Torah Codes: A Publicly Verifiable Compendium - Draft 1.2

Art Levitt

October 16, 2025

Abstract

Building on the original Witztum-Rips-Rosenberg (WRR) findings, we present a simplified approach to Torah Codes research. Our results further strengthen WRR's evidence in favor of the Torah Code hypothesis, which posits that historically or logically connected key words are encoded in close proximity in the Torah more often than expected by chance. While maintaining core principles from the original research, we have simplified topic selection, data collection and significance testing.

This makes it easier for any observer to verify the work, from its beginnings to its conclusions. For example, our data collection uses key words collected from the Torah itself. Significance testing is via a constrained subset of the standard proximity algorithms in place since WRR. The constraints limit the degrees of freedom, and also result in only the strongest codes qualifying for inclusion in our study. In many of the cases, we reduce the degrees of freedom even further via a new "crystal-growth" algorithm. That algorithm is based on the standard WRR distance measures, and it enables extensive analysis of short key words that were not accommodated by WRR.

In some of the main findings, we also highlight self-validating features that appear to be intrinsic to the code itself, whereby one code "refers" to another. This has been a gradual discovery over some years, consistently confirmed.

With the exception of only a few results, the p-values for these tables are significant below the p=.001 level.

Finally, to achieve exhaustive coverage of the simplest keyword combinations, we identify and analyze several remaining cases not covered in the initial set. Using the same pre-specified algorithms, these cases also yield significant results.

1 Introduction

The Torah Code hypothesis states that historically or logically connected key words appear as equidistant letter sequences ELSs[1] arranged more closely together in the Torah text than expected by chance.

[1] "Encodings" are identified by examining "equidistant letter sequences" (ELS) in the text more often than expected by chance. For instance, the letters f, i, t, form an ELS for the key word "fit" at the beginning of this sentence, with skip distance 3, i.e. counting every 3 letters and ignoring spaces between words.

Testing the hypothesis is essentially a statistical sampling exercise. There are two critical issues to address in such a pursuit:

1) How to fully account for successes and failures?

2) How much flexibility exists (wiggle room)?

This work addresses these issues with verifiably rigorous protocols, centering on the names of God as recorded in the Torah - a domain that we usually think of in non-scientific terms, but it actually provides a testing ground for our scientific hypothesis as few other domains could - as described in section 2.1.

Note: Tradition holds that the main Hebrew names of God[2] should not be directly pronounced or printed/discarded. Therefore we denote in this paper aleph-lamed as AL (Mighty One, protectively pronounced as Kal); we denote shin-dalet-yod as EDY (the Almighty, protectively pronounced as Shakai); we denote aleph-lamed-hey-yod-mem as ALHYM (God, protectively pronounced as Elokim; similarly for other words with the same root, such as aleph-lamed-hey-yod-kaf, your God, denoted by ALHYK), and aleph-hey-yod-hey, denoted by AHYH ("I will be", protectively pronounced as Ekyeh). The "Tetragrammaton" is considered the most sacred and all-encompassing name. We denote it as YHWH[3]. Finally the phrase, "I am HaShem", is denoted by "ANY YHWH".

In addition, displayed tables use asterisk (*) instead of hey (H) when ALHY- or YHWH appear horizontally.

2 Gold Standards for Verification

In this work, we fully address the two issues outlined in the introduction above. We detail here our approach for each of these issues.

2.1 Successes and failures - and the expanded Torah code hypothesis

Our approach stems from an expanded hypothesis that posits that genuine encodings are inherently widely verifiable. Our methodology preemptively resolves the issue of calculating a success rate, by using only widely verifiable topics as the proving ground. Candidate topics must have such foundational or historical magnitude that their significance is self-evident and beyond reasonable dispute. By definition, very few subjects meet this stringent criterion. This inherent scarcity of qualifying tests is precisely what allows for a clear and statistically robust assessment of our success rate.

One (of only a few) ways to identify such topics is to limit ourselves to ideas that are intrinsic to the Torah itself, the document that is purportedly encoded. We see that the concept of God is introduced in the very first verse, stating that God created the heavens and the earth. This reasoning guides our selection of God's names for the current study - a

[2] Especially Elokim and the Tetragrammaton.

[3] The Tetragrammaton refers to the four-letter Hebrew name of God (YHWH), often substituted with "Adonai" (Lord) or "HaShem" (The Name) when reading scripture or praying. We use "HaShem" as our protective pronunciation in this work.

normally non-scientific area that in this case serves our scientific objectives by providing a testable topic for the expanded Torah code hypothesis.

To confirm this choice, we asked three LLMs: "Can you name between 1 and 10 topics that are more important, from the written Torah's perspective, than the topic of God?" The unanimous response was that not even one such topic can be identified.

Limiting ourselves to this topic, then, automatically limits the possibility of unreported failures [4]. This constraint aligns with our thesis: selecting highly verifiable topics inherently create smaller, more easily validated sets.

2.2 Wiggle room

We eliminate the potential for wiggle room (arbitrary data adjustments that could bias results) by using broadly accessible, unambiguous data sources for all key words. In this way, our guiding principle of wide verifiability not only constrains topic selection (above), but it also constrains our selection of key words.

In the current study, the key words are explicitly discussed in the Torah itself, in a communication from God to Moses (see Exodus 6:2-6:3 presented in Figure 2) - so the Torah is both the source for the key words, and the text in which we look for encodings for these words. The names of God in these verses include only the following:

(a) ALHYM, AL, EDY, and YHWH (the Tetragrammaton). We search for these four words individually or paired (YHWH ALHYM is a natural pair, as is AL EDY[5]).

(b) A separate declaration in the same verses: "I am HaShem" (written again as the Tetragrammaton).

That is the entire data collection, virtually removing any chance for wiggle room.

2.2.1 Prioritizing Stringency and Minimizing Degrees of Freedom

The experimental design of this work follows the principle that the most compelling evidence for an effect is its persistence under the most stringent conditions. Consequently, we have intentionally minimized our degrees of freedom by using constraining limits on all tables accepted into the study[6]. The decision to use these strict constraints is not arbitrary; it is a fundamental design choice aimed at accepting only the strongest results. In conjunction with our narrowly defined search topic and word lists, these procedures avoid the "garden of forked paths" trap, lending greater confidence to the results.

[4] In fact, when we attempt to list additional realms of study that fall into this definition of "widely verifiable", we come up with only two: (a) Famous watershed events that changed the course of history (such as the Twin Towers attack, studied by Rips and Levitt in (ref)), and (b) mathematical patterns - such as another find by Professor Rips - a very basic study of YHWH occurrences at the beginning of Vayikra: in the positive direction, they occur only at skips of 5, 8, 13, 21, 34 and 55 - the Fibonacci series). In summary, the concern about unreported failures is reduced even further, given that we are hard-pressed to identify even one or two additional realms, and given the successes among the realms that we have identified.

[5] both pairs are seen together numerous times in the plain text of Torah

[6] Specifically: (a) We exclude all diagonal ELSs; (b) We allow only the most compact configurations. For repeating words within the same table, they must adhere to a minimal set of "crystal-growth" properties detailed in Appendix Section 12. For all other words in the table, they must appear as contiguous strings, not skipping rows or columns.

3 Measurement of significance

3.1 Methods borrowed from WRR:

Our methods of measurement retain and build upon key principles of the WRR precedent:

- The evidence for redundant encoding (inherent in the WRR methodology), was the motivation for our emphasis on repeating patterns, described in Appendix Section 13. However, in contrast to WRR, which mathematically sums proximities from many ELS pairs, the repeating patterns in the current work are observable in the visual tables discovered - sometimes within single tables and sometimes as repetitions between two related tables.
- In the crystal-growth algorithm, we use the primary WRR distance variables, f and f'.
- The significance of a table is defined by a relative measure of compactness, ranked among a randomized set of comparison tables.

3.2 Metrics defining table compactness:

Measurment of results is based on two metrics (built into separate software tools):

- The new crystal-growth algorithm is focused on a single key word that repeats many times as a cluster of ELSs in a single table. See for example the AL and EDY clusters in Figure reffig4. The algorithm simply identifies the largest cluster of the key word of interest in the table, surrounding the axis, and determines how often in the text such a compact arrangement occurs (width times height of the rectangular bounding box).
- 2D measurement - Like WRR, our metric is based on proximity between ELSs. Unlike WRR, and much simpler, we are dealing with a single table at a time, and we measure the proximity by simply using the area of the 2D rectangular bounding box that contains all ELSs of the table.

3.3 Ranking of metrics:

We record the metrics of interest, not only for the Torah table but also for all competing tables in our Monte Carlo runs. The Monte Carlo runs examine thousands of tables found through randomization. Each randomization is known as a "monkey text"[7]. Our result is the relative rank of the Torah table against all of these monkey texts. As described in Appendix Section todo, some of the randomization methods can be biased for or against the Torah result. The most reliable method is ERP, providing our table does not contain multiple ELSs

[7]The text can be randomized, for example by permuting words within verses or verses within chapters. Alternatively, the location of the table's main word - its axis - can be randomly shifted to other locations in the text, and the non-axis ELSs are searched in the area surrounding the moved axis. This is known as the ELS Random Placement method, first proposed by Harold Gans in 2001.

that overlap[8]. We avoid less reliable measurement situations, described is section todo, by simulating the simple cases using a technique we call "surrogate axis substitution", described in section todo. This technique also avoids overlap issues among crystal formations, such as the highly overlapping AL cluster in table todo.

3.4 The importance of anchoring:

In many cases, we establish a table by first finding a core subset of the key words of interest, and then search only in that area for a secondary subset. We can find one subset using the traditional 2D algorithm (todo explain) and the other subset using the crystal-growth algorithm. Either subset can be primary - i.e. act as an "anchor" for the other subset - the anchor pinpoints the location that we focus on, to the exclusion of all other locations in Torah, and we measure the compactness of the secondary subset that surrounds the anchor[9]. This is important because unanchored words - free-floating anywhere in Torah - are orders of magnitude less significant than anchored words. For anchored tables, we derive our measurements from the secondary subset, factoring in only the number of anchoring attempts needed to establish the table[10].

