3. Data Chapter 2: Towards a high-utility general signature for sequence structure

3.1. Introduction

'*K*-mer' is a term typically used to describe the set of fixed length substrings found within a larger string. In recent years *k*-mer based analysis, is used widely to perform QA/QC on NGS data (Andrews 2014), to estimate pre-assembly statistics for genomes (Simpson et al. 2009), to build predictors for sequence associated biological features (Liu et al. 2015), and even to taxonomically classify the content of metagenomes (Ounit et al. 2015). K-mers may also sometimes be referred to as n-grams, or in the case whereby the length may not be fixed: *l*-mers. For the sake of clarity, in this chapter the following terminological code will be followed: *k*-mer will be used to describe the fixed length substrings which constitute the maximum length inputs to a substring-based method, while *l*-mer will be used to describe instances of substring usage over the range of [1, *k*] within those methods.

A primary problematic issue with *k* or *l*-mer based methods for classification is the high dimensionality of longer DNA sequence. For a *k*-mer of *n* length, the number of available sequence types is 4^n, when predictive sequences reach 15-20 base pairs, it typically becomes necessary for the sake of computing power to develop heuristics which limit the sequence space explored by the classifier. Various programs have also been developed to optimise the containment of high dimensionality in working memory for the sake of *k*-mer counting (Marcais & Kingsford 2012) (a routine operation in various other pipelines, such as the Trinity Transcriptome assembler (Brian J Haas et al. 2013)).

Another issue with k or I-mer based approaches to DNA sequence computing, particularly with respect to machine learning, is the fragility of longer *k*-mers. Most modern machine learning methods rely on inputs of fixed dimensionality and size, which results in DNA classifiers using kernel matrices of *l*-mer frequencies derived from a training set of sequences. Although amongst the most highly predictive sequences, longer *k*-mers are also incredibly sparse entries in kernel matrices, which make models derived from them difficult to train. In response to this shortcoming, work has recently been done to attempt to bandage this issue using a gapped *k*-mer approach to kernel matrix construction for support vector machine (SVM) classifiers (Ghandi et al. 2014).

Ghandi *et al's* algorithmic method involves the construction an efficient tree-like data-structure with additional branching between nodes which differ by N bases, this may allow the aggregation of many similar long *k*-mers into a single entry in a kernel matrix, which can produce a more reliable input to an SVM (Ghandi et al. 2014).

The idea of a gapped *k*-mer tree will be central to the foundations of the method described here. However, there are several other categories of biological sequence processing which inform the development of this method. The first, as mentioned above, are the counting and statistics tools used in the data processing pipelines for many NGS experiments. Work, although limited in scope, has been done to apply these tools to derive an informative bird's eye view of an organism's biology. Most straightforwardly, this has been done by calculating whole-genome *k*-mer frequency histograms as a comparison tool between species (Chor et al. 2009). Another way in which these tools may directly inform us biologically include allelic diversity estimation (Simpson 2014), although a *k*-mer based estimation of heterozygosity will lose sensitivity when the density of genomic SNPs is regularly greater than *1/k*, or when the overall rate is exceptionally small. A third way might be for the preliminary detection of intracellular parasites or other sources of non-host DNA present in the sequencing experiment without direct classification (Kumar et al. 2013). While useful, these tools also have a relatively low-dimensional output relative to their inputs, often taking the form of a two or three-dimensional distribution of frequencies. The notion that a large-scale sequence set might be described biologically in a knowledge-free manner seems appealing but achieving much depth to the analysis is challenging.

Research not directly related to the use of *k*-mers in the same manner, which yet still attempts to gain a bird's eye view of the DNA's information content comes often from an 'information entropy' perspective. Information Theory developed by Claude Shannon (Shannon 1948) has been the basis for much entropic theory of information and is referred to as Shannon Entropy (Lin 1991). The methods developed around which are principally concerned with the nature of DNA insofar as it diverges from a random noise comprised of the same alphabet (Mantegna et al. 1994). Attempts have been made to describe an information entropic 'signature' of DNA (Schmitt & Herzel 1997). Others have also found novel approaches to the idea of entropy, such as via 'Chaos Game Representation' (CGR) (Oliver et al. 1993). Purely entropic or signature-based descriptions of DNA do not appear to be in frequent use in the age of NGS. There has been some perennial interest in CGR signatures however, efforts have been made to deploy these for the comparison of genomes between species (Karamichalis et al. 2016). Euclidian distances of CGR matrices have also been proposed as a quantified measure of species-distance (Karamichalis et al. 2015). Although the perspective of defining sequences, and even life, by the scale and shape of their entropic properties might capture the signatures of far deeper complexity, the outputs produced by these methods are difficult to translate into stand-alone biological insight in the same way that a whole-genome *k*-mer analysis might be. The objective of this research effort is to determine

whether it might be possible to achieve the best of both worlds: deeper complexity signatures containing direct biological insights.

3.2. Methodology Development

3.2.1. Rationale

It is hard to spend much time as a bioinformatician in the modern day without being required to 'choose a value of *k*' for a program. Although some assemblers such as MEGAHIT (D. Li et al. 2015) may by default opt to run multiple *k* values in serial, whether error correction, genome assembly, or read library pre-analysis, typically a single value of *k* is required. This highlights the difficulty of integrating *k*-mer based algorithms across multiple *k* values simultaneously. Consider that, in an alphabet of size four, ATGC, there may exist one to eight different 9-mers set for every 8-mer. If a given 8-mer's frequency could be explained by the frequency of a single 9-mer, it would be natural to point to the 9-mer as the sequence of interest if it was constitutive of multiple roughly equally frequent different 10-mers. This is quite a simple way of looking at the set of *k*-mers in an entropic manner: If the frequencies of shorter substrings disperse evenly amongst the longer substrings which contain them, it is probably the shorter substrings which carry the biological interest. If the presence of these shorter substrings at an unusually high frequency rate is explained by equivalently high frequency longer substrings which contain them, then perhaps it is the longer that are the more relevant to whatever biological question is being asked. The next step might then be to consider if, for a general methodology which is inclusive of a *range of k (or l)*, rather than selecting *l*-mers by their interestingness or (in the terminology which will be used from here on) distinctness, all *l*-mers over *[1, k]* might be included in the set, but their merit be subject to a 'distinctness weighting'.

To assemble a large set of substring information in such a manner as would allow us to ask this question of an arbitrary *l*-mer, the most basic computational requirement is access to the frequency-containing variable associated with an *l*-mer, and a set of associations between it and the frequency variables of the length *l+1* substrings which may contain it. Fortunately, this condition is satisfied by the widely used efficient *k*-mer tree structure. This is essentially a search-tree with *n* possible children per node, where *n* is the size of the alphabet (In this case, four). From now on the rationale will assume the employment of a *k*-mer tree as its primary data structure. Technically speaking the *k*-mer tree would be defined as a 'trie', rather than a tree, as the actual sequence content of the *k*-mer is not stored in any variable and instead may be inferred from the tree position of a given node. Despite this, since the tree, or trie, is not actually being used for search operations, we will continue to refer to it simply as a '*k*-mer tree'.

3.2.2. Initial Formalisation

Given the parent/child relationship between characters within a set of *k*-mers, and the usage of frequency dispersion to measure distinctness, we can begin to define the formulae employed. Given that child node frequencies are contained by an ascending-value-ordered n-tuple **F =** $(f_1, f_2, \ldots f_n)$.

Formula 1: $$d_{min} = f_p$$

Formula 2: $$d_{max} = \left\lfloor \frac{d_{min}}{n} \right\rfloor n^2 + (d_{min} \bmod n)^2$$

Formula 3: $$d_{child} = \sum_{i=1}^{n-1} (\boldsymbol{F}_i - \boldsymbol{F}_{i+1}) i^2 + \boldsymbol{F}_n{}^2$$

Formula 4: $$D = \frac{d_{child} - d_{min}}{d_{max} - d_{min}}$$

Giving:

Formula 5: $$D \in \mathbb{R}\ (0 \leq D \leq 1)$$

Formulae 1-4 describes the *l*-mer distinctness found for the node described as the parent in this context. Here f_p and f_c refer to the parent and child node frequencies respectively, *n* describes the length of the alphabet, and *d* the various frequency distribution scores. The vector of child node frequencies is also pre-sorted from low to high. The distinctness D thus measures in a linear fashion the distance in frequency distributions between the least distributed state (one child node equals parent node frequency), and the maximally distributed case (child nodes divide the parent node frequency in the most even manner possible given the potential remainder). This linear measurement of distribution equality within a set of values functions as a type of Gini coefficient (Gini 1912), for indivisible integers.

We must first note however an important aspect to the 'distinctness' weight calculation here when using trees over a contiguous range of *l*. Distinct *l*-mers will have evenly distributed child-node frequencies, yet so will even totally indistinct *l*-mers in a tree at a depth shallow enough to be saturated by the input set. This is to say that a null case random 'DNA-noise' input would cause this method to identify many distinct *l*-mers in the tree where $l < log_4(F_r)$, with F_r being the root node frequency (the number of input *k*-mers). To remedy this, we might return to the entropic way of thinking. Simply put, it is not just that structure breaks down at a certain point below an *l*-mer branch, but that it also did *not* do so beforehand. Phrased differently, we could say that a given high frequency branch of the tree ought to have shown some resistance to the expected noise-case dispersion above the depth being considered if its own dispersion of frequency is to be indicative of actual structure. In fact, if the same formulae were applied to both cases, a solution could be to

multiply the distinctness of a node at *l* by the inverse distinctness of its immediate parent at *l-1.* To avoid confusion, we might separate distinctness into D_b: 'base' and D_a 'actual'. Such that:

Formula 6: $$D_a = D_b(1 - D_{b_{parent}})$$

To find a sum of all *l*-mers which escape the entropy of noise, it ought to be enough to perform the above on every node over the range *[1, k-1]*. We might also optionally multiply *Da* by the length of the *l*-mer, *l*, to scale the measurement by the sizes of the retained structures, and/or we might multiply by the node frequency. A combination of these terms from now will be referred to as a resistant structure score. Generalising slightly from the range of *all* nodes, we can observe that is would be possible to find the resistant structure score, *S,* of any sub-tree recursively with respect to its root *r,* using a depth-first-search (DFS). See Formula 6. In the case of finding a singular quantification of the scale of the entropic-resistance in the genome, the root *r* would be the actual head-node in the tree.

Formula 7:

$$S_r = \sum_{v \in Ch(r)} Da_v * f_v * l_v$$

Here *v* represents a node (or vertex), and *Ch* the recursive application of the summation function to its children.

One aspect of the dispersive tree measurements process which has not yet been addressed is the directionality of the tree. As discussed in the Rationale set down in Section 2.1, there are eight, not four, DNA (*l+1)*-mers which may contain any given *l*-mer, however the tree structure accounts for the terminal extension bases. Since the tree expands by powers of four (in the case of DNA), the depth at which the tree becomes less saturated, and more informative will only be increased by an average of 0.25 by doubling the frequency. This means there is perhaps enough wiggle room to merely read all inputs twice: once forwards, once backwards.

However, this issue also intersects with the strandedness of DNA, which contains one forward and one reverse complementry sequence. A simple solution could be to capture the other four base extensions in the form of reverse complemented *k*-mer inputs, this would also have the effect of unifying motifs that have been sequenced on multiple occasions from different strands, thus separating their frequencies, despite their biological identity. For protein sequences however, a simple reversal would suffice.

There is another slightly counter intuitive aspect to this calculation which also requires attention. The statistical means taken of any categorical vector of structure scores will always resolve at their current depth, with respect to the non-dispersed structures of higher values of *l*. This is to say that a high frequency 20-mer which shares a constitutive 8-mer with another low frequency 20-mer will cause a relatively low distinctness score for the 8-mer at the point of separation. This effect lowers the mean for the scores at the 8-mer depth. When the whole tree is summarised however, so long as it is deeper than 20 in this case, the higher distinctness of 20-mers and it's the multiplication by *l*, will yield an overall higher structure score for the entire tree. If, however the tree does not extend to that depth, the unregistered frequencies that have 'escaped' will have the effect of incorrectly lowering the structure-score.

This is a boundary problem – the tree cannot be infinitely deep, in fact computational constraints limit its size quite significantly, and all frequencies cannot be guaranteed to disperse within it. As a result, spectra which cannot be captured by the tree ought, in the case of aggregation methods *within k*, be negated. This involves simply reducing all leaf node branch frequencies to 1 and propagating those subtractions recursively up the tree to maintain equivalent sum frequencies per depth. If the tree were to be used for non-aggregative methods (i.e. motif discovery) this would not be required, it is also not necessarily the case that this correction be required in the case of signature generation. The boundary frequency correction will thus be applied only to the more compact aggregate matrices.

Next, whilst the above may suffice to inform us of a certain property of the strings in the input set, we also must return to the biological manifestation of the *k*-mer, principally, to return to the classification issue: the biological fragility of long substrings. Not all bases in a string may be constitutive of the active motif. There may also have been duplications of motifs which then experienced mutations, none of these aspects of genomic structure would be detectable by a simple *k*-mer tree as we have described so far. For example, an 8-mer might smoothly distribute its frequencies amongst 9-mers, yet all subsequent substrings up to length 20 may continue identically, yet they will do so in four separate branches of the tree. In this case, the 20-mer with a single flexible base will not be discovered at its true frequency. However, if one were to introduce an extra character 'N' as a child node through which all input strings reaching its parent node are additionally to be passed, the subtree originating from the 'N' child node would describe accurately the full frequency 20-mer sequence. Figure 29 shows a basic example of how the merging of subtrees occurs to create an effective 'N-mask' in a *k*=4 binary tree.

