-Badillos, Angel (1993). *A History of the Hebrew Language*. Cambridge University.

Young, Ian (2005). "Biblical Texts Cannot Be Dated Linguistically". In: *Hebrew Studies* 46, pp. 341–351.

Young, Ian and Robert Rezetko (2008). *Linguistic Dating of Biblical Texts*. Vol. 1. Equinox.