4 Results

4.1 Identifying the Names of God

The Torah narrative itself records the names of God[11], as shown in Figure 2. Focus 1 for this study: the 3 yellow-highlighted areas (4 names of God): Kal, Shakai, Tetragrammaton, Elokim (AL, EDY, YWHW, ALHYM). Focus 2 is the declaration "I am HaShem" (ANY YHWH). Figure reffigall shows these names at the top, followed by all the other related codes that spring from them. They are descibed individually below, but it is helpful to observe their relatedness and mutual reinforcement in this single figure.

[8]Reliability is determined by "sanity checking" - estimating the probability through analytical means, and comparing that result to the Monte Carlo result. Because we deal with one single table at a time, quite often this is a viable and useful technique. For the simple case of two vertical ELSs in close proximity, our sanity checks tend to align closely with the Monte Carlo reult. In this simple case, we consider one of the two ELSs to be the anchor, defined below, and the other ELS is the one to be measured. The measured ELS is positioned within an observed distance to the anchor. The analytical probability is simply the product of the frequencies of the measured ELS's letters, times the number of positions surrounding the axis that are as close or closer to the axis as the measured ELS.

[9]This is done under yet another constraint: for this study, we require that the anchor be among the top three best locations in the entire Torah for the core words.

[10]The anchor establishes, or fixes, the location of the table. For example, if the second-best anchor is used for establishing the table, the final significance is reduced by a Bonferroni factor of 2 - directly, if calculating the probability as a sanity check; or indirectly, if running a Monte Carlo experiment - in which case we ensure that the actual (or surrogate) anchor appears twice or more on average, per comparison text.

[11]The Torah's fundamental teaching that there is one God, does not conflict with the idea that the multiple names of God reflect his multiple attributes of judgement, mercy, etc

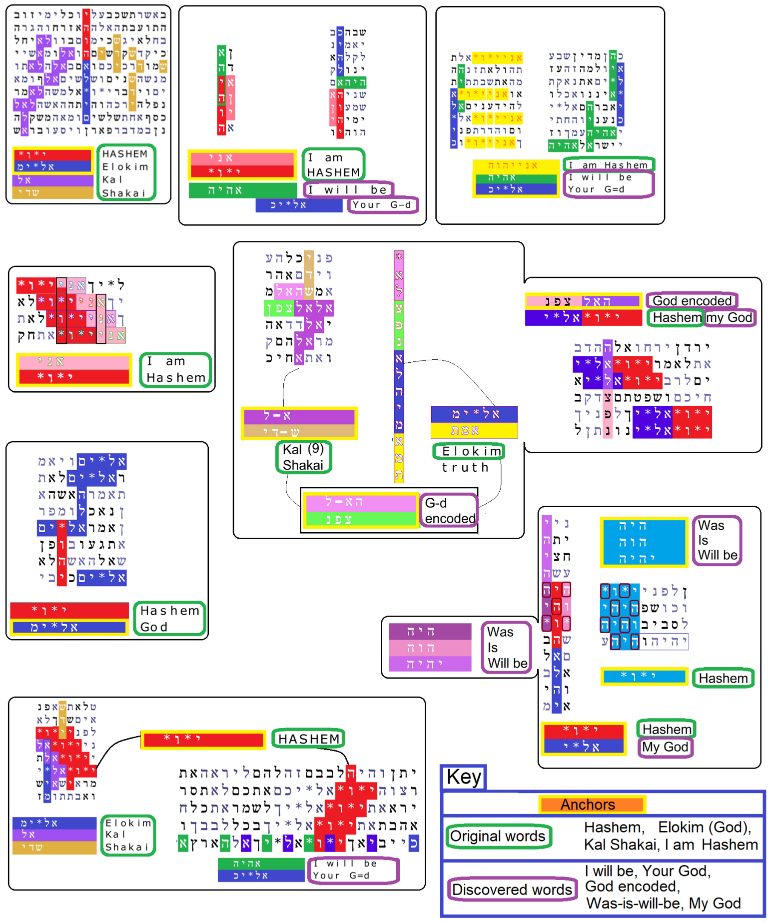


Figure 1: Summary of the encoded names of God

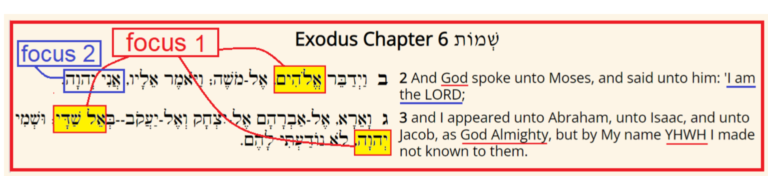


Figure 2: The main names of God as dictated to Moses

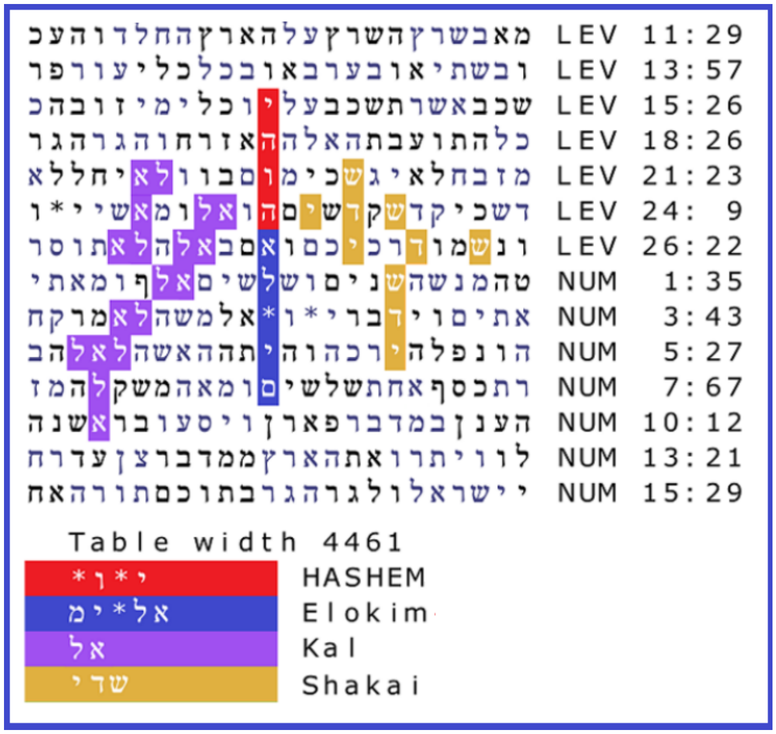


Figure 3: Best cluster in Torah for the four main names of God

4.2 Focus 1

Figure 3 shows the best cluster in Torah for the four main names of God.

We find the table's anchoring "axis" (main vertical backbone) by combining into one string, the two names that form a natural pair: the Tetragrammaton and Elokim.[12] The occurrence of this string which has the minimal skip in all of Torah (skip 4461) is our axis. The other two names (Kal and Shakai) each, independently, form clusters around this axis. This clustering appeared to be exceptional for some years, and this is now confirmed, as it is measurable using the new "crystal-growth" algorithm described in Appendix Section 12. Included in this discussion is the use of the Bonferroni correction as we apply it below[13].

Clustering of Kal: 4.2×10^{-414}.

Clustering of Shakai: 1.6×10^{-415}.

Together they are significant to the level of 6.2×10^{-816}. This is the strongest and simplest

[12]These two names appear together 21 times in the plain text of Torah, including in key passages in the story of creation.

[13]Bonferroni is a standard and statistically conservative way to account for all decisions used in experimental setup.

[14]The raw result for the clustering of Kal is significant to the level of 1.4×10^{-4} and we adjust it with a Bonferroni factor of 3 for our choice of crystal-growth parms $f = 1$ and $f' = 2$, giving the final result of 4.2×10^{-4}.

[15]The raw result for the clustering of EDY is significant to the level of 1.79×10^{-5}, and we adjust it with a Bonferroni factor of 9 for our choice of crystal-growth parms $f = 9$ and $f' = 4$, giving the final result of 1.6×10^{-4}.

[16]This combined value is the Fisher statistic calculated from the 2 individual results.

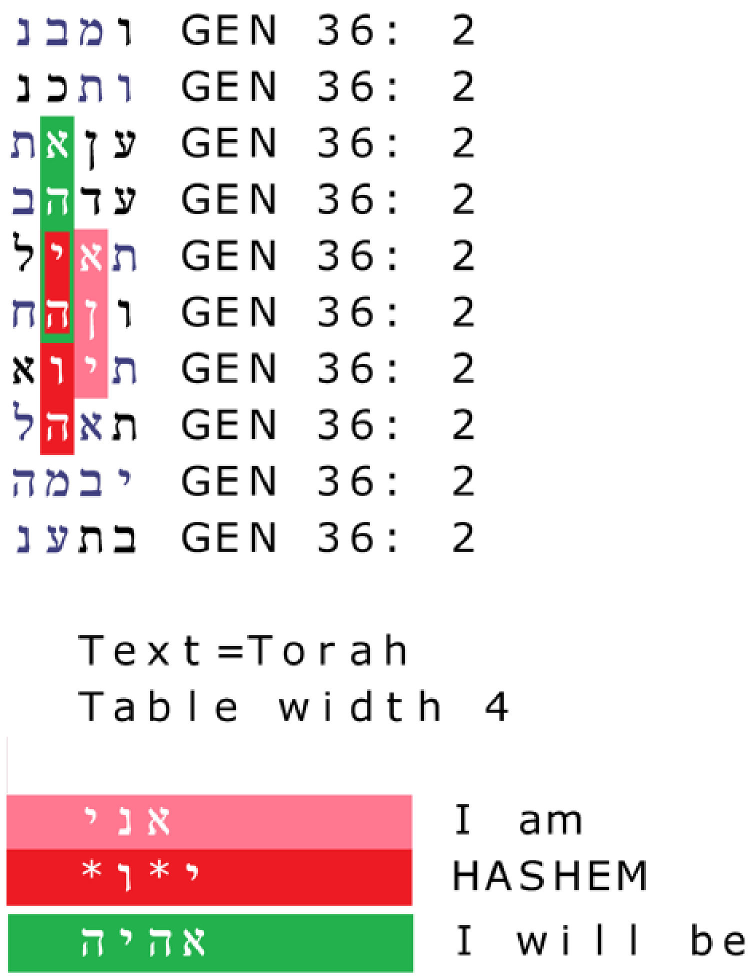


Figure 4: Best configuration of “I am HaShem” - extended by “I-WILL-BE”