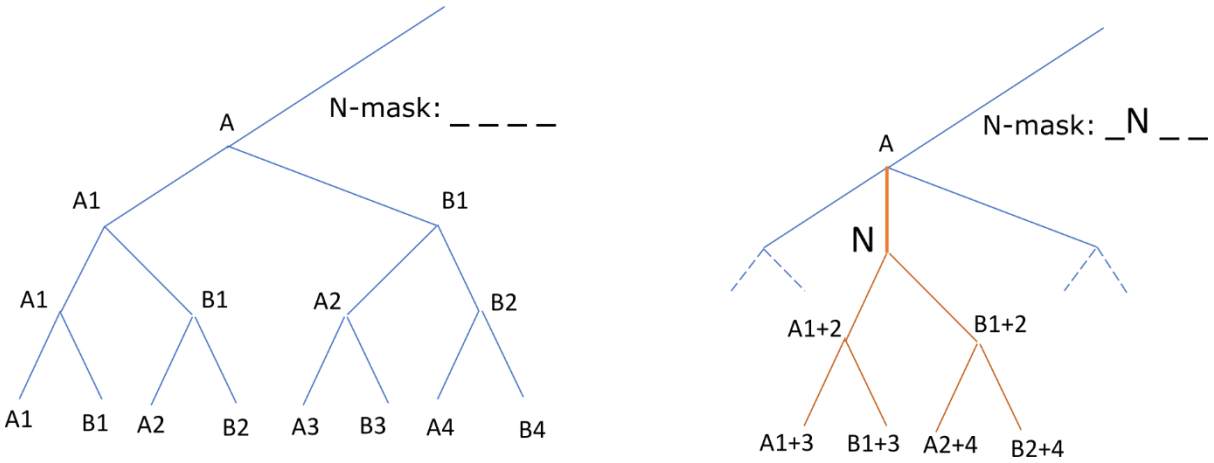

Figure 29. Example of binary tree aggregated for a certain N-mask.

Generalising from this example, we can see than for a single flexible base, an 'N' subtree would then have to be generated for every node in the tree with more than one non-zero frequency child. In the case of multiple 'N'-containing motifs, an 'N' subtree would also have to generated for nodes in the initial 'N'-child subtrees, and so forth. This does however have advantages. Firstly, since each 'N' subtree is independent from the rest of the tree above its parent, the memory usage can be contained by only generating (and deleting) subtrees as they need to be measured in a single 'depth-first search' (DFS). Secondly, the expansion of computing power and memory usage with additional Ns remains constant when the number of children per node is increased (as we are not at this point investigating transitions vs transversions, or other partially selective evolutionary conditions). As a result, its polynomial efficiency might yet be a suitable trade-off, particularly in the case of larger alphabets, such as with peptide sequences.

3.2.3. The Aggregation Methods for 'N-masked' *l*-mers

Aggregation methods in this case refers to a structured and systematic way that variables can be aggregated from a complex source. The aggregation methods can also be thought of as independent of the variable types gathered. Given that we assess the tree on a per-node basis, an aggregation method could be applied to gather various measurements in the same manner, although at first, we will explore them from the perspective of the development of signatures derived from structure scores.

Let us return temporarily to re-examine what is meant by a 'signature'. It could be said that the signature of an aggregated set of scores is created as much in the process of selective aggregation as it is in the data's original complexity. As in the case of imaging sequence Shannon Entropy (Tenreiro MacHado 2012), or CGR images (Oliver et al. 1993), we can see that the signature is typical displayed as a 2 or 3-dimensional array of points. The case of CGR images used for distance metrics also

highlights the importance of comparability between signatures (Karamichalis et al. 2016). This is to say that, a signature ought to retain the same dimensions and size regardless of the input data. When aggregating scores from the tree therefore, we ought to construct the dimensions of the output matrix from sources which can be measured regardless of the sparsity of the tree.

The first dimension seems most suitably to be *l,* over the range of *[1, l-1].* The terminal value of *l* cannot have scores data extracted as the calculation involves the node in question to have children with populated frequencies (*i.e.* it cannot be leaf node). All *k*-mers read into the tree are of length *k* and therefore all depths of the unmasked tree will share an equal sum of frequencies. This means that each category of *l* will always reliably contain measurable structure scores. The most basic output summary of the flat tree will thus be a single vector of structure scores of length *l – 1.*

Formula 8: $$Sig1D = \begin{bmatrix} S_1 \\ \vdots \\ S_{l-1} \end{bmatrix}$$

When choosing the second dimension, we begin to consider the structure of the N-masked tree also. In this case there are multiple options, and there might also be multiple correct answers. For example, it would be of biological interest perhaps to aggregate all scores which originate from N-masks with equivalent numbers of Ns. This could give us an estimate of the interaction between *k*-mer replication and divergence. This is also quite a straight forward output matrix. It is also worth noting that the first output is a subset of the second: the vector of N=0 scores comprises the first column.

Formula 9: $$Sig2D = \begin{bmatrix} S_{1,0} & \cdots & S_{1,N} \\ \vdots & \ddots & \vdots \\ S_{l,0} & \cdots & S_{l,N} \end{bmatrix}$$

Whilst the above output matrix is suitably interesting for an expanded *k*-mer spectral summary, and worth including as an informative set of datapoints, it also fails to include much of the inner complexity of the space of N-masked frequencies. One issue with categorising N-masks however is that their categorical dimensionality for deeper tree is very high (at 2^k), and with the sparsity of a DNA tree at *k*=31, the expected sparsity of the individual N-mask categories would disqualify them from direct usage as a means of aggregation for the creation of a signature. Therefore, we might try to find a middle road for the creation of second and third output matrix dimensions. The first pair of dimensional measurements to be investigated here will be the left and right ‘seed length’. Seed length refers to the size of the either side of the N-mask (beginning with either the root or leaves of the tree) which contains no Ns. In other words, the length of the fixed seeds pre- or post the variable region of the *l*-mer. Let these index terms be *s,* and *d* (sinistral and dextral). Since not all values of *s* will be valid

for all values of *d*, the output matrix will instead be a 3D wedge-shape. This indexing system is further explained by Figure 30.

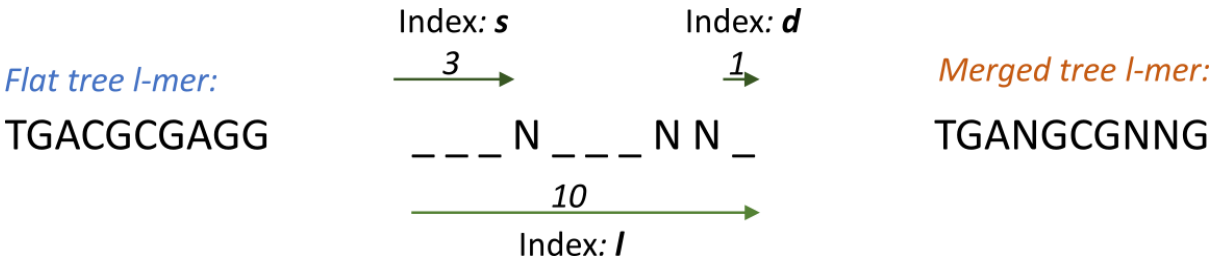

Figure 30. 3-Dimensional Indexing system for N-masks (DNA).

The matrix indices are defined as:

Formula 10:

$$\{ l \in \mathbb{Z} \mid 0 < l < k \}$$

$$\{ s \in \mathbb{Z} \mid 0 \leq s \leq l - d \}$$

$$\{ d \in \mathbb{Z} \mid 0 \leq d < l - s \}$$

Such that:

Formula 11:

$$Sig3D = \begin{bmatrix} \begin{bmatrix} S_{1,0,0} & 0 & 0 \\ \vdots & \ddots & 0 \\ S_{1,s,0} & \cdots & S_{1,s,d} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} S_{l,0,0} & 0 & 0 \\ \vdots & \ddots & 0 \\ S_{l,s,0} & \cdots & S_{l,s,d} \end{bmatrix} \end{bmatrix}$$

This matrix has the property of finding some of the inner complexity in motif flexibility shapes. However, one of its flaws is that the information space from which S is sampled is variable. For example, the lower values of both *s* and *d* present a much larger computational space of sequence flexibility when N is high, than the higher values of *s* and *d*. To create better consistency in the scaling of aggregation categories. We could also re-introduce the number of Ns in the mask as fourth dimension:

Formula 12:

$$Sig4D = \begin{bmatrix} \begin{bmatrix} S_{1,0,0,0} & 0 & 0 \\ \vdots & \ddots & 0 \\ S_{1,s,0,0} & \cdots & S_{1,s,d,0} \end{bmatrix} & \cdots & \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \\ \vdots & \ddots & \vdots \\ \begin{bmatrix} S_{l,0,0} & 0 & 0 \\ \vdots & \ddots & 0 \\ S_{l,s,0,0} & \cdots & S_{l,s,d,0} \end{bmatrix} & \cdots & \begin{bmatrix} S_{l,0,0,n} & 0 & 0 \\ \vdots & \ddots & 0 \\ S_{l,s,0,n} & \cdots & S_{l,s,d,n} \end{bmatrix} \end{bmatrix}$$

However, this version of the aggregation method may not be of much advantage versus the compactness of the 3D version, particularly when the analysis is limited to lower values of N (1-3). For very large sequence input sets (as in the original intended purpose), it may be computationally difficult to increase N substantially, as such the 3D signature aggregation matrix may suffice, however the 4D version perhaps ought to be applied should a version be developed with either smaller input sets, or substantial efficiency improvements in achieving summaries of higher dimensional N masks.

3.2.4. Cases for Aggregation Modes

Although not the principal objective of this research, summarising the total contained structure in the tree is something which might also be useful for large scale projects comparing hundreds or thousands of input sets in an external informative context (i.e. phylogenetics, lifestyle, environmental variables). It might also be useful in the case of segregations made within individual genomes, making experiments between annotation types possible. For example, testing regional information structure between intra-and inter-genic DNA, or between repeat types, or along physical chromosome maps etc. For this reason, the Structure score summaries will still be included as outputs using the simpler 2D signature matrix (Formula 9).

Formula 7 describes a quantification of *l*-mer structures found in the genome. However, it does not provide us with a metric that is easily comparable between genomes, principally because of the confounding factors of size and ploidy. The simple solution would be to always divide the figure by the head-node frequency (post boundary correction). However, this solution only normalises the un-masked tree, simply because the creation of merged subtrees duplicates and re-measures the same frequencies in a different way. Additionally, in many cases subtrees will not be generated where they are not needed. The solution would be to sum all frequency duplications and add them to the head-node frequency, such that all recorded structure is normalised to the summed scale of the frequencies used in the entire data structure.

Formula 13:

$$S_r = \frac{\left(\sum_{v \in Ch(r)} Da_v * f_v * l_v\right) + \left(\sum_{v \in Ch(r)} \sum_{x \in Merge(Ch(v))} (Da_x * f_x * l_x)\right)}{f_r + \sum_{v \in Ch(r)} f_v}$$

Formula 13 shows the summation of genome-size normalised structure for N=1.

Formula 14:

$$S_r = \frac{\left(\sum_{v \in Ch(r)} Da_v * f_v * l_v\right) + \left(\sum_{v \in Ch(r)} \sum_{x1 \in Merge(Ch(v))} \left(Da_{x1} * f_{x1} * l_{x1} + \sum_{x2 \in Merge(Ch(x1))} Da_{x2} * f_{x2} * l_{x2}\right)\right)}{f_r + \sum_{v \in Ch(r)} \left(f_v + \sum_{x1 \in Merge(Ch(v))} f_{x1}\right)}$$

Formula 14 thus shows the summation function for N=2.

Formula 15:

$$S_r = \frac{\left(\sum_{v\in Ch(r)} Da_v * f_v * l_v\right)+\left(\sum_{v\in Ch(r)} \sum_{x_1\in Merge(Ch(v))}\left(Da_{x_1} * f_{x_1} * l_{x_1} + \ldots + \sum_{x_n\in Merge(Ch(x_{n-1}))} Da_{x_n} * f_{x_n} * l_{x_n}\right)\right)}{f_r + \sum_{v\in Ch(r)}\left(f_v + \ldots + \sum_{x_{n-1}\in Merge(Ch(v))} f_{x_{n-1}}\right)}$$

Formula 15 shows the generalised extension of the formula for N=*n.* This will be how we assign structure scores to the sequences in the input data.

Although Formula 15 shows the more complete summary of a structure score for a whole tree, its components being frequency, distinctness and size, there are cases in which these components might be more usefully extracted as separate measurements. Indeed, since the signature indexing system does not recursively allocate to the same variable either, a slightly different definition is required.

It is here that we draw a distinction between signatures and summaries. The inter-comparable utility of signatures is maximized not just by equivalent dimensions, but by comparable term regularisation. For example, the mathematics required to compare two sets of variables over *[0, 1]* will inevitably be simpler than in the case of natural range of structure scores, which are essentially unlimited in scale. The variance of the sets of structure scores will also vary wildly with the size of the input sets. Given the range of eukaryotic genome sizes (The 12 MB of *Saccharomyces cerevisiae* to the 149 GB of *Paris japonica*) a signature ought to at least attempt to constrain the distributions of its values to normalised range, even if variance differences will still be inevitable to some degree. For this reason, for the purposes of complex signatures, the aggregation modes described above ought to be applied to gather the *weighted arithmetic mean* of the distinctness of each category. For example, in the case of the 3D signature (Formula 11) matrix:

Formula 16: $DF_{l,s,n} \ni \sum D_a * f$ and $F_{l,s,n} \ni \sum f$, then:

Formula 17: $\overline{D}_{l,s,n} = DF_{l,s,n} \div F_{l,s,n}$

3.2.5. Derived Measurement Types

When considering additional descriptors of the aggregate categories in the signature matrix, it is worth observing that each categories could also be thought of as its own vector of values with its own distribution. Here we propose two additional possible distribution qualities to be measured, formatted as concurrent signature matrices, and the rationale behind them.

The way in which the distribution is qualified will depend on the expected size of the vectors. There are two perspectives considered here. The first is the ‘small vector’ distribution. This is the case

where the signature's input set might be small, for example, a single gene-family, repeat type, or a set of differentially expressed transcripts. Here we might be more concerned about the variance in the distribution, as a single reading may have captured a specific few biologically relevant active motifs. The second perspective is the 'large vector' distribution/large inputs (-omic scale data types). Here we can begin to make safer assumptions about the shapes of the distributions encountered and measure them differently.

Regarding the 'small vector' distributions, as each N-mask category has a given weighted mean *l*-mer of distinctness, this does not tell us the anything about the distribution of that property. Biologically it might be informative to know whether a given N-mask category reliably produces low or high distinctness, or whether its mean is the result of a broad range of inconsistent measurements. For this purpose, we could simply employ a weighted standard deviation (WSD). Like the weighted mean, the frequencies would be used as weights. This would allow us to produce a parallel signature matrix of deviations.

Formula 18:

$$\sigma_{l,s,n} \ni \sqrt{\frac{\sum_{i=1}^{n} \boldsymbol{F}_i(\boldsymbol{D}_i - \overline{D}^*)^2}{\frac{n-1}{n}\sum_{i=1}^{n} \boldsymbol{F}_i}}$$

The second distribution of interest, regarding the 'large vector' case, relates to the power law. The power law has been observed to be broadly acting property of many natural systems (Newman 2005). Pareto-like distributions of properties have in fact been observed as consistent features of life systems at many scales (West et al. 1999). For example, it has been demonstrated to be a consistently emergent feature of metabolic networks that they be scale-free (Jeong et al. 2000). Additionally long right hand tails on most observed *k*-mer frequency graphs produces of biological sequence also show the Pareto-like distribution of frequency amongst substrings (Chor et al. 2009).

Although it cannot be guaranteed of any given input set that the F * D scores of *l*-mers will follow a pareto distribution, in the case of the largest scale biological data it is an assumption which allows for a more sophisticated measurement. The Pareto distribution formula in its original form is parameterised by two variables, *a* and *m*. The 'shape' parameter, *a,* acts as the variable which may be used to fit the distribution in a real data set, *m* (or minimum) is a simply a translating parameter defined as the minimum value in the data set. We would therefore choose the shape parameter as the most informative component of the distribution to estimate. A maximum likelihood estimated of *a* is quite straightforward (de Zea Bermudez & Kotz 2010):

Formula 19:

$$\hat{a}_{l,s,d} \ni \frac{n}{\sum_{i=1}^{n} \log\left(\frac{\boldsymbol{S}_i}{\widehat{m}}\right)}$$

Where $\boldsymbol{S}$ is given to be a vector of structure scores, and m is the minimum value in that vector. And to avoid confusion, n in this case refers to the size of the vector of values. This gives us another parallel signature matrix. This calculation could similarly be applied to any of the defined output matrix types (Formulae 8-12). Given that structure scores below 1 are possible, it could also be a good idea to set a lower bound to the structures included to the calculation (at least > 1).

3.2.6. Null Trees: Local and Absolute

Here we tackle the issues of single base/peptide frequencies, and the limitations imposed by saturation of the data structure.

Saturation in this context refers to the extent to which a random set of strings, present at a high enough frequency, will populate fully the *k*-mer tree data structure up to a certain depth. The relationship between the frequency of the head node, and the absolute null expected saturation is simply $log_n(f_r)$, where n is the size of the alphabet, and f_r is the root node frequency. This is the case which assumes all character frequencies are evenly distributed, as are all multi-character combinations. This impacts our measurement of frequency, which is an essential component of most of the measurements used. There are two polarities we must contend with whilst we are measuring frequency: Situations where the depth of the tree is such that the null expectation of *any* given node having a frequency of one or greater is vanishingly small, and situations where the null expectation of frequency may by in the hundreds of thousands. It would be erroneous to attribute low-*l* high frequency nodes the property of possessing an indicator of biological structure particularly when their frequency is comparable to one that might be found in the absolute null tree. Similarly, it would run afoul of multiple-testing error to weigh deep high frequency nodes by their individual improbability. We can also note that frequencies are used in two cases, as in Formulae 1-3 to discover D_b, and as in Formula 16-17, to weight the contribution of D_a to the mean of the given category. The proposed correction to f only applies to the weight, rather that the calculation of D_b, as this is not susceptible to the same scaling issues.

Formula 20:

$$f_c = \frac{f_v}{\boldsymbol{max}\left(\frac{f_r}{n^l}, 1\right)}$$

Formula 20 shows the correction of f_v (per vertex/node), by finding the null expectation of frequency saturation at the current node by dividing the root frequency (f_r) by the size of the sequence space of the tree at the current depth (n^l). By providing the lower bound of 1 to the denominator, the effect of the function will only apply at the 'null saturated' depths of the tree. This correction will thus be applied to all cases where *f* is used as a weight.

Since we are expecting some degree of saturation, one complaint we could find against the application of formulae 1-3, is that they range between total conservation and the maximum possible dispersal. Given that most organisms tend to have some bias in their genomic base frequencies, the actual null (i.e. random) dispersal for most of high frequency *l*-mers will rarely reach the maximum possible. In fact, a 40% GC ratio (as in the human genome) would see many higher structure scores measured in cases where it is absent merely due to the base composition of the input. One seemingly intuitive way this problem could be addressed is by weighting the frequencies of the child nodes based on the input character ratios. However, this creates other unwanted sources of bias due to another aspect of DNA base ratios: they are not evenly distributed. The phenomenon of GC and CpG Islands is quite well established (Aïssani & Bernardi 1991). It refers to regions of the genome which are usually dense in protein coding genes. If a specific mean base frequency were used, it could lead to regions of 50:50 CG:AT having their structure scores weighted higher than they should, and the vice versa for other GC depleted regions.

The solution proposed to 'correct' for the base frequency artefacts is to generate a 'local null' tree prior to the generation of the main tree. The local null is a model of the null distribution of *l*-mers given only the actual uneven distribution of base frequencies as it occurs in the input set. This is created simply by building the main tree in all respects identically, except for a random shuffle performed on all input substrings. This preserves all base frequencies but eliminates their structures. In the case where reverse complements are also input, the random shuffle with occur first. The signature matrices (of *D * f*) generated by the local null tree might then be simply subtracted from the output. Integrating this with the weighted mean calculation would give us Formula 21.

Formula 21: $$\overline{D}_{l,s,d} = \boldsymbol{max}(DF_{l,s,d} - Null_{l,s,d}, 0) \div F_{l,s,d}$$

The subtraction of the local null might also be factored into the calculations of the other derived measurements. We will explore its applications next.

To correct the estimation of a Pareto shape parameter, we could, as mentioned in 2.5. increase the lower bound of the scores processed to the local null mean, however since we know that the local-null effect will apply to all structures, it would only serve to falsely alter the distribution. For the

purposes of the single shape parameter which describes in total signature, i.e. as an adjunct to Formula 15, the solution we propose here is to aggregate the total shape parameter in stages, and to weight the contribution of categories based on their null-to-actual structure ratio. To do this, we aggregate the components of the shape MLE separately, *n*, and *log(S/m),* via the 2D aggregation matrix. The categorical actual-to-null ratios then scale each contribution – such that the final shape parameter is largely comprised of the contributions from the tree unaffected by the local null.

Formula 22:

Where: $x = \sum_{i=1}^{n} \log\left(\frac{\boldsymbol{S_i}}{\hat{m}}\right)$, per aggregation category,

$$\hat{a}_r = \sum_{i=1}^{l}\sum_{j=0}^{n} n_{i,j}\left(\frac{\boldsymbol{max}\left(S_{i,j} - Null_{i,j}, 0\right)}{S_{i.j}}\right) \div \sum_{i=1}^{l}\sum_{j=0}^{n} x_{i,j}\left(\frac{\boldsymbol{max}\left(S_{i,j} - Null_{i,j}, 0\right)}{S_{i.j}}\right)$$

The individual category correction for shape parameters is more challenging, as the set of structures generated by both trees will be heterogenous and indirectly comparable in the same way as the output matrix. Currently a correction for single categories will not be deployed, particularly as the shape parameter becomes less stable/informative in lower values of *l* where the saturation is most likely to occur.

In the case of small input sets, we will argue that they ought not be 'local-null' corrected. In larger sets containing multimodal base ratio distributions, and a deeper and uneven saturation, the local-null can mitigate confounding effects to allow the structural content to be inter-comparable despite these factors. However, with small input trees saturation will be minimal and base ratios more typical and descriptive of the specific focus source of sequence, these things could be considered characteristics of the set rather than factors to mitigate. For this reason, the proposed usage of the pair of weighted structure score means and weighted SD for small sets will not be subject to null correction unless the results should provide a compelling reason to do so.

3.3. Implementation

The program was written in C++11 and is only compatible with UNIX-based systems. The program supports multi-threading, although at some memory cost, and at a non-linear performance benefit. The only external library linked is 'pthread'. The maximum depth of the tree in the implementation is currently 32.

Source code is available in Appendix 2.1 'Source code'. There are two slightly different versions of the program. 'UGPep' has a slightly altered indexing system optimised for peptide sequence. 'UGLearner' is the original program which works with both DNA and peptide sequences.

3.3.1. Procedure parameterisation

Input Data – a set of 'fasta' formatted strings of alphanumeric characters

K – The depth of the tree

N – The maximum number of 'Ns' to consider in a single 'N-mask'

3.3.2. Core Data Structures

The primary data structure of the *k*-mer tree is simply a search tree derived from a fixed character set. Each node in the tree is of a type representing a single alphanumeric character and contains available memory references to as many potential child node types as the character set defines. Each node also contains one unsigned integer, which describes the frequency with which it has been traversed during the loading phase.

3.3.3. Data Input

This phase of the algorithm comprises the construction of the tree from a set of strings – likely DNA or proteins. *k*-mers of length *D* will be read sequentially from each string in the input set. The *k*-mer substrings define, by their characters, a traversal of the tree. The head node passes the input string to a child node which matches the leading character in the string. The frequency integer within the child node is incremented by one. If no child node has been instantiated yet, then instantiation will occur. Only child nodes which have been traversed will be instantiated in this manner. Following submission of the *k*-mer string to the child node, the leading character is trimmed, and the function is repeated until a tree depth of *D* is reached, and the input string has been depleted of characters.

Once the first *D* characters of the first string in the input set has been read by the tree, the starting *k*-mer index is incremented by one, and in this manner the following *k*-mer is read.

Over the range of *[0, n]*, where *n* is the input string length, indices for *k*-mer substrings are found in the input string: *[(0, k), (n, n-k)].* This process is repeated for every string in the input set. Every *k*-mer read into the tree in this manner will also have its reverse, or in the case of DNA reverse complement, generated, which will be read into the tree in the same manner.

3.3.4. Sub-tree Merge

The implementation of subtree merging, particularly with respect to memory usage, will be covered before the larger DFS algorithm which calls it. When merging sub-trees with respect to a single node, we are theoretically creating an additional tree structure to hold the merged data. However, in many cases the memory allocated to the pre-existing tree structures can be taken advantage of. For this reason, all merged subtrees with respect to a single node store their variables in left-most child's

subtree. This is to say that the Node class also implements a ‘map’ type, which allows it to store additional unsigned integers, paired with an ID which identifies its N-mask ownership.

Mapped IDs are themselves *l*-length binary tree navigation pathways. The formula for navigating the merged sub-tree variable space is as follows.

Given parent ID:

To access leftmost Child’s frequency in the same subtree: ChildID = ID * 2

To access other children’s frequencies in same subtree: ChildID = ID

To access head-node of new merged sub-tree: ChildID[1] = (ID * 2) + 1

Just so long as all functions follow the above ID manipulation rules with respect to any given node, whereby the initial IDs of all scores in the unmerged tree are zero, the N-mask will always be derivable from the pattern of bits in the integer variable used to identify the score.

This indexing system allows the algorithm to virtualise the retention of merged tree scores within the current tree, without the need to create new Nodes, and the memory overhead that involves.

The tree merging algorithm is a DFS with paired navigation. This is to say that the virtual subtree stored in the leftmost branch of the node of origin is simultaneously traversed alongside an unmerged branch, summing their frequencies into the virtual subtree. This occurs *n* times, once for each of the child-subtrees connected to the node of origin.

3.3.5. Depth-First Search

The aggregation of data within the *k*-mer tree is organised around a recursive DFS function. It is described by Figure 31. The initial values for the parent distinction, depth, and ID arguments are all zero.

```
Function: Search (parentDistinction, Node, ID, depth):
      if(depth < K)
            if (IsMergeable(this))
                  SubtreeMerge(this)

            D_b = FindDistinctness()
            D_a = D_b * (1 – parentDistinction)

            Aggregator.Report(D_a, Frequency, depth, ID)

            if (HasMergedSubtree())
                  Search(D_b, Children[1], (ID * 2) +1, ++depth)

            for i in 1:n
                  if i == 1
                        Search(D_b, Children[i], ID*2, ++depth)
                  else
                        Search(D_b, Children[i], ID, ++depth)
```

Figure 31. DFS tree navigation pseudocode.

3.3.6. Multi-threading

Figure 31 shows that the memory used in the merged trees is created as it is needed. Although not described in the pseudocode, this memory allocation is also deallocated as soon as it is measured and no-longer required for any deeper merges. However, this means that each thread exploring the tree via DFS will have its own substantial memory overhead.

The positive aspect of multi-threading a tree structure is that every child node can point to an independent region of memory. From the root node, *n* threads can be created (one per child), and each thread will not have any memory access conflicts when reading the tree. The aggregator class also implements separate memory buffers per thread, which are periodically read into the output matrices. This mitigates any bottlenecking at that point. The thread allocator can expand tree access for all available threads in this manner, continuing to guarantee independence of memory. The caveat to this approach to threading is that the calling thread also continues to work on one branch of the tree whilst others work on others. Only once all work generated by a node has been completed by the worker threads can the collection of workers be freed up to be reallocated. This has the effect of limiting thread efficiency per spawning node to the performance of the most expensive subtree. Given that character frequencies in biological sequences are rarely equal, this can equate to substantial loss of overall threading efficiency.

The inefficiencies of this threading system are unlikely to be unmitigable. Further performance gains may almost certainly be achieved by optimising the thread allocator. This has not been undertaken yet due to time constraints.

3.3.7. Performance testing

The performance tests were running on a Linux desktop computer with 32Gb of RAM, and a 12-core Intel CPU. Due to the cores available, some of the tests using more than 12 threads may not be indicative of the true efficiency at this scale. However, given the thread availability issue, it can also be beneficial to add more threads to the allocator than can be simultaneously assigned to separate CPU cores.

The following tests are run using subsets of DNA from the NCBI *Escherichia coli* reference genome (Blattner 1997), and subsets of protein sequence from the *Apis mellifera* proteome (Consortium 2006). The depth of all trees, as in the value of *k*, was 30 for all tests.