Figure 5: Third-best configuration of “I am HaShem” - extended by “Your God” (adjoining “I-WILL-BE”)

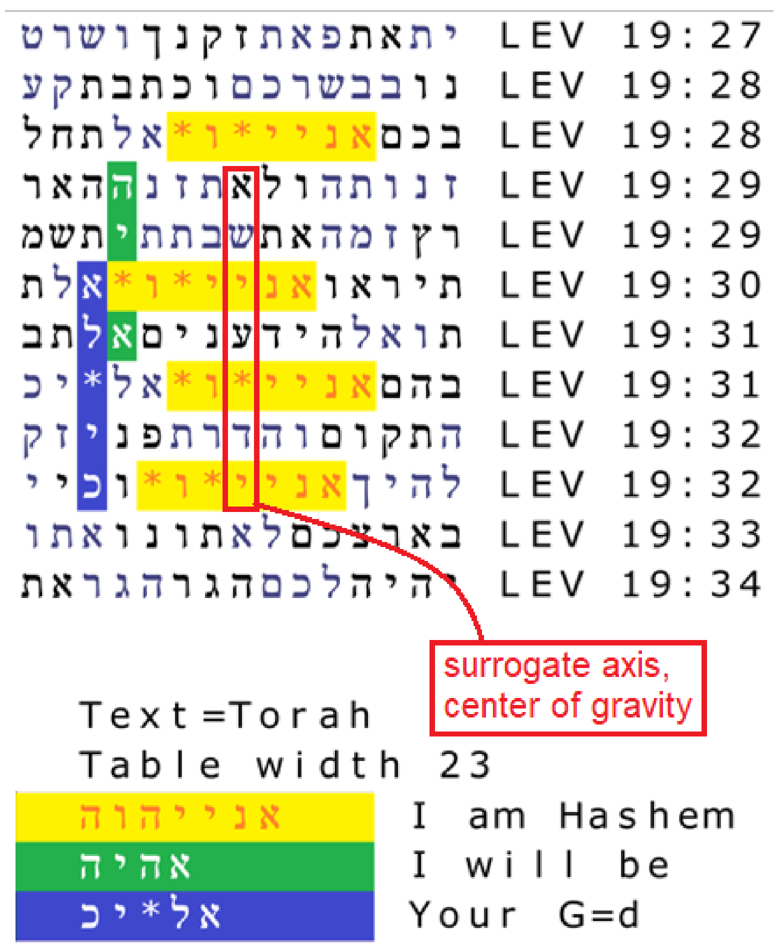


Figure 6: Offshoot code 1 of 2

כרךותקחהאשההי	EXO	2: 9
ביוםהשניוהנהש	EXO	2:13
לכהןמדיןשבעבנ	EXO	2:16
ואיולמהזהעזבת	EXO	2:20
אל*יםאתנאקתםו	EXO	2:24
נ*איננואכלויא	EXO	3: 2
*יאברהםאל*ייצ	EXO	3: 6
הכנעניוהחתיוה	EXO	3: 8
כיאהיהעמךוזהל	EXO	3:12
ניישראלאהיהשל	EXO	3:14
רפקדפקדתיאתכם	EXO	3:16
יםבמדברונזבחה	EXO	3:18

Text=Torah

Table width 195

אהיה	I will be
אל*יכ	Your G-d

Figure 7: Offshoot code 2 of 2

result in our study.

4.3 Focus 2 - Set-Up

We focus next on the summary expression in Exodus 6:2 - *"I am HaShem"* ANY YHWH. The tables in Figures 4 and 5 are the "setup" codes. They are the best and third best configurations, respectively, in all of Torah for this expression[17]. In both tables we see the adjoining word AHYH (I-WILL-BE) and we see ALHYK (YOUR-GOD) in the latter table. These two tables conform to a framework that qualifies them to be among the strongest setup codes:

1. The starting words (ANY YHWH) form the "core" for each setup code[18]. These two starting tables are among the top three in Torah, in terms of compactness and small skip.

2. The new adjoining words (AHYH, ALHYK = I will be, your God) are in the adjacent or same columns as the core words making them statistically formidable.[19].

3. The new words are close relatives to the starting words[20].

The strength of this framework for the setup codes, is one more way in which our study limits itself to the most select and verifiable patterns. The srtong framework also justifies treating these new words, AHYH and ALHYK, as *a priori* in the follow-up ("confirming") codes (*aka* "offshoot" codes) that we detail next.

4.4 Focus 2 - Confirmation

Two self-verifying offshoot tables spring from the above - offshoot 1, Figure 6 and offshoot 2, Figure 7. We find them by first looking for core formations created by the setup words AHYH and ALHYK. We do so with a single run of the 2D program, looking for all tables where the two words are vertically parallel, in the same or adjacent columns, with small skips. The top three such tables:

1. Skip 337 (not shown here) has AHYH in the same column as ALHYK, with one gap letter between, for a total area of 10.

[17]The two Hebrew words being adjacent, parallel and having the first and third smallest skips for this combination in all of Torah (skips 4 and 16 respectively).

[18]A core is a very compact formation, one of the three best such locations in Torah according to rule todo **??**, for ELS meetings of the starting words with each other. Like a single-word axis, the core anchors the table such that all related words must closely surround the core.

[19]In more detail for the word ALHYK, it contains 5 letters, and there is no better position for it, precisely extending from the final letter of YHWH. Although ALHYK consists of common letters, the product of the letter frequencies of ALHYK in Torah is still 2.35×10^{-6}, and this is diluted by only a very small number of alternative words that would be equally fitting. Note, however, that we do not need to pin down a final statistic since our goal is met by justifying the table as a set-up for a confirming code.

[20]AHYH appears in the plain text of the Torah in Exodus 3:14. This is the place in Torah where (paraphrasing) Moshe asks God, "when the people ask who sent me, what shall I tell them", to which God replies, "tell them AHYH (I-WILL-BE) sent you". The phrase YHWH ALHYK (HaShem your God) appears 230 times in the plain text of the Torah.

2. Skip 23 (offshoot 1) has AHYH and ALHYK in adjacent columns, for a total area of 12.

3. Skip 195 (offshoot 2) also has AHYH and ALHYK in adjacent columns, for a total area of 14.

We consider an offshoot table to be confirming if it contains repetitions of the words from the setup tables - namely, ANY, YHWH, AHYH, ALHYK (I am, HaShem, I will be, your God). We do not find these words surrounding the core ELSs (AHYH and ALHYK) of the first case, but we find them very strongly encoded in the other two cases, as follows.

4.4.1 Offshoot 1 and its "surrogate axis"

For offshoot 1, we find via the crystal-growing algorithm that the compact cluster of four occurrences of "ANY YHWH" (I am HaShem) occurs only one other time in Torah with such compactness. There is also one case with 5 occurrences. Therefore we have a maximum of 2 other clusters in Torah that are comparable to our observed case. That means that our cluster is as rare as a single long vertical ELS that appears only 3 times in Torah. We can set up a "surrogate" axis with an appropriate skip range such that there are 3 occurrences of the surrogate axis in that skip range[21]. Accordingly, we choose the string EYIHDYA - in the red box of Figure 6, allowing skips 2 to 3000. This string appears at the approximate center of gravity of the "ANY YHWH" cluster[22].

The Monte Carlo run randomly moves the surrogate axis to other starting points and skips (using skips 2 to 3000). The question that this run answers is how often we find both AHYH and ALHYK so close to the randomly placed surrogate axis - the measurement being the rectangular area of the bounding box that includes the axis along with AHYH and ALHYK[23]. The result is that we find 18 competing tables in 2.6M monkey text attempts, which is 6.9×10^{-6}.

4.4.2 Offshoot 2

For offshoot 2, the anchoring core is the side by side parallel AHYH with ALHYK (I will be, your God) on the right side. We are interested in the cluster of three occurrences of AHYH (left side) near this core. We find via the crystal-growing algorithm that such a compact cluster of three occurrences of AHYH (I-WILL-BE) does not occur at all in 50,000 randomly placed attempts up to skip 10,000 (and there were also no competing clusters in this set with *more* than three ELS occurrences). Therefore we use a long surrogate axis, todo-spell-it-out, in this case substituting that axis for the adjacent words AHYH and ALHYK. The crystal growth algorithm measures the cluster formed from the other 3 ELSs for AHYH, with result todo.

[21] This is following the recipe detailed in category 3 of special cases, Appendix Section 16.

[22] Off by 1 column, giving monkey texts a small advantage since they have an extra column in which to find non-axis words.

[23] We add the rectangular sub-area of (AHYH and ALHYK together) to the total bounding box area, in order to reward those competitors that have a compact arrangement of this subcluster.