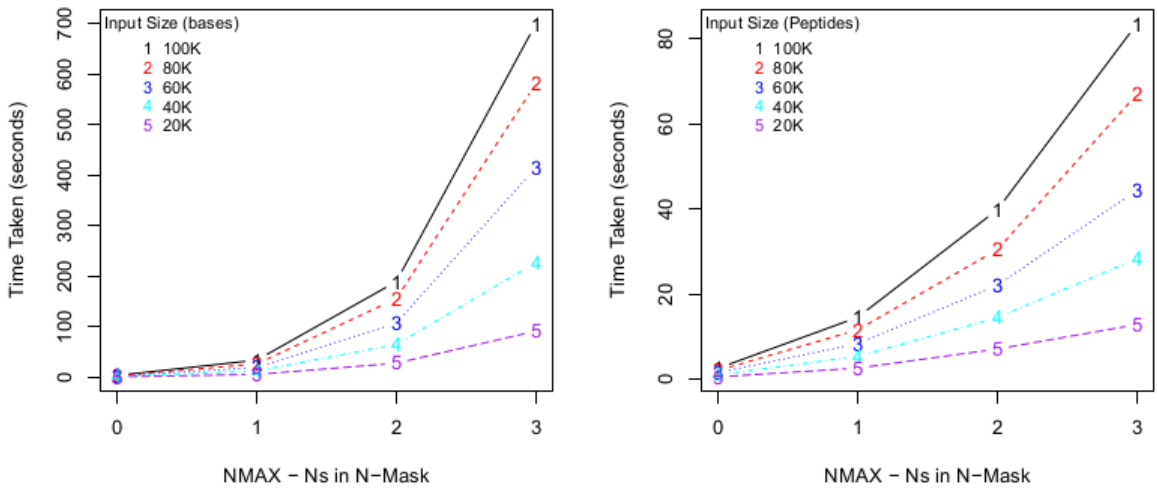

Figure 32. Performance tests for N values over [0,3], one thread. Left: DNA input sets, ranging from 20-100K bases sampled from the E. coli *reference genome. Right: 20-100K peptides sampled from the* Apis mellifera *proteome.*

Insofar as the complexity of the N-mask increases by powers of 2 with every additional N (2^n), the performance of the program reflects this with exponential computation time increments. Interestingly the difference in performance between DNA and peptide input sets are almost a factor of 10. This is likely due to the extreme sparsity of the peptide tree (the space expanding to 20^{30} at the end), resulting in far fewer nodes are meeting the qualifying conditions for the generation of a merged subtree. Additionally, owing to the higher alphabet, the average saturation depth in the peptide tree will also be much shallower (3.94 with reversals in the peptide 100k test set, versus 8.8

in the DNA test).

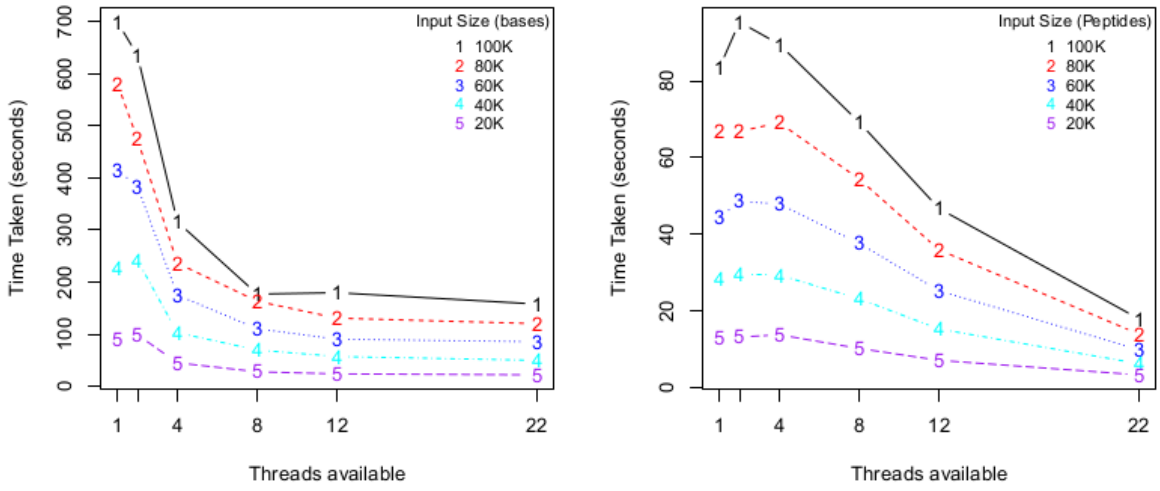

Figure 33. Performance test, multithreaded (N=3), for 1-22 threads executing on a single machine. Left: DNA input sets, ranging from 20-100K bases sampled from the E. coli reference genome. Right: 20-100K peptides sampled from the Apis mellifera proteome.

The results shown in Figure 33 show that the DNA search tree fails to make performance gains above 8 threads. The difference in performance between 2 and 4 threads also suggests that the equal distribution of work between threads from a single originating node of the tree (as discussed in 2.3.6) plays the most significant role in thread efficiency. Figure 34 shows that in both cases, the only time the per-thread efficiency increases following incremental increases from single threaded performance is when the thread number becomes equal to the alphabet size.

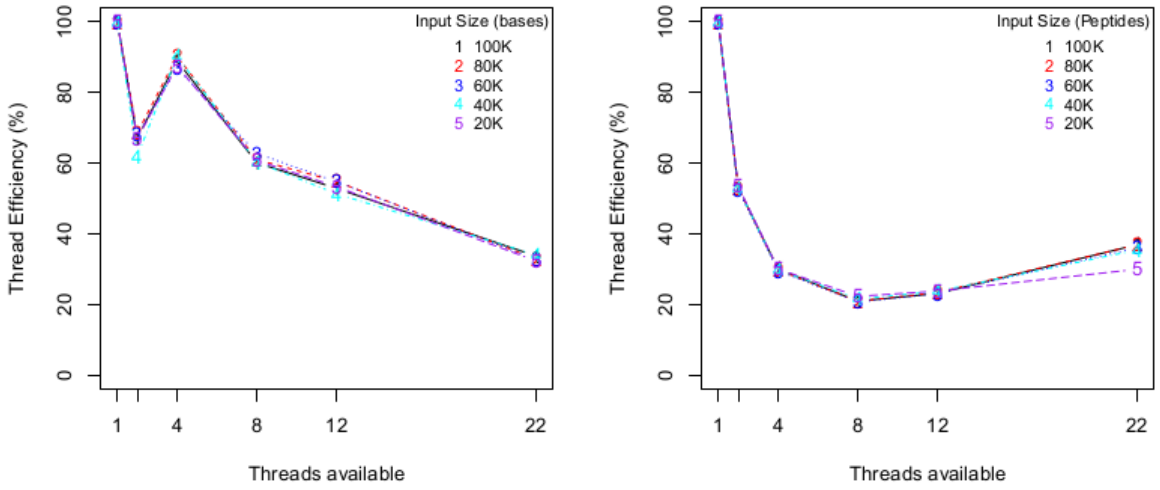

Figure 34. Performance test; Thread Efficiency. Left: DNA input sets, ranging from 20-100K bases sampled from the E. coli reference genome. Right: 20-100K peptides sampled from the Apis mellifera proteome.

The peak RAM usage (see Figure 35), rather than being exponentially related, only increases in a proportional linear manner as the N-mask increases in complexity. This is in part due to the immediate memory deallocation performed on all measured subtrees. Only a single slice of the exponentially increased complexity space need be stored in memory at any given time. Since this test was performed using 12 threads, it would be easy to trade-off performance time for memory usage by decreasing the thread count.

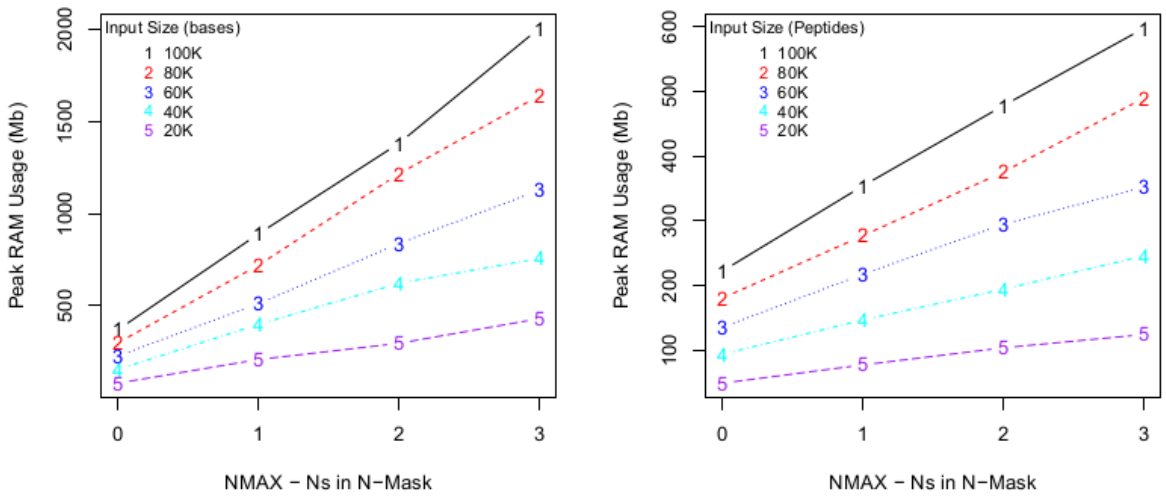

Figure 35. Performance test; Memory Usage (12 threads). Left: DNA input sets, ranging from 20-100K bases sampled from the E. coli reference genome. Right: 20-100K peptides sampled from the Apis mellifera proteome.

The implementation of this method is may still be subject to improvement in terms of computation time and memory usage. Despite this, it is currently enough to calculate signatures for the smaller values of N and has not crashed during testing on several machines.

3.4. Results

Given the exponential time cost of calculating more complex N-masks (as seen in 3.3.7.), the demonstrated application of this program will be limited to values of N at 3 or lower. For the sake of generating inter-comparable signatures, it is also important that all parameters be equal aside from the input set. As in the performance tests, the value of *k* will be 30 in all cases.

3.4.1. Visualisation

Biological information is often only so meaningful as the human eye can comprehend. As the multi-dimensional nature of these signatures does not plot spatially in an intuitive manner in their native dimensions, the visualisations have been flattened into 2D plots, with extra-dimensional information encoded in colour, alpha, and point size. To provide a basic set of interpretive aides for the signatures, the illustrations in Figures 36-38 were created as a reference for users looking at more

complex plots. These can be referred to by the reader whilst viewing later sections. For a quick guide to the colour key see Figure 38.

Another useful visualisation which has been applied to the 3D output matrix is the concept of ‘threads’. As will be shown, the higher dimensional output signatures typically have categories which follow a linear or curved gradient at multiple depths. These categories are usually identical in ‘left seed’ length but increment by one in ‘right seed’ length between depths. For this reason, faint lines have been added to plots which connect all points that observe this single right increment relationship. Figure 8 shows the creation of single thread visualised.

To clarify the meaning of ‘dispersal’ patterns, Figure 36 shows two miniature cases of sequence structures.

The source code written in R for the visualisation functions described here is available in Appendix 2.1.3 ‘Source Code->Visualisation’.

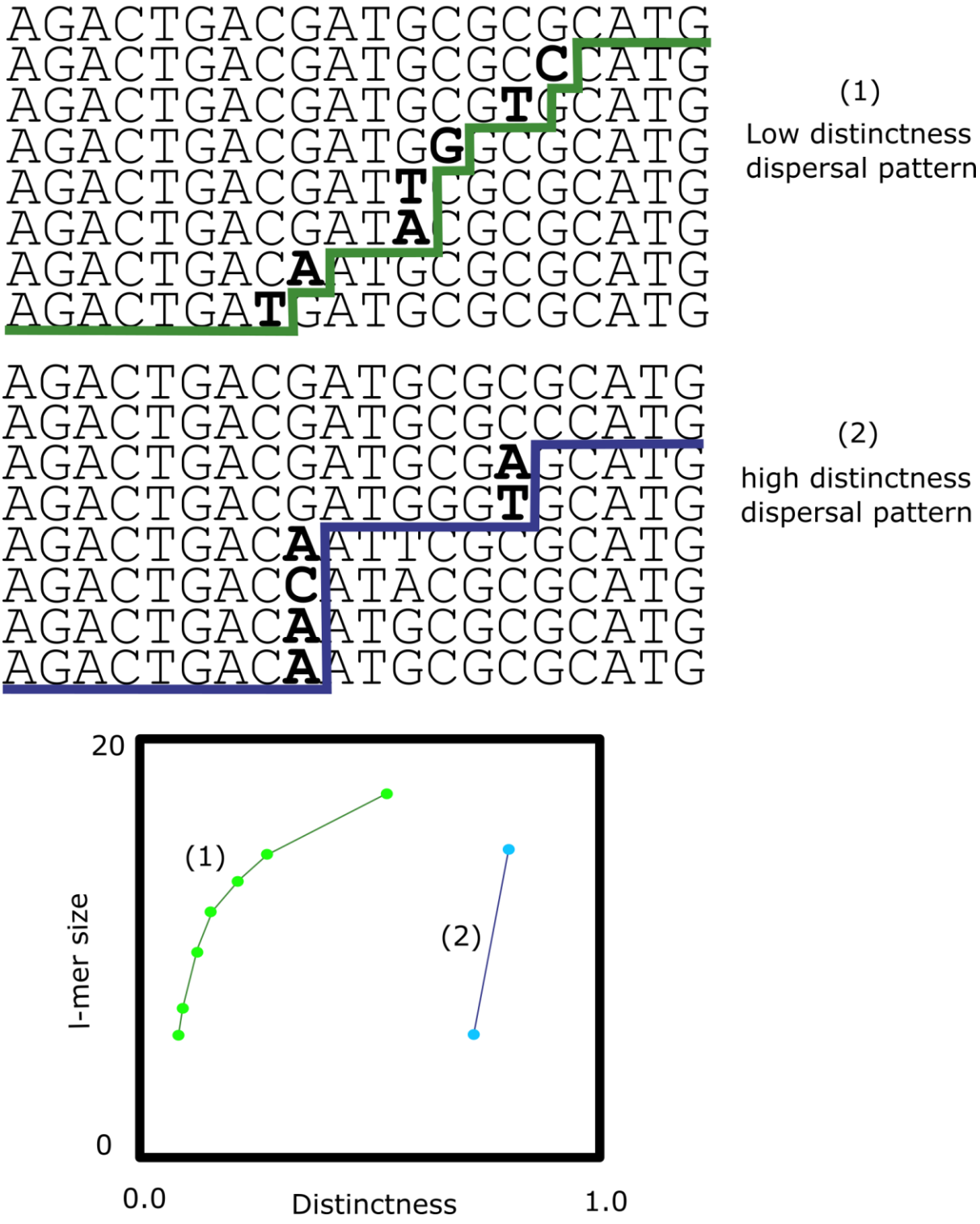

Figure 36. Relationship between unmasked sequence threads and motif variability (not to scale).