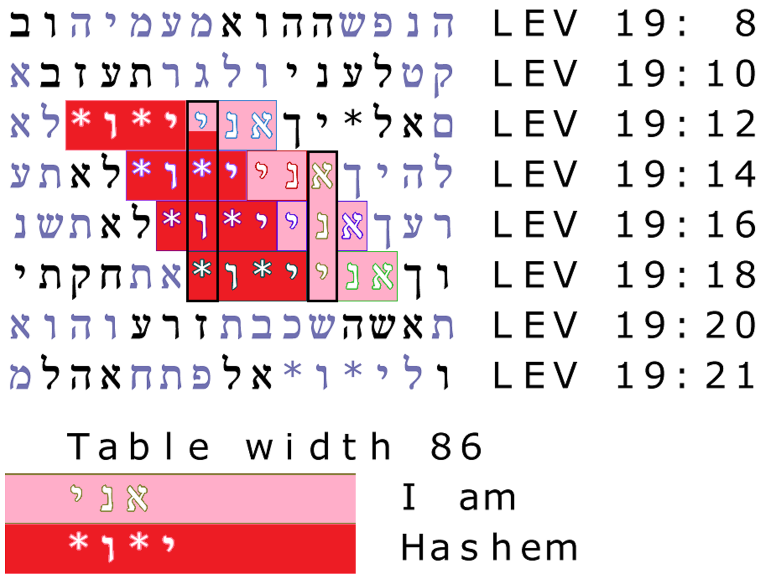


Figure 8: Additional case 1

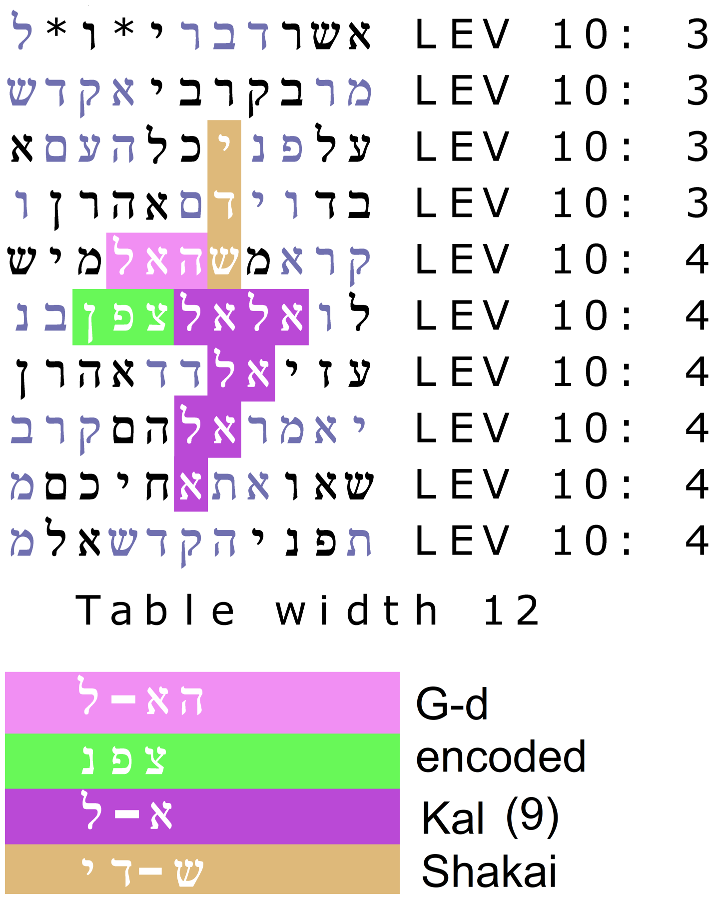


Figure 9: Additional case 2

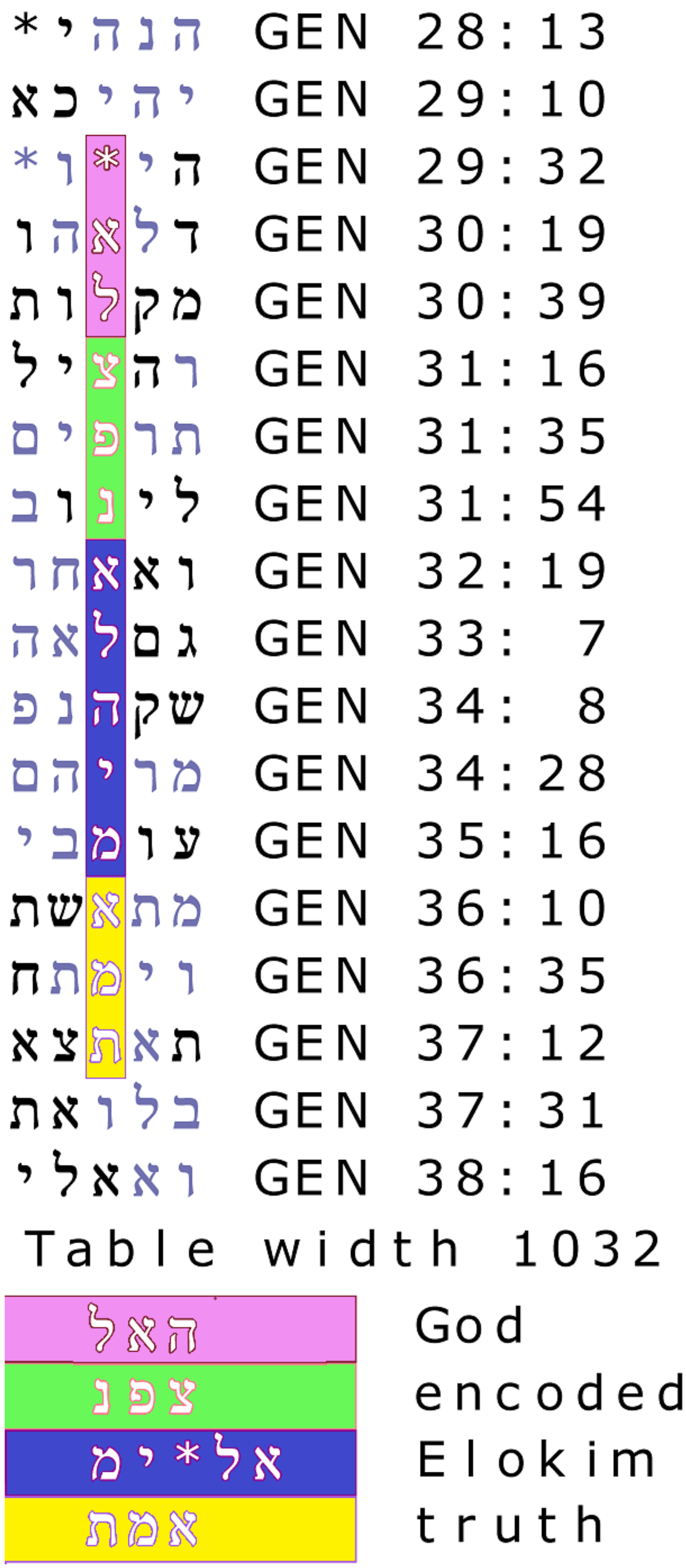


Figure 10: Code summarizing the entire topic, from Doron Witztum

ואשתוולאיתבשש	GEN	2:25
ערוםמכלחיתהשד	GEN	3: 1
ו*אל*יםויאמרא	GEN	3: 1
אמראל*יםלאתאכ	GEN	3: 1
ןותאמרהאשהאלה	GEN	3: 1
הגןנאכלומפריה	GEN	3: 2
הגןאמראל*יםלא	GEN	3: 3
ולאתגעובופןתמ	GEN	3: 3
נחשאלהאשהלאמו	GEN	3: 4
דעאל*יםכיביום	GEN	3: 5
ונפקחועיניכםו	GEN	3: 5
ימידעיטובורעו	GEN	3: 5

Table width 22

י*ן*	Hashem
אל*ימ	God

Figure 11: Additional case 3

5 Additional validation

Here is a summary of the known tables for the names of God, covered above:

For the four focus 1 original key words, we covered "pair1" (AL EDY) with "pair2" (YHWH ALHYM); and we covered "pair3" (ANY YHWH) [the focus 2 original words] with "pair4" (AHYH, ALHYK) [offshoot words].

We also knew about, and presented above, the table pairing AHYH and ALHYK - the individual words split out from pair4.

The untried combinations prior to our study are analogous this pair4 split-out: the individual words within pair1, pair2 and pair3 i.e. AL with EDY, YHWH with ALHYM and ANY with YHWH. "Cross-pollinating" (for example, trying ALHYM with AHYH) would lead to many more forking paths in the garden. Therefore we have a well-defined basic set of tests and we intentionally do not consider additiional, less basic combinations.

We only recently developed the new crystal-growth algorithm and surrogate axis method. We were therefore very interested in whether these same methods would yield significant results using these untried combinations.

As demonstrated below, we do find significant results for these new combinations, both directly and by examining offshoot tables for them. This strengthens the evidence for the Torah being an encoded document, since these new findings all spring from our small, closed set of key words.

Following are the new results:

1. Additional case 1: ANY near YHWH.

   This case (Figure 8) is neutral because the cluster of ANY is really formed from the YHWH cluster that lies horizontally next to it. Therefore the ANY cluster is dependent on the horizontal stack of YHWH occurrences. Also both the vertical ANY and the vertical YHWH are dependent on these horizontal strings. Despite these dependencies, preventing us from measuring the clusters, the fact of their existence is interesting. On balance, this case neither adds to nor detracts from the other evidence.