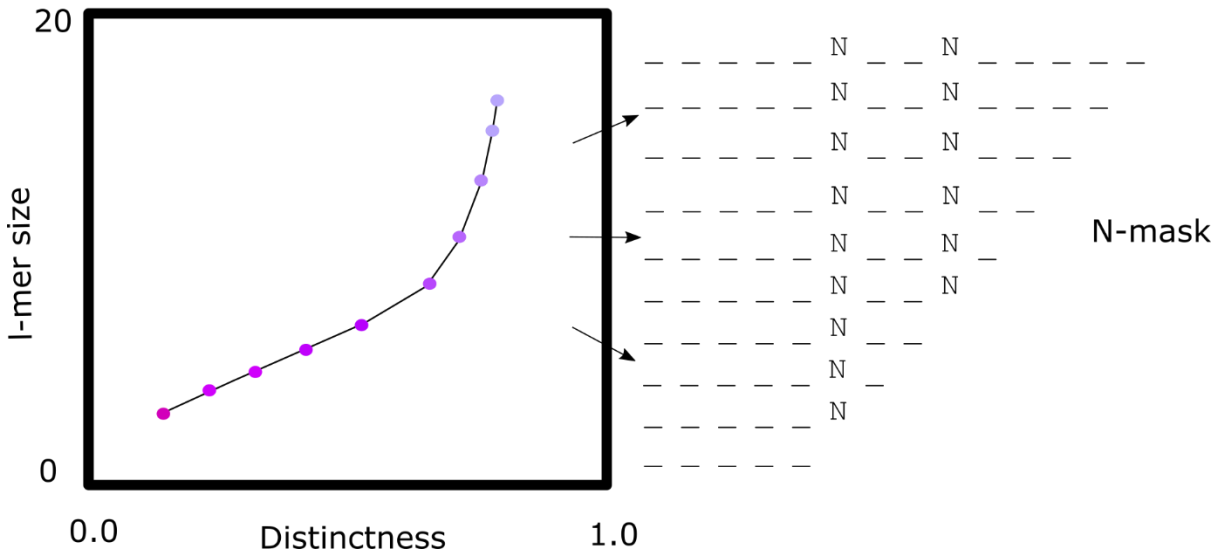

Figure 37. Illustrative relationship between N-masks and threads.

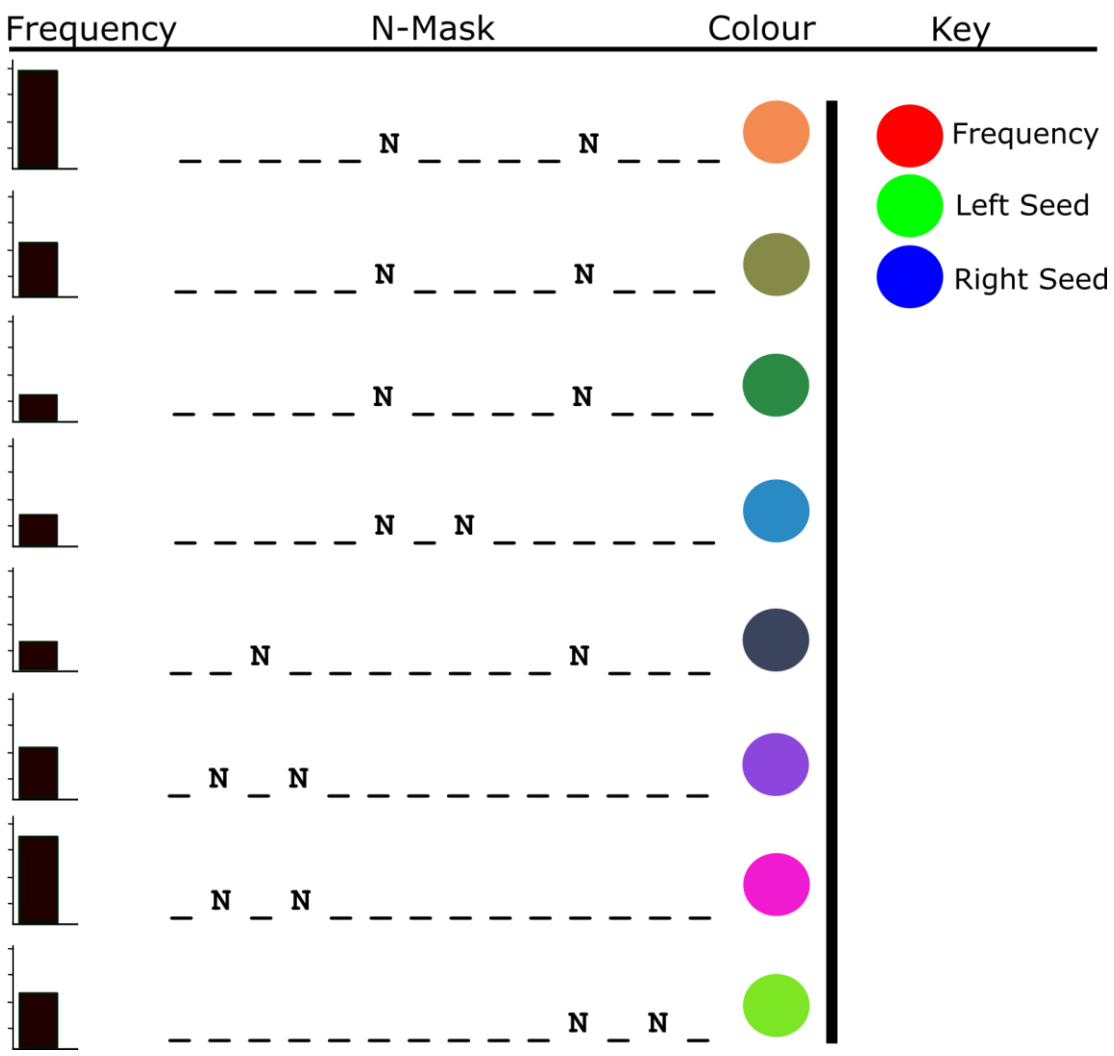

Figure 38. Illustrative relationship between graph colours, seed lengths and l-mer frequency.

3.4.2. Random Case Signatures

To interpret any signature that is produced by this method it is important to understand visually the baseline null case from which all structured sequence inputs will deviate. For this reason, two (DNA and peptide) random sequence noise input sets were tested. Both of 100K characters in length. Since the purpose of the random tests is to establish a null-looking signature, there was one considered difference in generation between DNA and peptides. Random DNA sequences were generated with even base ratios, but random peptide sequences were generated with the average peptide frequencies found in the UniProtKB database (see Table 4).

Table 4. Peptide Frequencies Used in Random Tests (EMBL et al. 2013)

Typical AA Composition of UniProtKB/Swiss-Prot database (%)							
Ala	8.25	GLU	6.75	Met	2.42	Tyr	2.92
Arg	5.53	GLY	7.07	Phe	3.86	Val	6.87
Asn	4.06	His	2.27	Pro	4.7		
Asp	5.54	Ile	5.96	Ser	6.56		
Cys	1.37	Leu	9.66	Thr	5.34		
Gln	3.93	Lys	5.84	Trp	1.08		

The reasoning is simply that the typical peptide ratios vary so greatly between them that even unstructured input sets will universally register higher structures at the top of the tree, unlike most DNA sequence trees, which are expected to be closer to 0.

Figure 39 shows the output of the 2D aggregate matrix (Formula 9) using the local null corrected weighted arithmetic mean distinctness per N, per *l* (as in Formula 21), for the random noise input sets. The typical pattern for distinctness values at *l,* as they ascend beyond saturation depths, is to move swiftly towards 1 (the value found when a frequency 2 branch splits). The return to zero is thus indicative that there are no more structures to be measured for distinctness in the entire tree at this point. Even a very small and improbable number of >1 frequency branch will cause a distinctness mean to be recorded.

Noticeably, the random DNA tree continues to find some measurements of structure even as high as *l*=22 when N=3 (effective minimum sequence space of size 4^{19}). While highly improbable, the frequencies of these small structures may also be due to the slightly inconsistent effects of pseudorandom number generation.

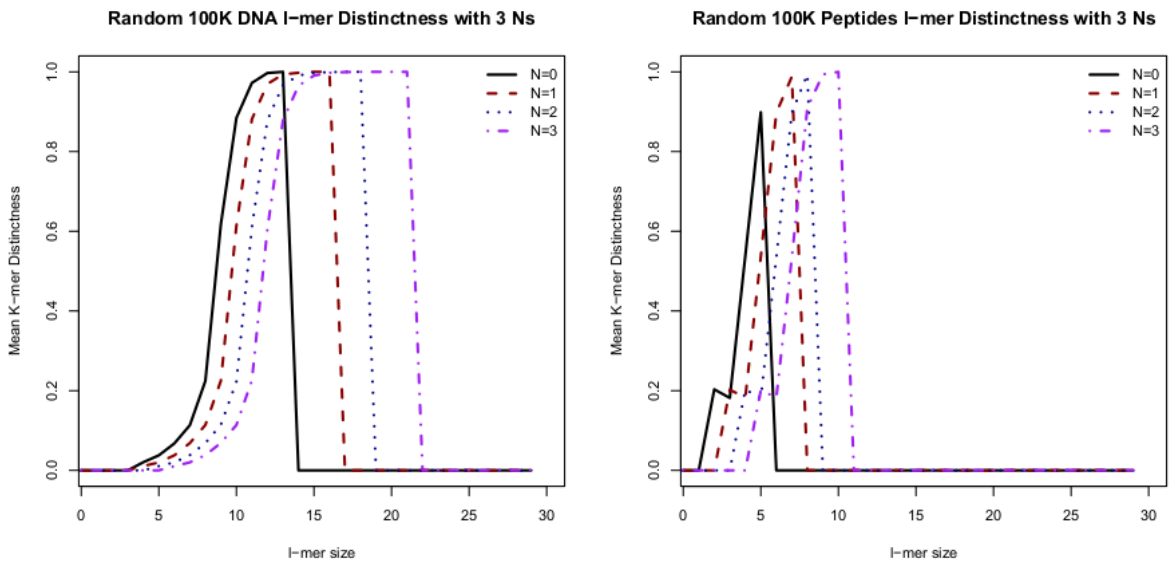

Figure 39. 2D Structure matrix outputs for null cases. Left: 100K DNA bases, Right: 100K protein peptides.

The random peptide tree, with its greater sequence space, loses all structure very quickly in comparison.

The null dispersion of frequency can be seen very clearly in Figure 40, with the value of N only slightly modulating the depths at which the sequences disperse. The relationship between saturation and distinction scores also is clearly displayed. At 200Kb (100Kb input + 100kb reverse complement), the null average saturation depth is approximately 8.8. It is only after this depth that the cohort of means begins to show the results of the dispersal of the set of structures retained by chance. Naturally, as the depth gets lower, the probability of any given structure retaining enough frequencies to disperse amongst the child-nodes decreases exponentially. This also applies to the parent nodes of by-chance dispersals, meaning that the calculation of $(1\text{-}D_{parent})$ component of the calculation of D_a (Formula 6) is far more likely to also be 1. It is this relationship with drives the distinctness curve to 1 in the null/random case.

Looking at the random peptide output, Figure 41, the curve is similar in shape but occurs far more rapidly, as in Figure 39. One positive aspect of the null peptide signature is that we can reasonably expect almost all structures recorded above $2 * log_{20}(f_r)$ to be the result of actual biological effects.

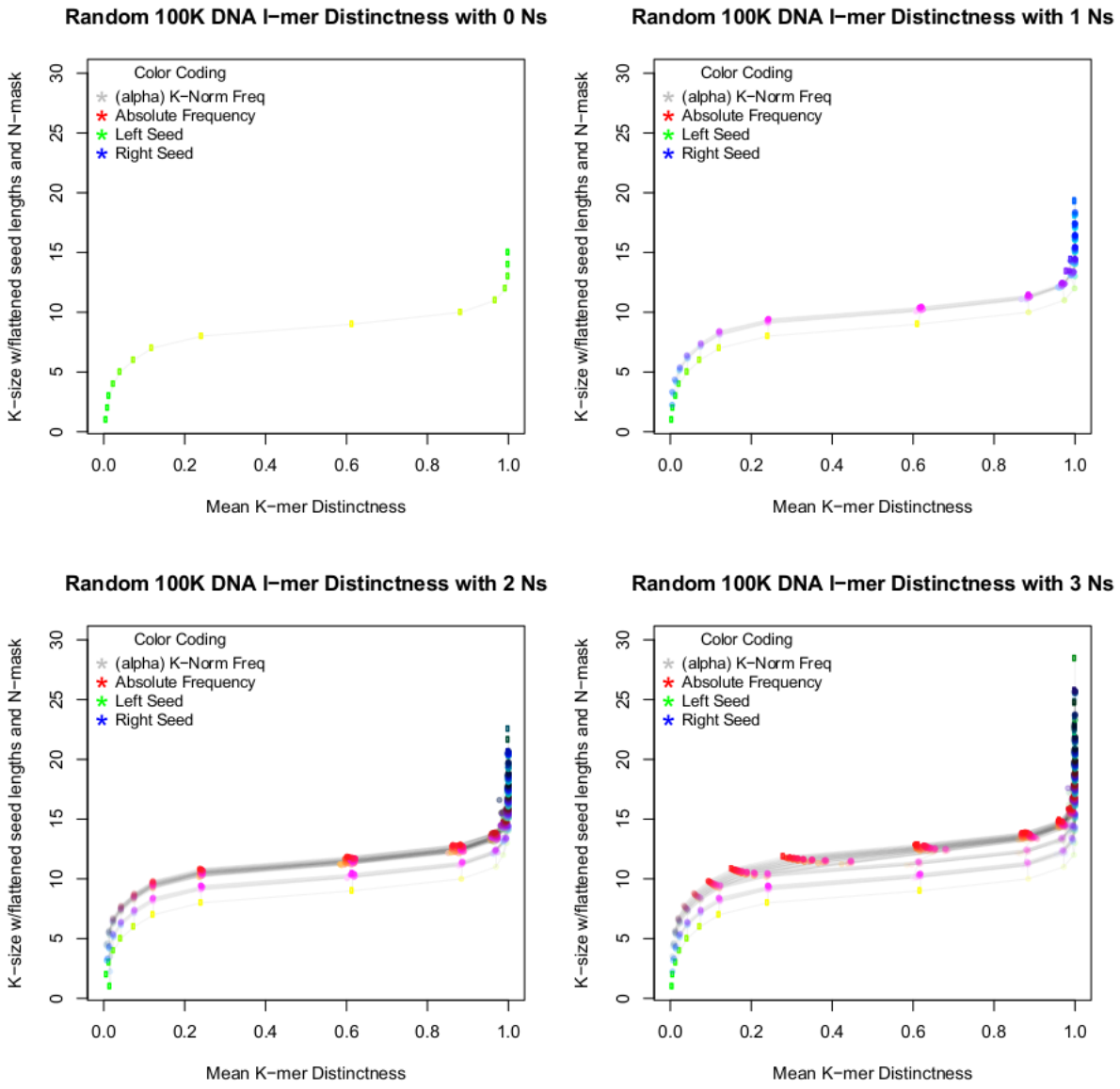

Figure 40. 3-Dimensional output signatures for random DNA. 100Kb random sequence used in each execution, visualisation of four values given for parameter N. Top-left: N=0, Top-right: N=1, Bottom-left: N=2, Bottom-right: N=3.

Insofar as these graphs inform our interpretation of other plots, we should make note of the natural signature of null sequence dispersal and attempt to distinguish it from structured dispersal. For the DNA graphs we observe the steepest part of the 0-1 distinctness curve beginning near the saturation depth, the tendency towards 1 at the top of the signature, and the tendency towards zero near the start. This shape will be referred to as the 'DNA null curve' in discussion of later plots. We should observe therefore the modulations of the null curve as biological signatures. Similarly, the pattern of natural effects which occurs in the peptide graphs, as discussed at the start of this section, varies slightly. We observe the head of the tree commencing at ~0.4 distinctness, moving quite sharply lower, and reversing after the saturation depth to curve back towards 1. Again, this will be referred to in later sections as the 'peptide null curve'.

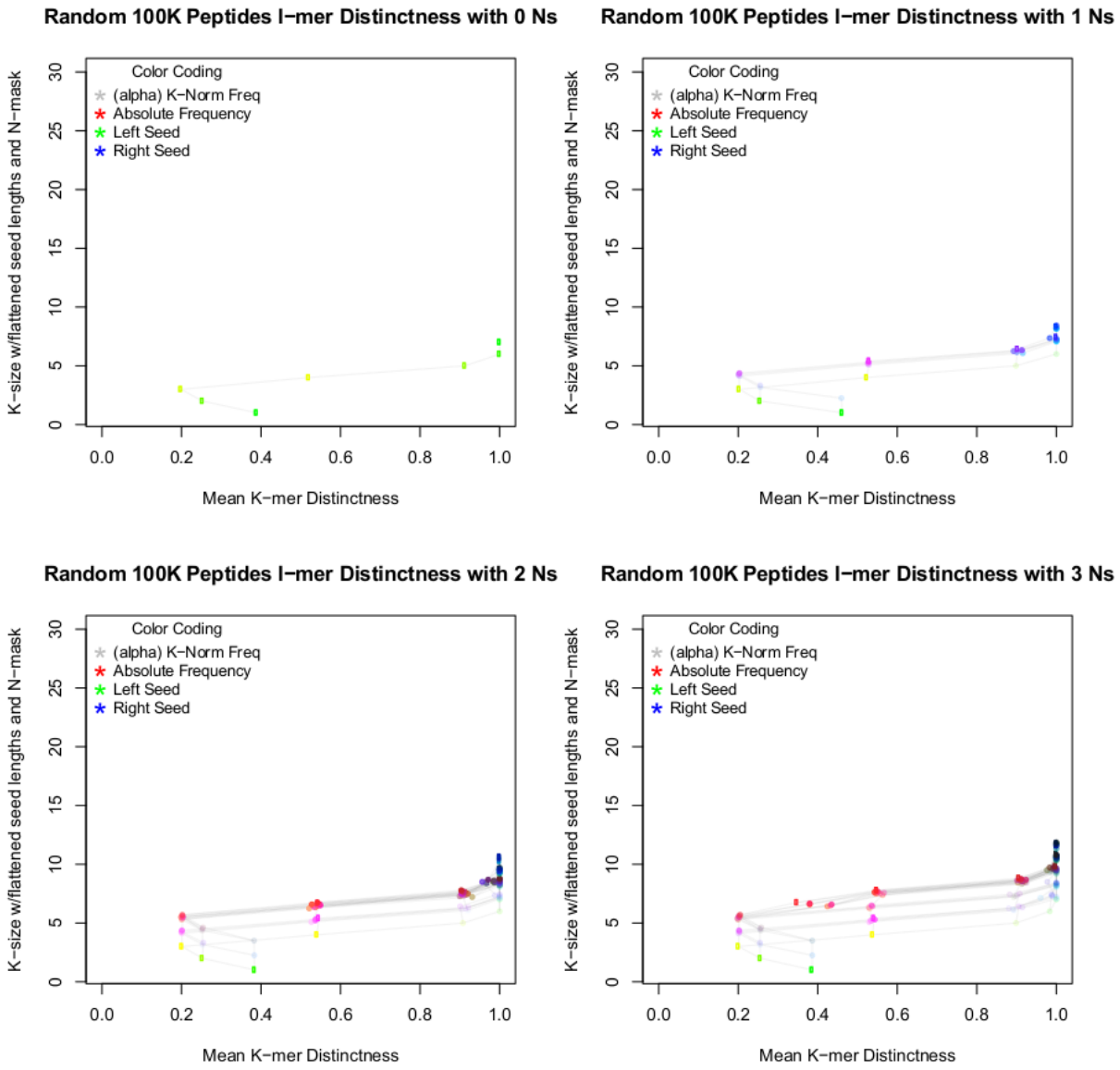

Figure 41. 3-Dimensional output signatures for random peptides. 100K random peptide sequence used in each execution, visualisation of four values given for parameter N. Top-left: N=0, Top-right: N=1, Bottom-left: N=2, Bottom-right: N=3.

3.4.3. Small Subset Signature Tests

The next series of tests involves using subsets of biological sequence at the same scale as the test set (100K characters). There was no additional randomisation of subset, in both cases they were selected under the conditions of being the first 100K characters in the files they were extracted from. The two source material files were as such*: Apis mellifera* proteome retrieved from Uniprot (EMBL et al. 2013), and *Escherichia coli* reference genome retrieved from the NCBI genome database (NCBI 2016).

The objective of these tests is to examine the way in which the null curve begins to change when biological sequences are used, with relatively low structure in the input. In the case of ‘omic scale

datasets, many of the sequence structures that might be found are only discoverable in the context of the entire set. For example, a 30-mer which occurs only five times in the genome is unlikely to be present more than once in a 2% subset. By extension, we can suggest that whilst these small subsets of sequence will be more structured than random, the actual discoverable structure ought to be on a much lower scale than in a typical input set. This makes them a good 'stepping stone' between the random signatures and full input sets. The signatures developed here are purely aggregates of weighted mean distinctness scores and have not been subject to 'local-null' corrections.

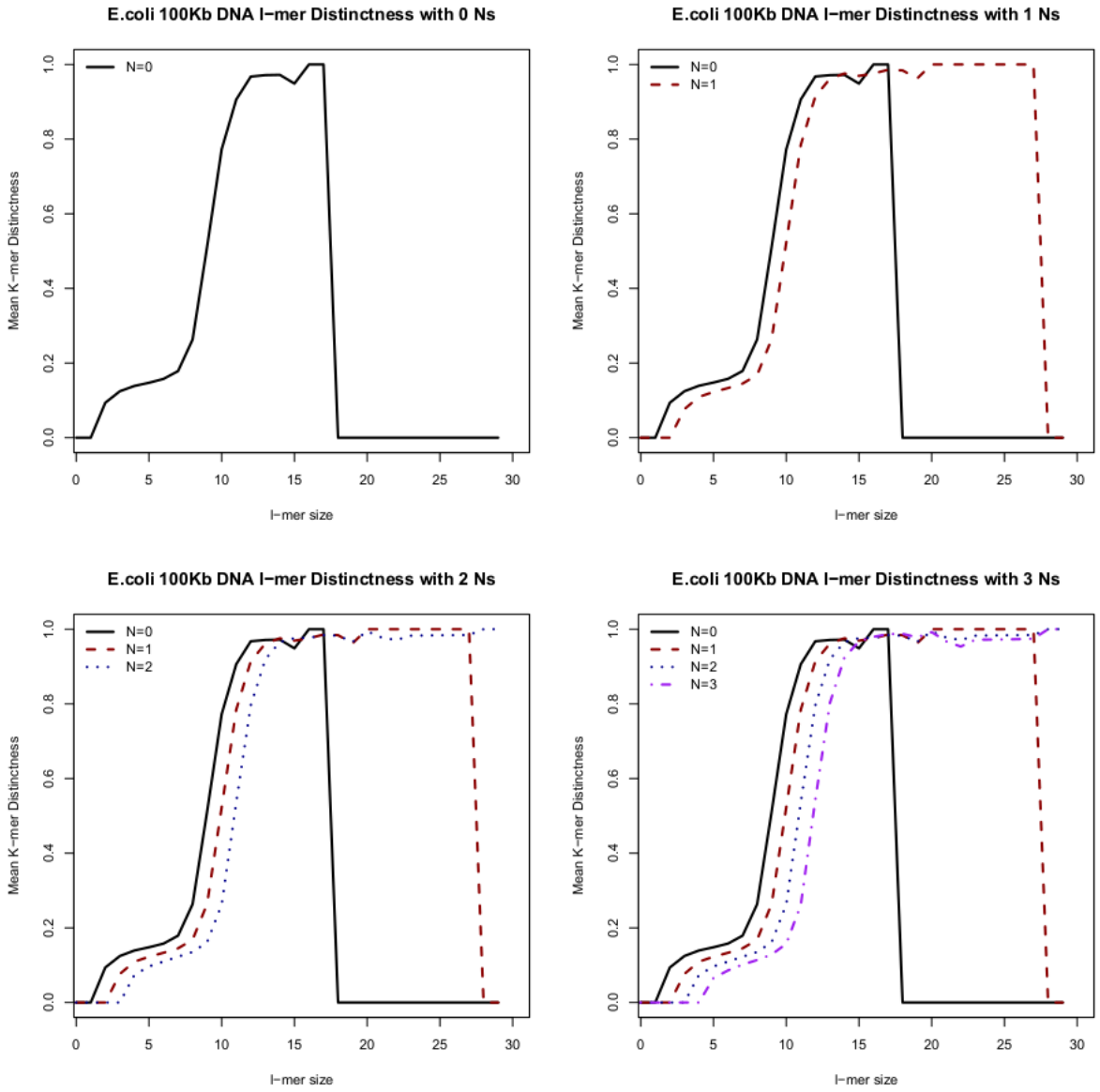

Figure 42. E. coli 100Kb subset 2-Dimensional signature output. Top-left: N=0, Top-right: N=1, Bottom-left: N=2, Bottom-right: N=3.

Figure 42 is directly comparable to the Figure 39 (left), this is to say that the N=0 plot follows a similar pattern with two exceptions, a faster ascent in the saturation depths and a longer reach into

the unsaturated depths (14 vs 18). The effects of introducing Ns has a far more marked effect. N=1 only loses all structure at *l=28*, and the other values of N continue to find low frequency merged long l-mers throughout the set. This speaks to the fragility of long substrings in biological sequence more generally, and would be expected concordance with the development of gk-SVM (Ghandi et al. 2014), as discussed the introduction.

Figures 42-45 are all subsets of larger permutation tests. Their expanded paired images are available in Appendix 2.5, as IMG1-4 respectively.

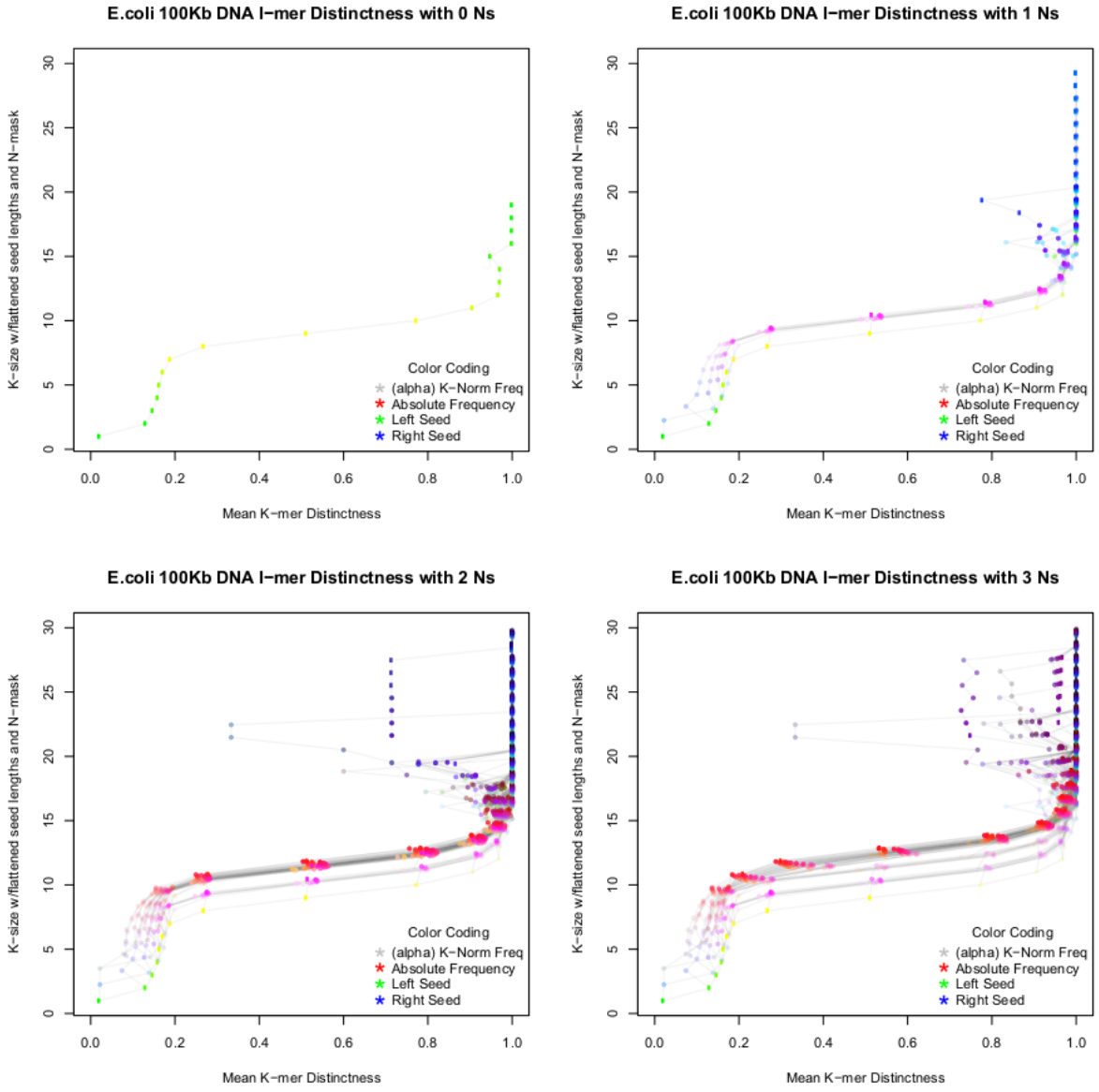

Figure 43. E. coli 100Kb subset 2-Dimensional structure output. Top-left: N=0, Top-right: N=1, Bottom-left: N=2, Bottom-right: N=3.