2. Additional case 2: א-לnear ש-ד-י

   Figure 9 is the most densely packed configuration of AL ELSs next to EDY in the Torah. @@@ throw much of this into an appendix. Out of more than 8000 EDY axes with small skip (11 to 120[24]), this table is the winner (the closest competitors have skips of 25 and -42; in these cases the AL clusters are smaller but they are also further removed from their EDY axis). Also note: due to the small skips producing narrow tables, we surround the central letter of the axis with 5 columns on each side and 5 rows above and below, yielding a grid size of 11 by 11 instead of the default 21 by 21; and we use the tightest crystal-growth parameters $f = 1$ and $f' = 0$. This winning table may or may not be significant compared to tables that might be found in other texts, but the important point is that it our table is the best location *in the Torah* for such a small-grid AL cluster. Being the best in Torah (and using *a priori* core words) qualifies this configuration of [AL cluster and EDY], to be an anchor for the two connected and

[24] We tested skips up to 10 times the observed skip of the first winning table - see @@@ discussion of skip versus area

adjacent horizontal words, HAL CPN (God encoded). These discovered words are not *a priori* but they immediately connect to a prior code (Figure 10), which Doron Witztum discovered in the early years of his research. In the decades since then, it has remained one of a select class of showcase examples. Its meaning, "God encoded Elokim, truth", is highly relevant to our study of God's names, which is why at the outset - prior to discovering Figure 9 - we considered using Witztum's code as the opening figure for our study.

Given this prior code, and the short list of names of God, one of the most natural offshoot ideas connected to the early result, would be to search for "God encoded א-ל ש-ד-י", which is precisely Figure 9. The raw probability of finding the two horizontal words ("God encoded") so close to the anchor configuration Kal Shakai, is 1.58×10^{-425}. The missing factor that lowers significance from this raw number, is the number of equally surprising (to HAL and CPN) connected words that may have arisen. This factor is generally not possible to determine, but in this case it is at least demonstrably low, given the precedent of Witztum's table. On balance, this case adds to the total evidence of an encoded Torah, qualitatively if not completely quantitatively. In addition, it helps validate the crystal-growing concept for 2-letter words.

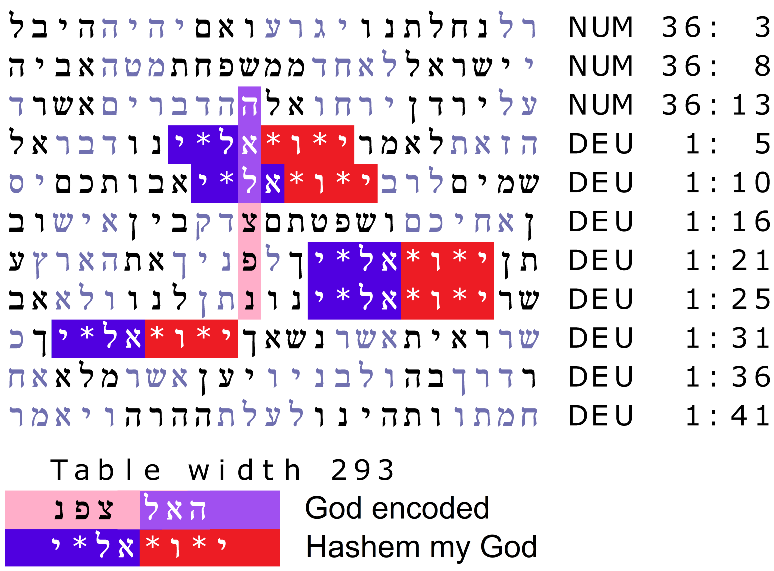


Figure 12: Additional case 4

3. Additional case 3: Tetragrammaton near an א-ל-ה-י-ם cluster

   This case (Figure 11) is the best cluster in Torah for ALHYM, as discovered by the crystal-growth algorithm. The proximity of the Tetragrammaton to this cluster is

[25] The strength of this result is due to the fact that CPN appears only twice in the plain text of Torah. We obtained the result using the 2D program, and a surrogate axis substituting for the anchor configuration - this method is showcased as an example in Appendix section todo.

significant at the level of 2.0×10^{-2} using the magnif program[26].

4. Additional case 4: HAL encoded the Tetragrammaton

   This is a direct offshoot table from "Additional case 2" above. Discussion of that code linked the concept that HAL encoded AL EDY, newly discovered, with the former concept that HAL encoded ALHYM. The obvious missing link (recalling that our set consists of ALHYM, YHWH (the Tetragrammaton), AL, EDY) would be "HAL encoded the Tetragrammaton". We in fact find this concept in Figure 12. It starts with the vertical HAL CPN (God encoded) - second minimal skip in all of Torah. The 5 surrounding horizontal occurrences of YHWH are close by but also very common. What is interesting, however, is that all of these YHWH occurences are extended by א-ל-ה-י(my God)[27]. The upper 4 are clustered very closely, with the crystal-growth algorithm yielding a p-value of 9.8×10^{-5}.

5. Additional case 5: God was, is, will be - part 1

   The next table, Figure 13, is a well-known showcased finding from Professor Rips, but it did not quite qualify for early drafts of this report[28]. Figure 13 was initially excluded because it was not an offshoot of our known cases. It is now a direct offshoot of Figure 12. It starts with the same expression that we analyzed as a cluster in case 4 - YHWH ALHY (HaShem my God) - specifically it starts with the minimal skip of this expression. The table is "frozen" in place around this axis, making the positions of the adjacent words, WAS, IS and WILL-BE, highly significant. These three ELSs, in an important sense, define the Tetragrammaton[29]. We obtain a raw p-value of 2.3×10^{-8} for the extraordinary strength of the compact meeting between the two elements of the table:

   (a) Element 1: The axis expression "HaShem my God" י-ה-ו-הא-ל-ה-י

   (b) Element 2: The word set WAS, IS and WILL-BE

   Element 1 is a fundamental expression involving the Tetragrammaton. Element 2 is a direct "description" of the Tetragrammaton.

   We allow for 10 equally strong alternatives for each element, which gives us a total reduction in significance of 100, a resulting "working" p-value of 2.3×10^{-6}. We can not obtain a final value because it is not possible to pin down the reducing factors precisely, but in our case both factors are demonstrably very low[30]. But because we

[26] We obtained this by substituting a surrogate axis, MAWALHYM, in place of the ALHYM cluster. This axis is the vertical string of letters in the exact center of the table.

[27] Some of them have additional letters, such as א-ל-ה-י-כ(your God) or א-ל-ה-י-נ-ו(our God) but all 5 have the first 4 extended letters in common (my God).

[28] it was one of only 3 or 4 tables we initially considered and rejected

[29] The Mishnah Breurah 5:2 discusses that a person's focus should be precisely WAS, IS and WILL-BE when their eyes encounter the Tetragrammaton in study or prayer

[30] Element 1, the axis, YHWH ALHY, is one of only a few expressions involving the Tetragrammaton and a "close relative" of ALHYM, which are the starting words for our study. Element 2, the word set WAS, IS and WILL-BE, have few if any documented alternatives so closely related to YHWH. Note that we also

ארץוידב	NUM	14:38
סכיהעמל	NUM	14:42
לבניישר	NUM	15: 2
יעיתההי	NUM	15: 5
מןחציהה	NUM	15: 9
םועשהאש	NUM	15:14
הוהיהבא	NUM	15:18
רצוהי*ן	NUM	15:23
יי*ן*על	NUM	15:25
לעשהבשג	NUM	15:29
ציםאלמש	NUM	15:33
בראלבני	NUM	15:38
תיוהיית	NUM	15:40
קראימוע	NUM	16: 2
באליוזא	NUM	16: 5
בדתמשכן	NUM	16: 9

Table width 186

י*ן*	Hashem
אל*י	My God
היה	Was
הוה	Is
יהיה	Will be

Figure 13: Case 4 - offshoot 1

Figure 14: Case 4 - offshoot 2

can not entirely remove the above aspect of subjectivity, we do not claim that the working p-value is final. We simply conclude, like case 2, that the current code adds to the total evidence of an encoded Torah, qualitatively if not completely quantitatively.

This code also acts as a setup code for case 6 below.

6. Additional case 6: God was, is, will be - part 2

   Figure 14, is a direct offshoot of case 5, using the Tetragrammaton and WAS, IS and WILL-BE. Throw into appendix@@@. However, there is significant dependency on the horizontal words, so we measure the one aspect of this result which we can confidently untangle from the dependencies - the 1D portion is the second-best such configuration in Torah and the 2D portion is the best in Torah. We can measure simply the probability that they would occur in the same location. To demonstrate that just one aspect of the 2D find is the best in Torah, we see via the crystal-growing algorithm that 6 occurrences of HYH packed so tightly next to YHWH is the best in Torah. We can now create a surrogate axis for this 2D portion of the code. For the 1D portion, we use only the non-overlapping letters YHYHW - this string occurs 24 times in Torah

create conditions that are somewhat biased in favor of the monkey texts. For example, we use a surrogate axis, YXIHWHE (right column of the Figure), which allows each monkey attempt to use 1-2 columns on either side of this randomly relocated axis, resulting in more fillable positions surrounding this axis than we had in the Torah.

with skip 1 or minus 1. Measuring this string to the surrogate axis representing the 4 by 4 2D grid, results in a p-value of 2.5×10^{-4}. [31]

6 Additional Insights

It is useful to review some additional points about how we verified these results, and why that process was simpler than in previous studies.

6.1 Comparing the old and new methods of measurement

While following the basic principles of WRR we also simplified and extended some WRR methods:

- We are dealing with only a handful of key words, the main names of God, compared to hundreds of combinations of dates and appellations of Rabbis in WRR.
- The words we use are spelled out in the Torah itself, compared to the WRR method of extracting its key words from an encyclopedia, guided by a human expert. Follow-up work for WRR evolved to be rule-based, and multiple data sets continued to succeed with this fixed approach, but it is still resource-demanding to verify in all its details, compared to the current work.
- To estimate significance, WRR used accumulated scores, which combined dozens of measurements for just one keyword pairing (one appellation and one date), spread over many small tables - and this exercise had to be repeated for hundreds of such pairings. In contrast, our approach analyzes a handful of visual tables.
- Short words (of 2-4 letters) were excluded by WRR. We include them. For a 2-letter word, we require it to repeat in a table at least 3 times in order to consider that it is even present (see todo Constraint 1 in Appendix Section 16).