The 3D matrix outputs (Figure 43) begin to show in more detail the N>0 results found in Figure 42. The braid-like structures forming at the saturated depths show that the detection of uneven 2-8mer

substring frequencies becomes possible at this scale. The masked-index category threads here tend to repeat earlier unmasked, or lesser masked threads at higher depths. The post-saturation gradient is still largely present, however the component which collapsed at 1 in the null curve is showing various relatively distinct medium-to-low frequency N-masked complex structures with long right seeds at the higher depths. This is an example of how the specificity of the signature allows direct description of the type of flexibility found in the reference structures.

The peptide 2D subset test (Figure 44) produces a substantially difference result to the null test. With saturation depths typified by an early spike in distinctness followed by a curve which tends slowly higher. The peptide null curve tendency to return to lower mean distinctness immediately following saturation is repeated here, however the dispersal of frequencies seems to be far more gradual for each structure. A case where a frequency-50 12-mer loses 10% of its frequencies per depth, would be typical of a sequence structure that drags the mean distinctness towards 0.1, as can be seen here.

What this suggests biologically is a set of similar sequences which are each dissimilar from each-other in different ways, suggesting that an N-mask would struggle to reunite them at longer for fragile values of *l.* The opposite case would be a set of sequences which all differ a one or two fixed location, this would disperse over far fewer depths, generation very high distinctness scores.

The rapid spike towards 1 demonstrated by the N>0 should also be considered more the effect of the terminal-k depth backward subtraction process described in 2.2. Figure 44 is also directly comparable to Figure 45 in shape. However, Figure 45 also begins to show another feature related to the single right seed extension per depth relationship discussed in 3.4.1., threading, and a certain banding pattern of threads. A banding pattern can be described as a case where multiple threads cluster into a single channel. To understand banding, consider the opposite cases described above, of high frequency structures which typically disperse either over many depths, or only over one or two, as in Figure 36. Bands represent specific biological prominences in the modes of structure dispersal within that range. This might suggest evolution acting differently on several different types of sequence motifs. Some motifs are flexible in a highly regular manner, these may present as higher distinctness bands, some motifs have the evolutionary flexibility to diverge at almost any base, just so long as most of the sequence remains similar, these types of sequence structure will manifest more as bands towards the lower distinctness range. The number, and complexity of the bands, are thus to be read as indicative of the prominence of sequence structure types exhibiting separate modalities of evolutionary change.

For example, the N=3 graphs in Figure 45 shows in the 17-23 *l*-mer range, an unusually distinct set of structures typified by a relatively small region of flexibility and a long right seed. This structural category decomposes in a slightly more homogenous manner that the other content of the test set.

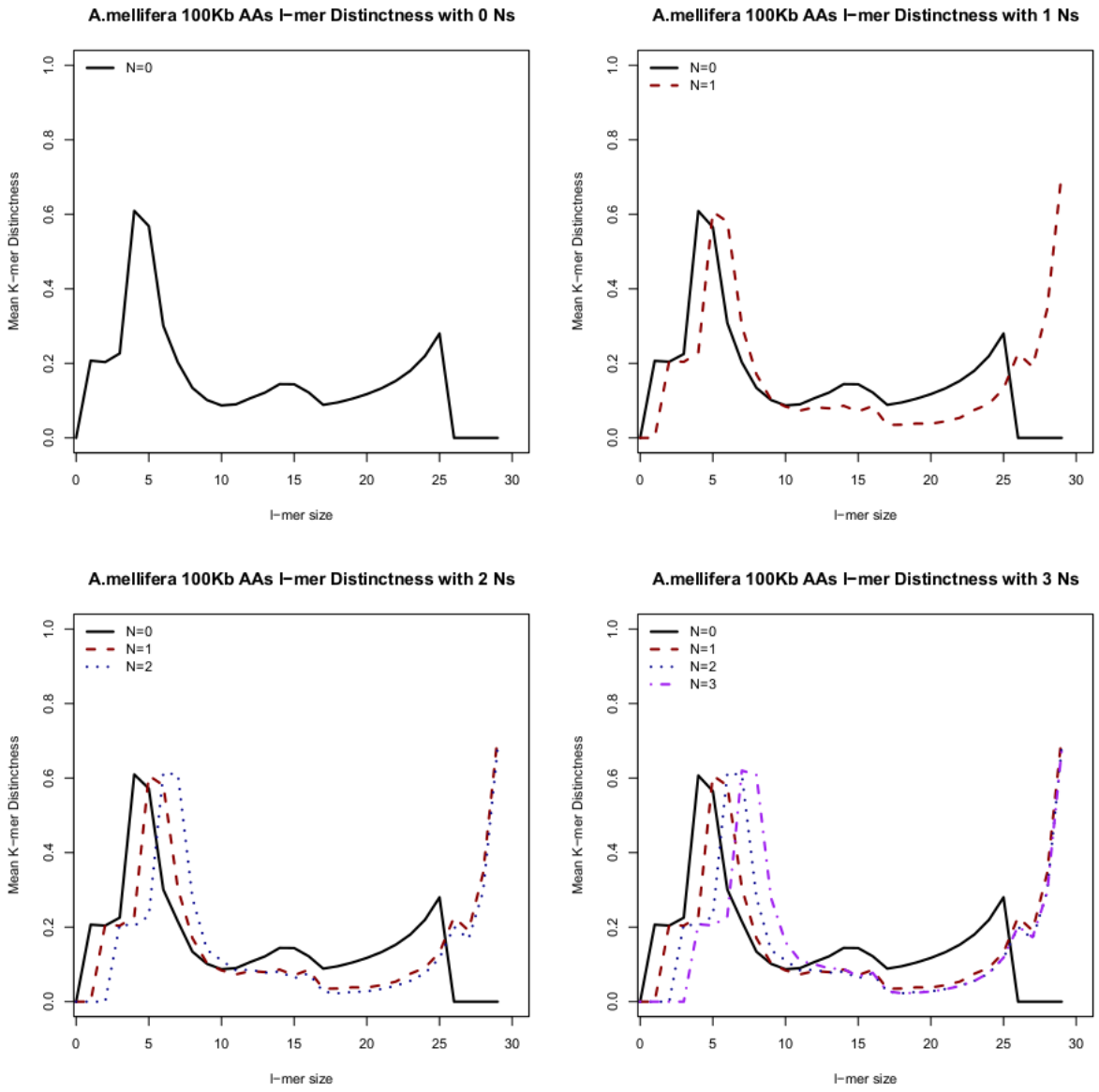

Figure 44. Apis mellifera 100K AA 2-Dimensional structure matrix. Top-left: N=0, Top-right: N=1, Bottom-left: N=2, Bottom-right: N=3.

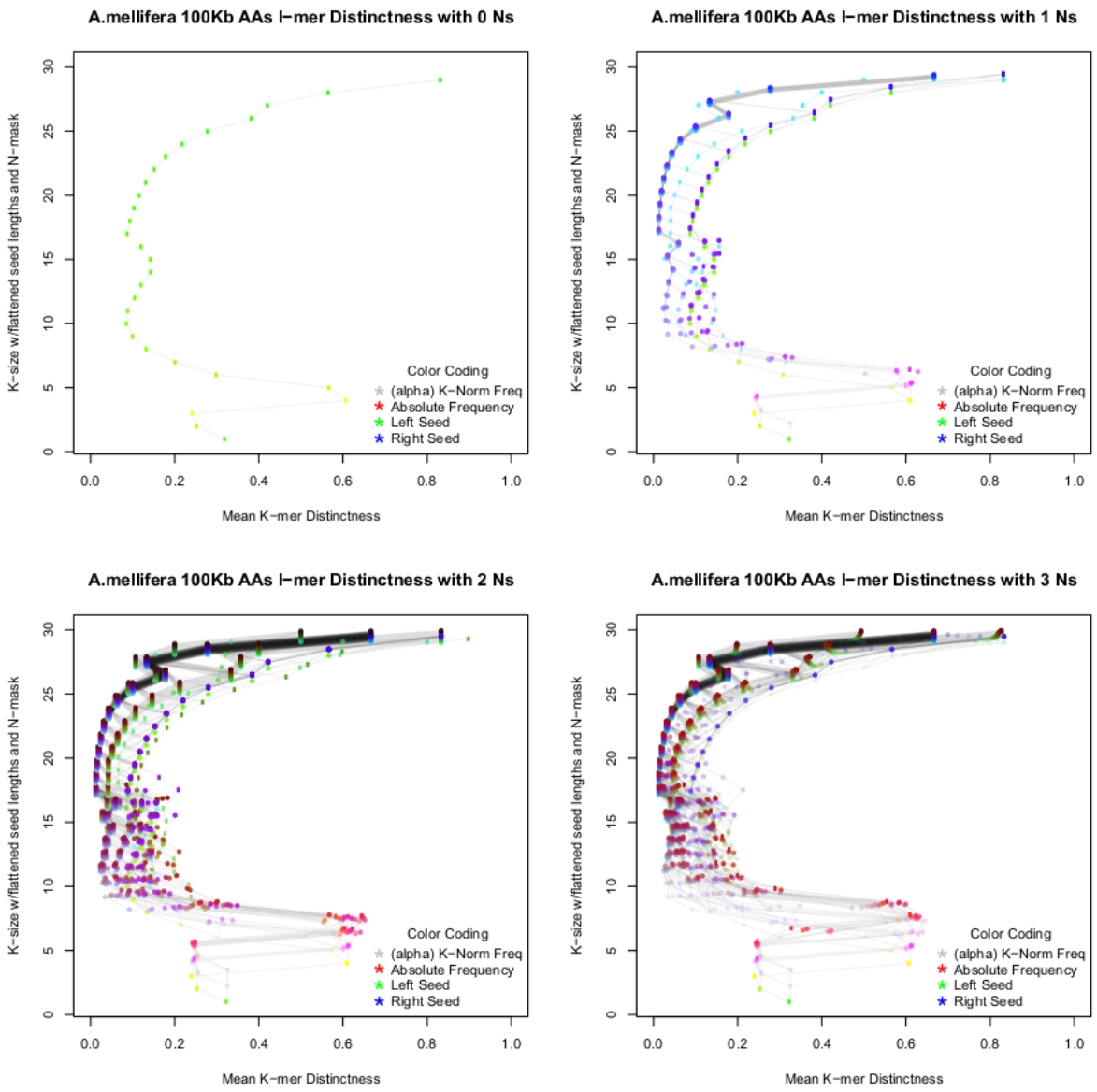

Figure 45. A. mellifera 100K AA 3-Dimensional structure matrix. Top-left: N=0, Top-right: N=1, Bottom-left: N=2, Bottom-right: N=3.

3.4.4. Test Set: Invertebrate Proteome Signatures

This is the first of three test sets designed to explore the ways in the which the signatures can be used to interpret biological data types. This test set also serves to further explore the peculiarity of *L. rubellus* in comparison to relative to three other annelids, and an arthropod. *Capitella teleta, Helobdella robusta* and *Apis mellifera* reference proteomes were all retrieved from Uniprot (EMBL et al. 2013), *L. rubellus* proteome was extracted from a genome assembly discussed in Data Chapter 1. *A. gracilis* proteome was extracted from the genome assembly performed in Data Chapter 3.

Here we introduce the test of Pareto shape parameters and full tree summaries as described in 3.2.5. and 3.2.4. respectively. Table 5 shows the summarisation of the total structure in the trees

built out of these proteomes. In addition to the summary scores, it also shows the result of applying the 'local-null' correction/subtraction discussed in 3.2.6.

Table 5. Proteome Tree Summaries

Species	Structure	Struct. - Sub	Shape	Shape - Sub	Size (aa)	>K Freq (%)
Apis mellifera	0.0227	0.0054	1.581	5.082	5,988,832	1.48
Capitella teleta	0.0286	0.0131	1.920	2.095	10,523,041	7.46
Amynthas gracilis	0.0408	0.0237	1.689	1.947	12,106,353	11.61
Helobdella robusta	0.0286	0.0137	1.614	2.671	8,079,707	3.36
Lumbricus rubellus	0.0313	0.0115	1.624	2.684	7,920,655	18.13

The final column in Table 5 also shows the percentage of the sequence structures which did not disperse within the tree and were negated by the terminal subtraction process described in 3.2.2. Interestingly the only non-annelid in the set has the lowest escaped frequency count by far, with *rubellus*, arguably the most difficult genome to analyse conventionally, showing a huge quantity of escape frequency – this suggests a very large number of either recent protein family expansions, or sufficiently divergent allelic copies, which might be right answer given the conclusions drawn in Chapter 3. For example, an >*K* Freq %-score of 50 could be achieved by every sequence being duplicated once identically.

The Pareto shape results are interesting for how consistent they are despite considerable changes across the rest of the scores. This indicative that the distributions of structure in these proteomes has a very steep pareto curve, to see simulated Pareto distributions which demonstrate the meaning of the shape parameters, see 3.4.7., and Figure 70.

Many of the other results in Table 5 are particularly informative when paired with the signature. Rather than describing the rest of Table 5 in depth independently, it will be frequently referred to in the following summary of the five 3D signatures. Each of the signatures also represent the post local-null correction.

The data representation provide in Figure 46, reveals further evidence of how the odd-one out in this set (in evolutionary distance) is also substantially difference in sequence structure. However, first it is also necessary to cover the usage of absolute and relative frequencies in this plot. The individual absolute categorical frequency density is coded to the red component of the pixels. Typically, we would expect the lower *l*-mer categories to be denser in frequency as the sequence space is substantially smaller, and the categories far fewer in number. However, towards the saturation depths the absolute frequencies will be lower for two reasons: 1) local null subtraction 2) absolute null correction, both of which are described in 3.2.6.