6.2 Avoidance of Unnecessary Complexity

A few side effects of some of our randomization methods can yield inconsistent results, as revealed when comparing results among the various methods. A description of these problems and our solutions are detailed in Appendix Section 14, mainly for future reference. In short, we avoid these potential problems in the current study by carefully noting when they could occur and implementing workarounds, such as the surrogate axis concept (todo

[31] Beyond this simple result, this is a case of near-maximal density, which typically has a rarity almost on the same scale as long phrases. In both cases, the letters of interest can be arranged in very limited positions. There is also a diagonal symmetry aspect, especially for the letter H. Neither the density nor the symmetry is factored into the statistics, but both are especially noteworthy because they both manifest in the 9 central letters of Figure 13 (3 by 3 area with all highlighted letters), which establishes these patterns as setups. Both patterns are then confirmed in the 4 by 4 area of Figure 14. Also interesting in the latter table, and contributing to its maximal density, is that all of the words WAS, IS, WILL-BE, and Tetragrammaton occur multiple times within the dense 2D portion of the table and also in the dense 1D portion of the table.

see Appendix section). This addresses situations where the standard measurement does not result in a level playing field between the original result and its competitors.

7 Discussion

todo - improve wording for this section

7.1 Another extension to the Torah Code hypothesis - todo - omit this?

In addition to the new claim that the codes are intentionally verifiable, we propose one other claim, which is that the codes are intentionally designed to appear real or imagined, depending on the observer. We have no simple way to test this extension to the hypothesis, except to make some observations that may support it. First, the discipline straddles the fields of statistics and linguistics. These are mathematical disciplines, yet there are enough variations in methodologies so that many codes lie on the boundary between coincidental and intentional. Related observations:

- Very smart people on both sides of the argument.
- Moses could not see His face.
- Likewise, an absolute proof might be overwhelming.

7.2 A methodological concern

Some of our findings involve confirming codes that branch off from initial setup codes. This can raise a concern: shouldn't control texts have the same opportunity to generate their own setup codes? However, that would present a practical challenge - setup codes are inherently rare phenomena that would require extraordinary effort to locate within control texts. To address this issue, we provide the setup words to control texts "for free", requiring only that the control texts form meaningful tables around these predetermined terms - because after all, we are only measuring the significance of the Torah's offshoot table, not the setup table.

This concern is further resolved by simple experiments that test thousands of random word sets: both the Torah and the control texts produce random results. This demonstrates that the ability to form significant repeating patterns is not simply a function of the search methodology, but depends on the specific words being examined. The control texts' failure to generate significant patterns even for setup tables demonstrates that the observed phenomena in the Torah are not artifacts of the approach.

OOD versus OOP is a second concern? Show that either order is strong - e.g. any yhwh skip 23

7.3 Varied statistical interpretations

It is also worth noting a systematic issue with statistical practice - interpretation is not universally standardized, even among credentialed statisticians. Consider a thought experiment: predicting an encounter with a specific person at a specific time after a decade without contact, but instead encountering his twin whom you also haven't seen for ten years. Some would argue this near-miss substantially diminishes the statistical significance, while others recognize it remains an extraordinary outcome - differing from the exact prediction in only one parameter, yet still defying reasonable odds. These or other statistical debates can also result from the fact that scientists are not always scientific (todo references). Some of these debates are useful and some are unproductive, in particular those that are inconsequential to the conclusion, and yet create unnecessary distractions along the way.

7.4 The overal phenomenon versus individual results

Our study uses several safeguards to ensure rigor. It is important to be aware that most informally reported code results follow few — if any — of these safeguards. Therefore, for many such findings, similar patterns can be found in any text of comparable length to the Torah, leaving the impression that the overall phenomenon is also comparable across texts. As we show in this study, gathering evidence for the existence of codes is an entirely different pursuit - vastly more comprehensive - than finding individual codes that look interesting. The former is exacting and can have deep implications; the latter can be ambiguous at best and misleading at worst. There are actually some results published with statistical claims and calculations that give an appearance of legitimacy but they contain logical flaws and lack of true rigor. In short, it is best to ignore codes produced without formal protocols, which unfortunately tend to be popular on social media.

7.5 The Synergy between the New and Old Cases

When we survey all of the known and new cases together, the reinforcement of the repeating patterns grows stronger, and the possibility of significant errors in the methodology grows weaker. Driving this point even further, we can apply the new methods to known cases not considered above - see Figures 15 and 16. These tables have been known for years but they can be freshly analyzed with the new crystal growth and surrogate axis methodologies. They can be re-identified by starting with the most densely packed configurations of the Tetragrammaton, using the crystal growth algorithm. In the first case, the other words from our standard vocabulary, ALHYM, AL and EDY, cluster in a near-maximal configuration. In the second case, the best 1D meeting of our remaining vocabulary words, AHYH and ALHYK, crosses the Tetragrammaton cluster. When analyzed using appropriate surrogate axes, both of these cases have high significance comparable to several other results reported here, and they further reinforce the usefulness and correctness of the algorithms.

7.6 Parallel lines of research

Finally, this work is only one approach to bostering the conclusions of WRR. See Gans and Witztum (todo references) for other work that reinforces WRR more directly - very closely replicating its methodology on multiple sets of related data that were established at the time that WRR was first published and debated.

8 Conclusion

Most of the significance levels of our tests meet or appreciably exceed the 10^{-3} level. By obtaining such results under our constrained conditions, the original WRR claims are further supported. todo - list the results in summary form.

When we consider the completeness of tests within our uniquely verifiable topic, this conclusion is further strengthened.

not as robust but unexplainable: rls? fibo? codes true? Show pix!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! does not detract from names of hashem

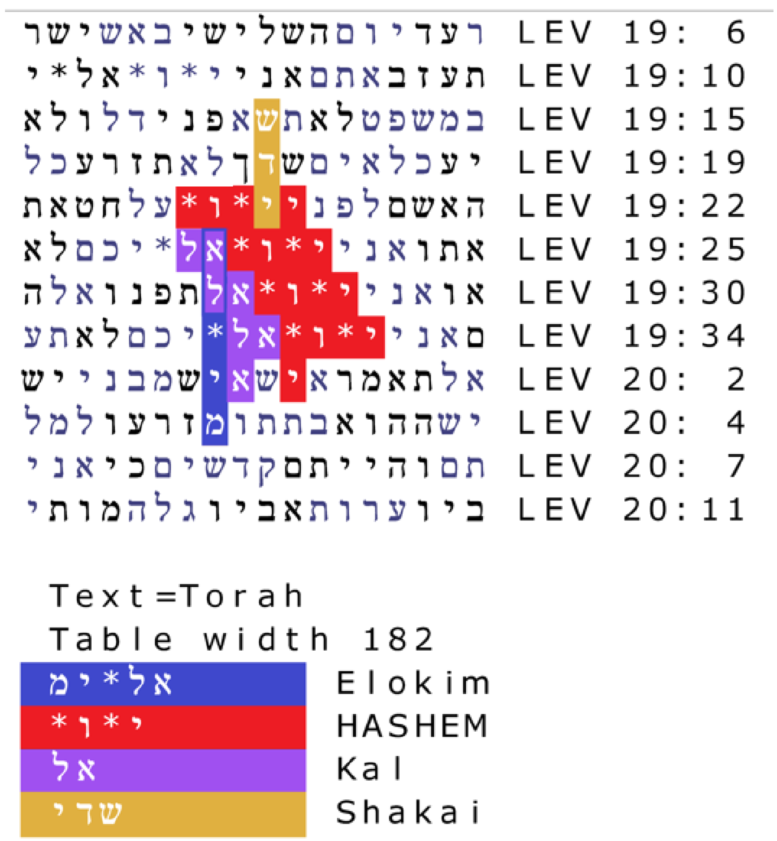


Figure 15: Known case 1, re-identified

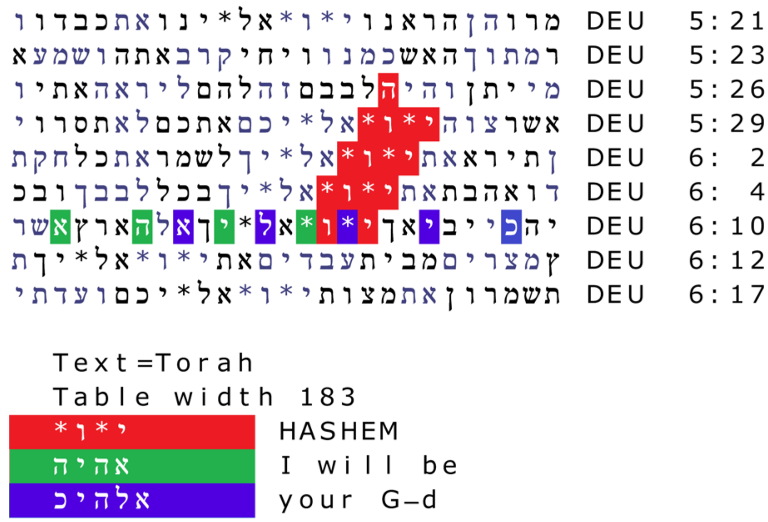


Figure 16: Known case 2, re-identified

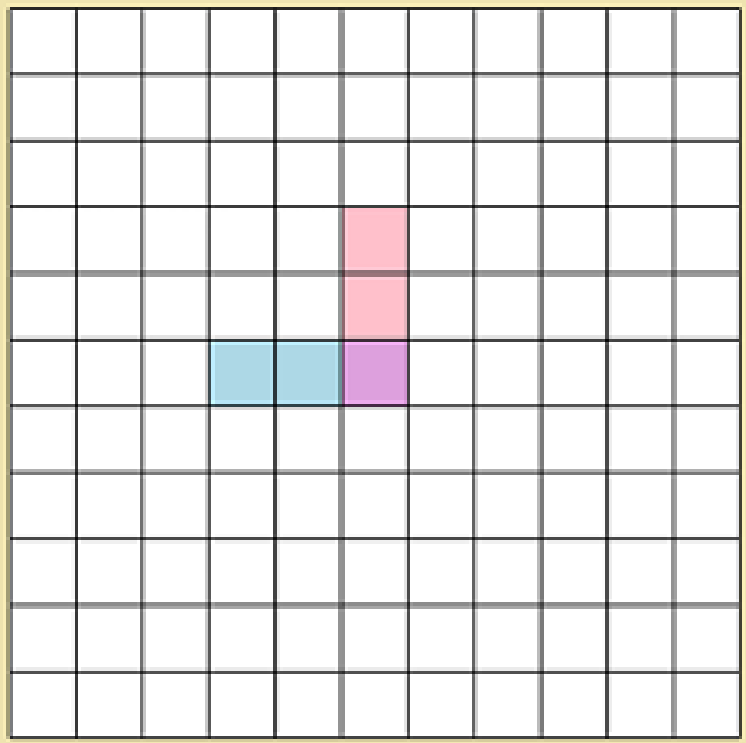
9 Acknowledgements

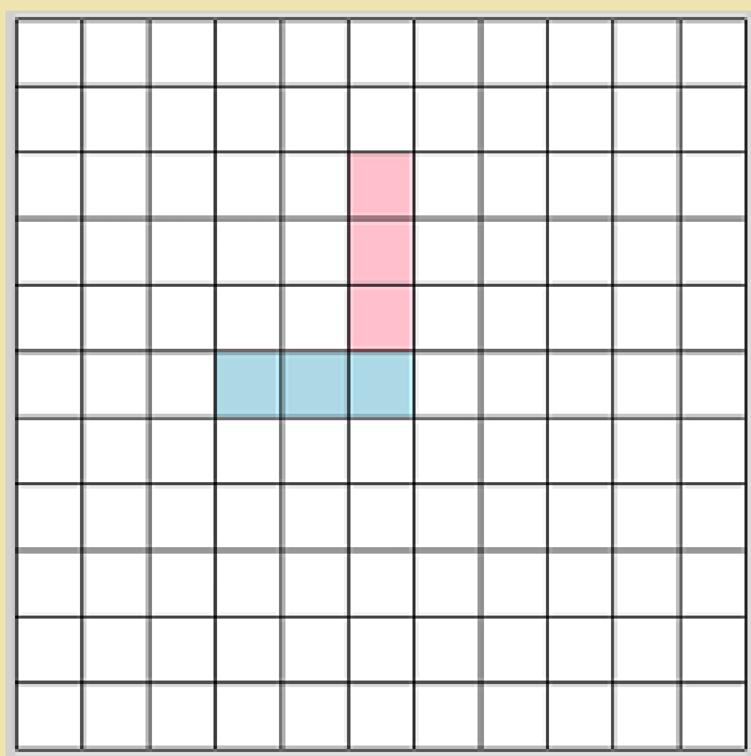
I wish to thank Professor Eliyahu Rips (1948-2024), Doron Witztum and Harold Gans. The pioneering work of Rips and Witztum provided a guiding light for all follow-on efforts. The strengthening of the pursuit by Gans, and his valuable insights that grew out of that, helped shape the current work. The key idea of a core set of words being the anchor for so much more, is mutually reinforced by the decades of astute research of Alex Rotenberg. Further reinforcement comes from the dedicated efforts of Rabbi Ephraim Roitman and Chaim Meir Pollak, who have helped to find or analyze several of the examples reviewed here. They have also broken new ground with growing collections of codes yielding similar results for other topics. Moshe Belaich, who discovered yet more collections, has also verified many of the current results by comparing them to similar protocols that he set up with his own software. Haim Chanun continues to teach and learn all-important details that have added yet more rigor. Most importantly, I came to understand the fundamental principles by studying under Rips' guidance from 1998 to 2024. This research fits precisely into the framework he established, and is dedicated to his memory.

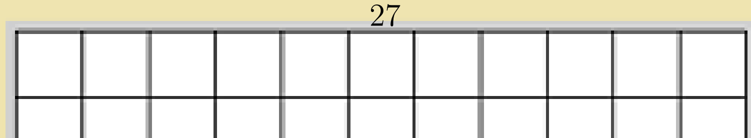
10 Appendix - Parameters and Constraints, and Heuristics

11 Summary of Results

Part 1 - figure todo







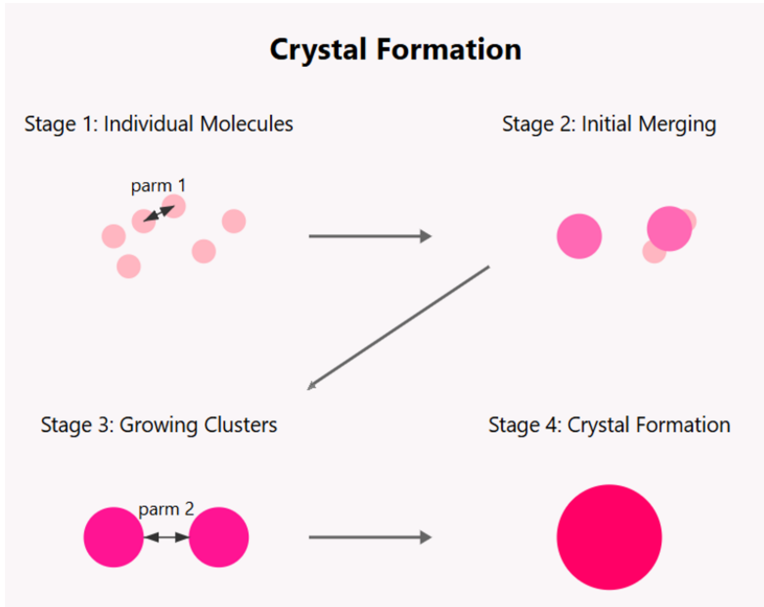


Figure 18: Simplified View of Crystal Growth

12 Appendix - Crystal-Growth Algorithm

Algorithm Summary

We use the following "crystal-growth" algorithm to determine the significance of clusters of short ELSs such as the AL and EDY clusters in figure 3.

This algorithm simulates a very simple model of crystal growth in order to identify groups of ELSs that form a unified cluster. We start with the center point of the axis ELS in our table of interest. The algorithm does not look outside a fixed grid size surrounding this center letter. It is limited to searching 10 rows above, 10 rows below, 10 columns to the right and 10 columns to the left, i.e. a grid that is 21 by 21 letters[32]. In such a grid, the cluster of ELSs for the key word of interest, is discovered one small part at a time - by our eyes or much more practically by the software - that is, one corner of it may grow into a small sub-cluster as the software finds nearby neighbors. In another corner of the table, depending on the configuration and the order of traversing the ELSs, the software may notice another sub-cluster. As the two sub-clusters grow toward each other, they may merge if they approach close enough. This is similar to crystal growth found in natural systems - over-simplified, but suitable for our purposes. Our measure of significance of the final accumulated cluster within the grid, is the relative size of that cluster compared to thousands of clusters that we find in control grids (we form these control grids by randomizing the location and skip that describes the axis ELS - its center point becomes the center point of our grid).

This whole process can be completed based on just two WRR parameters - the intra-word

[32] This uses the precedent established in 2007 when we first discovered how anchoring and clustering seemed to work: the three Patriarchs (Abraham, Isaac and Jacob) formed a cluster which became the anchor for several related results - among them, Sarah, the wife of Abraham, formed its own cluster of 7 ELSs around the center point of the Abraham ELS, in a 21 by 21 grid

distance and the inter-word distance, denoted by f and f' respectively (the Pythagorean squared distance is used). That is, a new ELS is accepted into the cluster if the distance between its letters is less than or equal to f, and the distance from the closest letters of ELS_1 and ELS_2 is less than or equal to f' (also recall that we accept only vertical and horizontal ELSs in this study).

Estimates of f and f' (Pending Calibration)

The following is based on our chosen window size of 21 by 21 (10 rows above and below the table's central letter, and 10 columns to the right and left of that letter).

We assign reasonable estimates for f and f', pending a more formal calibration procedure. The estimates are assigned based on expected frequencies of ELS and plain text occurrences. The estimates ensure that a key word which occurs many times (as a vertical ELS and/or horizontally, including skip 1) needs tighter f and f' values in order to result in only rarely occuring hits.

Case 1: For a 2-letter word which consists only of common letters (not G, D, Z, X, O, K, S, I, P, C, Q).

A) For f, the intra-word distance: we analyze the pool of ELSs available for a fixed table width when $f = 1$:

a) Vertical: for this case, there are typically between 3000 and 5000 vertical occurrences with row skip 1 (i.e. contiguous in the column) for any given fixed skip.

b) Horizontal: there are typically between 6000 and 9000 skip 1 or -1 occurrences for such a common 2-letter word.

The total pool of ELSs in the entire text is the sum of these, i.e. typically more than 10,000 total contiguous occurrences horizontally and vertically. Therefore limiting ourselves to $f = 1$ for case 1, limits the resulting clusters to the rare, strong cases of interest.

B) For f', the inter-word distance:

Figure 17 depicts the three closest kinds of meetings between 2 perpendicular ELSs such that the two ELSs could be considered "touching" - i.e. overlapping, side-by-side, or catty-corner. Since f' is the Pythagorean squared distance between the closest letters of ELS 1 and ELS 2, f' has a choice of 3 values - 0, 1 and 2 respectively - in order to comply with the "touching" constraint.

Summary for case 1: f must be 1 and f' must be 0, 1 or 2. This is a total of 3 possible combinations for our two cluster-growing parameters.

Case 2: For 3 or 4 letter words that do not occur often in the plain text. todo insert details.

Case 3: For 4-letter-plus words that occur commonly in the plain text. todo insert details.

Summary: For longer words, we allow f' to be 2, 4 or 9 (2 is the catty-corner case, while 4 and 9 allow for separation of 2 or 3 spaces respectively, in a horizontal or vertical direction.

These limitations imposed by f and f' yield not only rarely occurring clusters, but also clusters that are limited in size, so that the total grid size limitation suits most real world scenarios. In summary, the above grid size and parameter combinations meet our goal of focusing on only the most significant finds.

Bonferroni Adjustment

For whatever case we have at hand, we run the algorithm n times, where n is the number of calibrated choices for f multiplied by the number of calibrated choices for f'. We then take the strongest result from these multiple runs and multiply it by a correction factor of

n (following the Bonferroni approach). This approach accounts for all our attempts in a very straightforward way, but is overly conservative. The reason is that many of these runs produce similar outcomes because they are closely related to each other, and our correction method does not recognize these dependencies. A more sophisticated, less conservative approach could be proposed for future work, but for the current work we prefer simplicity.

13 Appendix - Repeating Patterns

In this appendix we summarize - and emphasize - the vital role that repetition plays in enabling public verification of the encoding phenomenon. The simple forms of repetition we review in this study, are in line with one of WRR's central insights of multiple encodings - i.e. many pairings of ELSs for related words tend to be encoded near each other. Typically the simple repetitions we observe occur very compactly, giving low p-values. Since there is no possibilty of wiggle room in cases of exactly repeating ELSs, these cases enable direct public verification of the existence of codes.

Our work highlights two forms of simple repetition:

Type 1 - Intra-table repetition: Figure 3 showcases this simplest and most immediate form of replication/verification. This is example of how codes seem to hide in plain sight, because it can easily be overlooked as random clustering - until we see how rare it is (todo - refer to the calculation of AL and EDY clusters).

Type 2 - Inter-table repetition: Specially connected "paired codes" provide strong self-verification for the codes phenomenon. Figure todo showcases this. One code could be thought of as a "set-up" and the second code as a "confirmation". Specifically, the setup code establishes a strong connection between a set of key words. Because of the strength of that code, this group of key words become noteworthy - they now have *a priori* status - so that when these key words are found in the second, confirmation code, they are even more noteworthy. Not only is it now statistically legitimate to measure them, but public verifiability is established due to the mutually reinforcing strengths of the setup and confirmation codes. Public verifiability had been a major challenge until this discovery. Years of post-WRR research revealed this apparent built-in verification system within the codes themselves. This was an unexpected finding, yet in line with WRR's attention to multiple encodings.

14 Appendix - Extra Care Needed for Randomization Methods

Following are examples of conditions where randomization methods can yield inconsistent results, and how we resolve or avoid them.

14.1 The Area Measure

Our use of area[33] as the metric for compactness can result in over-rewarding or under-rewarding Torah tables compared to monkey tables. Over-rewarding can happen if the Torah table takes advantage of a commonly occurring horizontal word (with skip 1, but also occasionally with skip ¿ 1). Under-rewarding could happen if the Torah table contains hard-to-find vertical ELSs for words that are common horizontally, since most monkey tables in such a case tend to form by using those more common, easier-to-find horizontal occurrences. When we recognize such an imbalance, we have a few options:

1. We can manually examine the outcome and reject the Torah table and/or those monkey tables that have a particular feature that is overly common[34].
2. We can require all hits to have a minimum number of vertically stacked letters.
3. We can set a parameter to prohibit all horizontal ELSs.
4. We can replace a formation of multiple ELSs in the Torah table with a single vertical surrogate axis (see **??**).

14.2 Overlapping letters

Overlapping letters can give biased results under the ELS random placement randomization protocol[35]. Alternative randomization methods, such as permuting words within verses or chapters, can solve this problem, but can also introduce new problems[36]. In the current study, there are only a few cases where this could have been a problem, and we resolve them by using a surrogate axis[37], and/or by using the crystal-growth algorithm[38].

14.3 Horizontal ELSs

Examine competitors - if lopsided (over abundance of horiz or vert - adjust accordingly - not needed in current study other than checking it for the HAL CPN AL EDY case, and that was OK).

15 Appendix - Parameters and Their Constraints

Overriding guideline:

[33] That is, area of the rectangular bounding box that contains all letters of all ELSs in a table.

[34] Such as an ELS with skip 1 for a common horizontal word in the text.

[35] This happens because a randomly shifted axis overwrites whatever letters it is shifted to, and if those letters would have been helpful for non-axis words that cross the axis, in most cases they have been replaced by a different, non-helpful letter.

[36] such as horizontal patterns that span more than one word of the text, being consistently disrupted due to the permutations.

[37] This method can avoid the overlap completely

[38] This method is not subject to an overlap bias because the randomized tables have the same potential for overlap as the original tables.

In our study we are interested in reducing the tree of possible cases to examine, by focusing only on those branches of the tree that are the strongest. We do so by limiting parameters to constrained values.

The 2D method does not consider a 2-letter ELS with common letters to be encoded unless it appears at least 3 times in a table. todo - this rule may have been used only once as part of a sanity check - todo - double check.

Section refparmf specifies the other constaints used in our work.

16 Appendix - Heuristics and Simplifications

This is a list of heuristics and simplifications used to ensure unbiased measurements. It is quite detailed and we include it for anyone who attempts to replicate our results.

16.1 Special Cases

Category 1 - skip versus area for the 2D algorithm:

For some Torah tables, when comparing them to competitors[39], we need a method that balances skip distance against table compactness[40]. WRR addressed this issue through two concepts: "expected number of ELSs" and "domain of minimality". These concepts were very useful when aggregating WRR's many ELSs arising throughout the text for single key word pairs. For our work, for the individual tables that we analyze, we avoid this extra complexity.

We use the following simple heuristic: we allow each competing table to use skips up to (todo 5) times that of the original Torah table. Within that set of competitors, those tables that have smaller areas are considered superior to the Torah table. This heuristic goes beyond a more traditional one which uses a factor of 2 rather than 5[41].

The heuristic is a temporary and simple measure to handle this issue, until we can implement a more sophisticated quantitative method. Note that this heuristic is needed only when we want to evaluate a table's initial framework. As such, it only affects todo (one?) table in our current study (figure todo).

In contrast to the above, we often start with a set of specific, core ELSs that form the framework for a table. If that framewok is known to be the best or near-best in Torah, it is used to anchor the rest of the table. In that case additional words in the surrounding area are evaluated relative to their proximity to the framework (example = figure todo), in which case there is no issue of balancing table compactness and skip distance - both are essentially set by the table's anchoring framework.

Short keywords

[39] whether those are other Torah tables or monkey tables

[40] We need to determine at what point a minimal skip outweighs mediocre compactness? Likewise, at what point does exceptional compactness justify a larger skip?

[41] The thinking for the factor of 2 was that the average skip of all monkey results will be close to that of the Torah table's skip - however that is dictating to the monkey texts based on a known Torah result, so we prefer the more conservative factor of 5.

Unlike WRR, two to four-letter ELSs are valid in the visual tables we deal with. Three and four-letter ELSs are not treated by the 2D program specially. In the case of a two-letter key word with common letters, there are two cases:

(a) If the key word is being sought as a cluster, the crystal-growing algorithm will look for many connected occurrences in small areas.

(b) Otherwise, the normal 2D processing will be used, but in all cases it will require at least 3 ELS occurrences (not necessarily connected) in order to consider the table to contain that key word.

Category 2 - overlapping letters:

Avoiding overlapping letters in monkey texts. Table 103 again is a good example. The original code included YHWH that overlapped with the other words. Overlap can result in bias in monkey text results when ERP is used, so we used a non-overlapping substitute word, HHW, in place of YHWH. The substitute word has more possibilities for placement in the monkey texts than the original word, so it is properly conservative.

Category 3 - surrogate axes:

We often use a surrrogate axis (category 3 below), typically a long ELS which occurs only once (or to be conservative, a few times) in Torah. We choose an ELS that actually exists at the location of the anchored table, close to the center of gravity of the portion of the table that defines the anchor.

Substituting a surrogate axis that is equally or less favorable than the core words of a code. The surrogate must be similarly challenging to find as the original core words were to find. The table with skip 12 is an example (Figure 9). The original core words (AL cluster combined with EDY) served to anchor the nearby horizontal words HAL and CPN. A surrogate axis, in our case NYMLARA (the vertical string next to the EDY column) substitutes for this original anchor. This is helpful because it has the same effect as the original EDY anchor (actually it is intentionally more favorable for the monkey texts because there are two columns that could be considered the center of gravity for the AL cluster with the EDY ELS - and our surrogate uses the column further from the HAL and CPN ELSs, making the area of the original table larger, and easier to compete with). The surrogate axis allows us to combine results of multiple programs; and it also simplifies the Monte Carlo measurment by allowing the most straightforward application of ERP (a single axis with nearby, non-overlapping words that we measure) - avoiding problems documented above in 14. There are two characteristics that we adhere to when choosing the surrogate axis:

1. It must appear in Torah with the same (or less) rarity as the original core word(s) we are substituting for. In this example, the original core, the AL EDY cluster, was the best in Torah. Therefore our axis must also be the best in Torah - for the skip range we choose - i.e. it must appear only once in the skip range we choose. Recall that each axis instance in the original Torah is randomly changed - both its location and skip - to produce a potential competing monkey table. The skip can be up to double the skip range we choose.

2. The skip range must be wide enough so that our monkey table's skip can be widely variable. In that way, we avoid many monkey tables coincidentally appearing in the same areas of Torah (we also outlaw the original skip (or its double) because otherwise the monkey table would sometimes use the same non-axis ELSs as the original table).

Category 4 - random number cycling:

Very occasionally a Monte Carlo run will be caught in a random number cycling problem - this is identified by both a repeating monkey competitor and an even spacing between these monkey competitors - for example after every 52,512 iterations we get the same specific offset and skip for the shifted axis (i.e. the run repeatedly finds the same monkey hit at even intervals). This can be solved by muliple small runs using a different seed for each run.