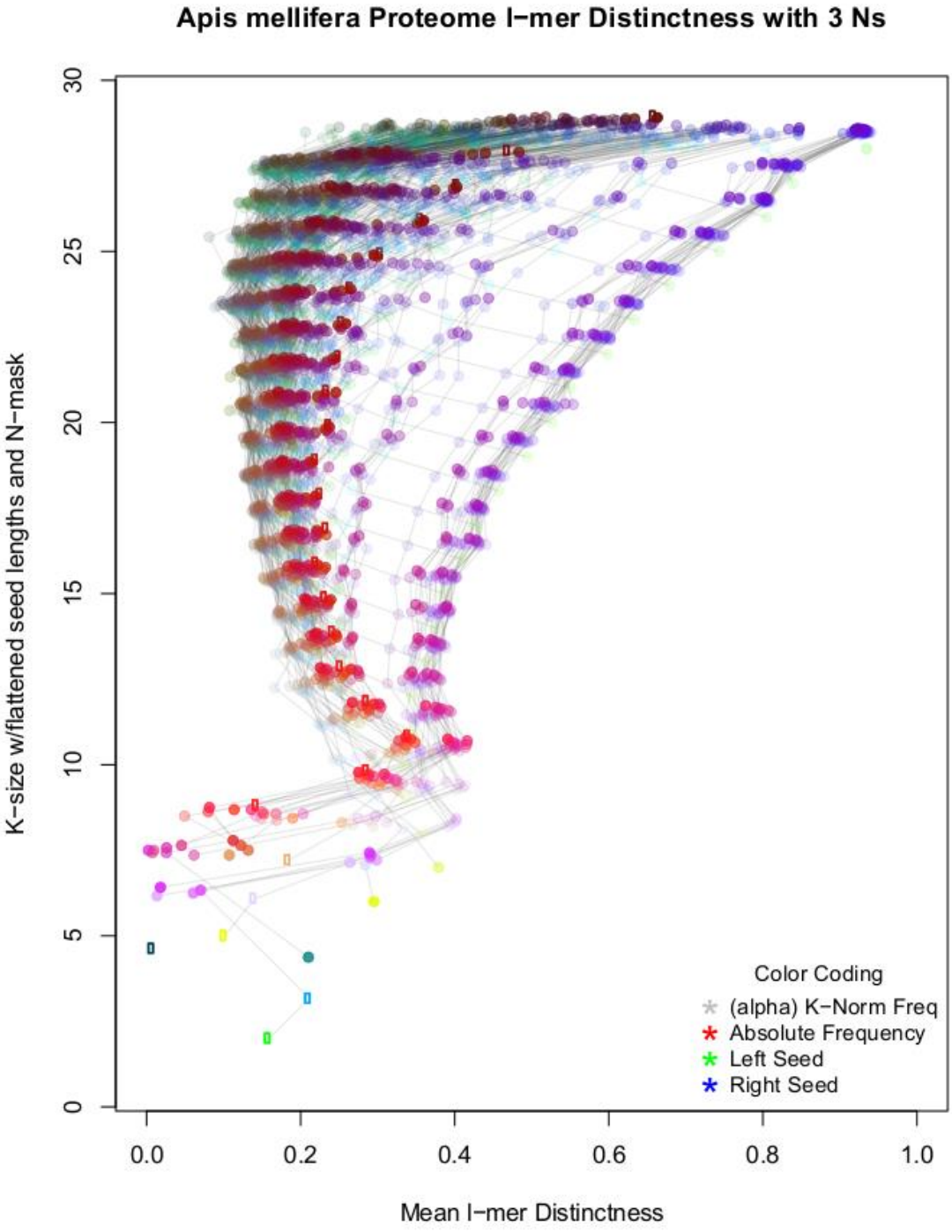

Figure 46. Apis mellifera full proteome 3D-signature.

Secondly, frequency scaling across a single depth has been coded to the alpha-value (transparency) of the point. This allows the user to see the relative quantities composing various features. For example, if we were to observe the faint single-thread proceeding from the left of the rightmost band from depths 11-28. It seems insignificant; however, this is likely the effect of one large, or a small number of similar domain types which have a very specific dispersal profile. Other similar

threads following their own patterns can also be seen further into the signature. One of the flaws of this visualisation methods is that much of the complexity discovered is packed into broad but dense bands of distinctness, making threads impossible differentiate, leaving only colour gradients as informative.

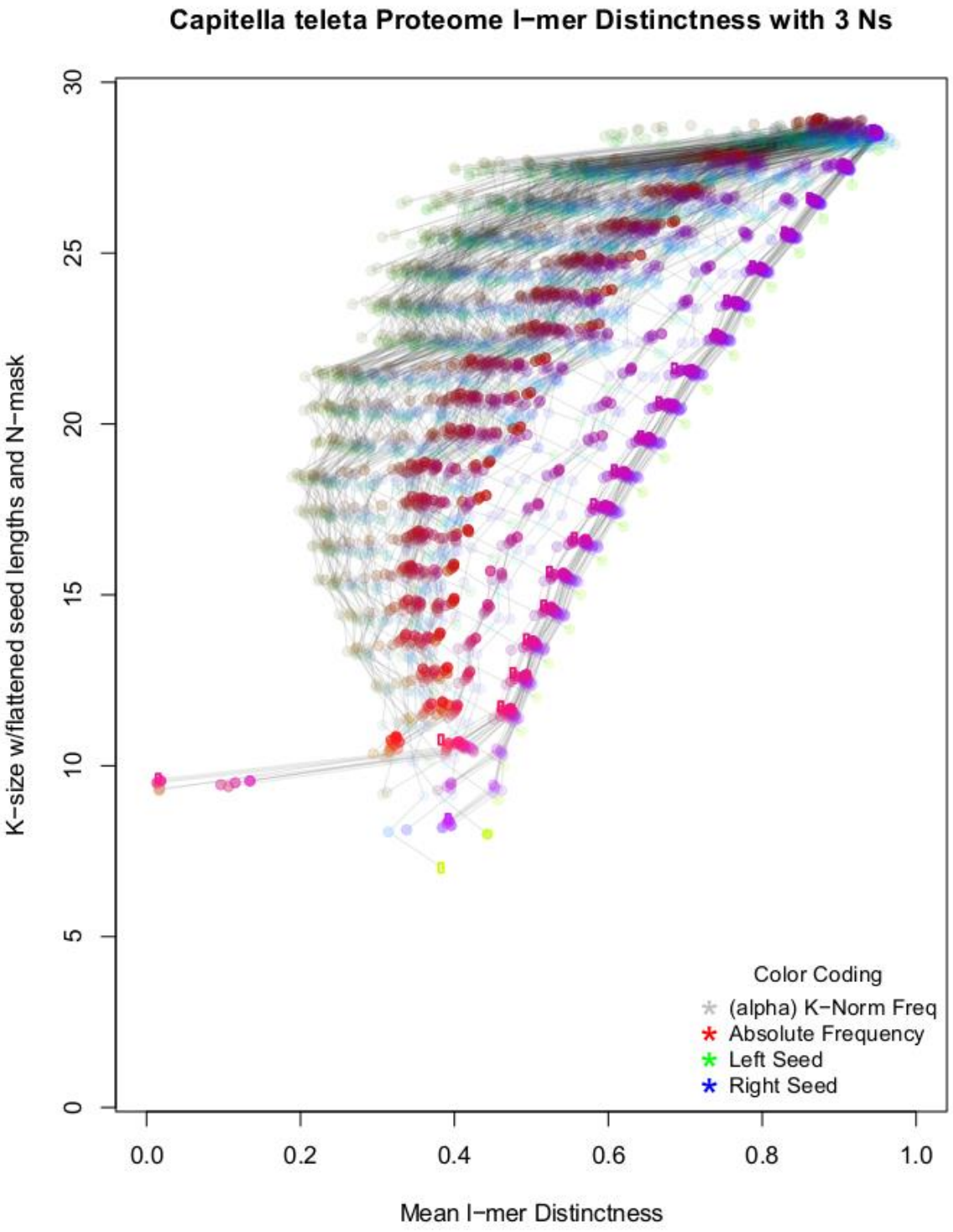

Figure 47. Capitella teleta full proteome 3D-signature.

Comparing Figures 47 and 48 shows the threading complexity which is visually collapsed in *mellifera* and still very hard to discern in *teleta.* Another feature which is particularly prominent in Figure 47 but is present in all proteome signature to some degree is the band separation between a narrower rightmost group, and a much broader leftmost group. Typically, the rightmost group is represented by categories with either a very long left or right seed, and perhaps only a single N in the mask. This can be thought of as the scaled modularity of the motifs which are the most resistant to variation. A rightmost band moving quickly towards 1 suggests a large portion of unique/non-duplicated sequence, or low frequency fragile long motifs. The width of the gap is perhaps more descriptive of distances between similar motifs groups, rather than the conservation patterns within the most uniform.

Returning to Figure 46, there are several datapoints which resonate in the interpretation from Table 5. The signature has the greatest tendency to curve towards 0 of all the proteomes, the tree structure summary is also the lowest of out the set, and when local-null corrected this difference only becomes more pronounced. It has the lowest rate of frequency escape, and the highest post-correction Pareto shape. What this says more broadly about the signature is that it is likely to have many smaller groups of internally homologous sequence motifs, rather than fewer larger groups (proportionally speaking). It is also the case that the most common protein domains are less likely to be disproportionately overabundant in the set. This could be summarised as a 'high complexity, low structure' proteome, insofar as structure is defined in terms of stacked sequence homologies.

Figure 47 and 48 both represent the tied 2nd most unstructured proteomes after *A. mellifera,* although *H. robusta* shows a much greater tendency towards the heterogenous structure dispersal that *C. teleta* this could reflect the higher shape score of *robusta. Capitella* also has a more differentiated set of deeper banding patterns across the categories of medium length left and right seeds, suggesting a wider variety of motif forms.

Looking at the earthworm plots (Figures 49 and 50), we can see a much greater range of banding patterns, particularly in the case of *L. rubellus* which seems to have a combination of a great many diverse smaller structures which can be merged into lower distinctness, higher frequency categories. Most interestingly the type of N-mask applied appears to have a very significant effect on the dispersal patterns, this might be suggestive of large-scale gene family expansions which diverged in several different ways and would also make some sense of the very high rate of frequency escape in Table 5.

One caveat to the 'banding spread equals diversity' argument is that dispersal patterns generating similar levels of distinctness need not originate from the same type of structure, it is only more

apparent when they are more spread out. Additionally, within the narrower, denser bands of many threads there may also be structure which is simply too densely arranged in these plots to be discernible. For this reason, additional visualisations have been generated, using a z-axis to expand these thicker leftmost banding patterns. Figures 51 and 52 show two versions of this additional dimensionality. They have the advantage of separating thick bands when rotated suitably, but the simultaneous disadvantage to obscuring other parts of the plot. To present the data in another form which attempts to make maximum advantage of the 3D plot, a series of animated rotations of the 3D plots for each image have been produced.

Appendix 2.4 contains ANIMATION2-11, which display the five proteome signatures rotating around multiple axes, both with and without 'thread' lines.

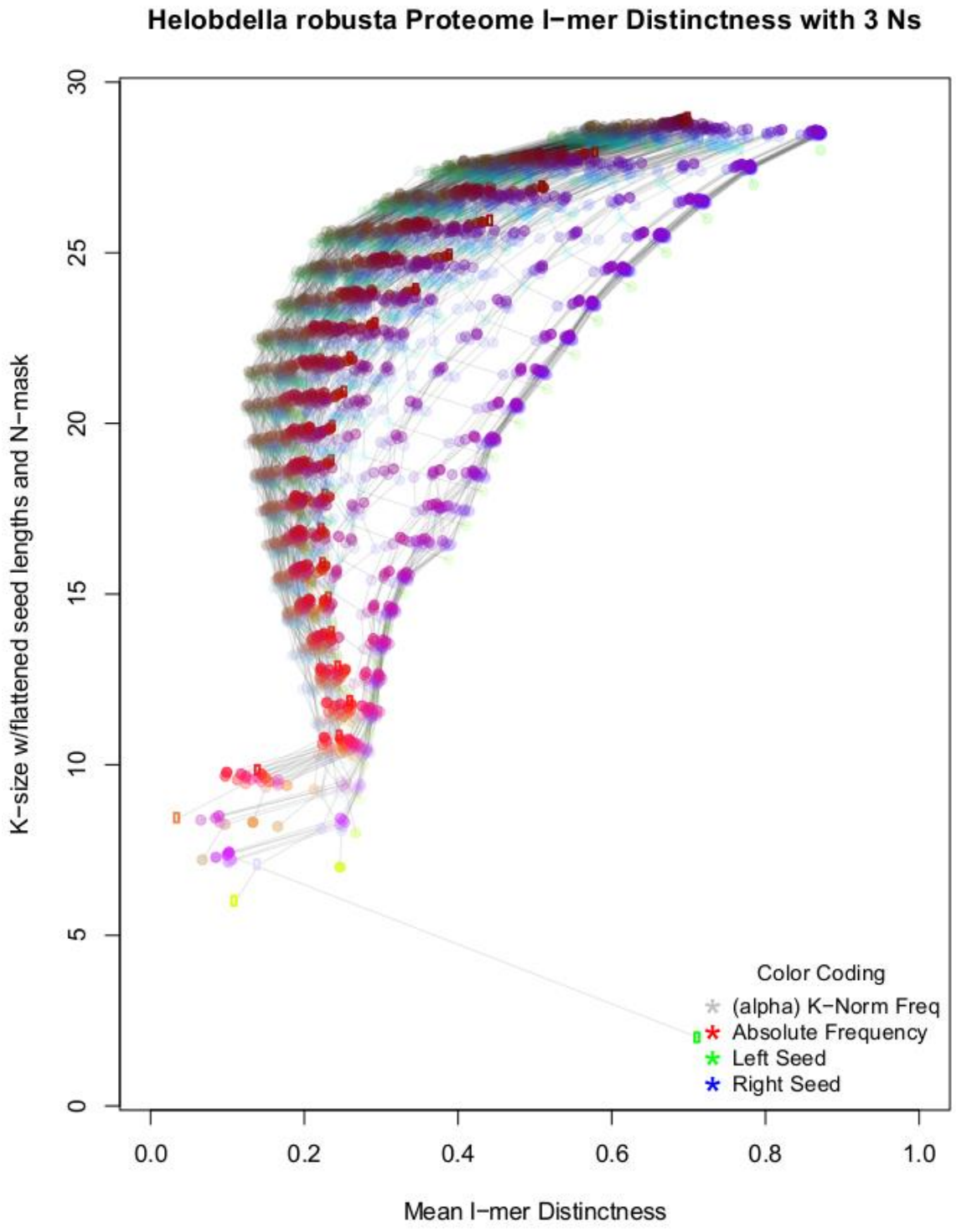

Figure 48. Helobdella robusta full proteome 3D-signature. N=3.

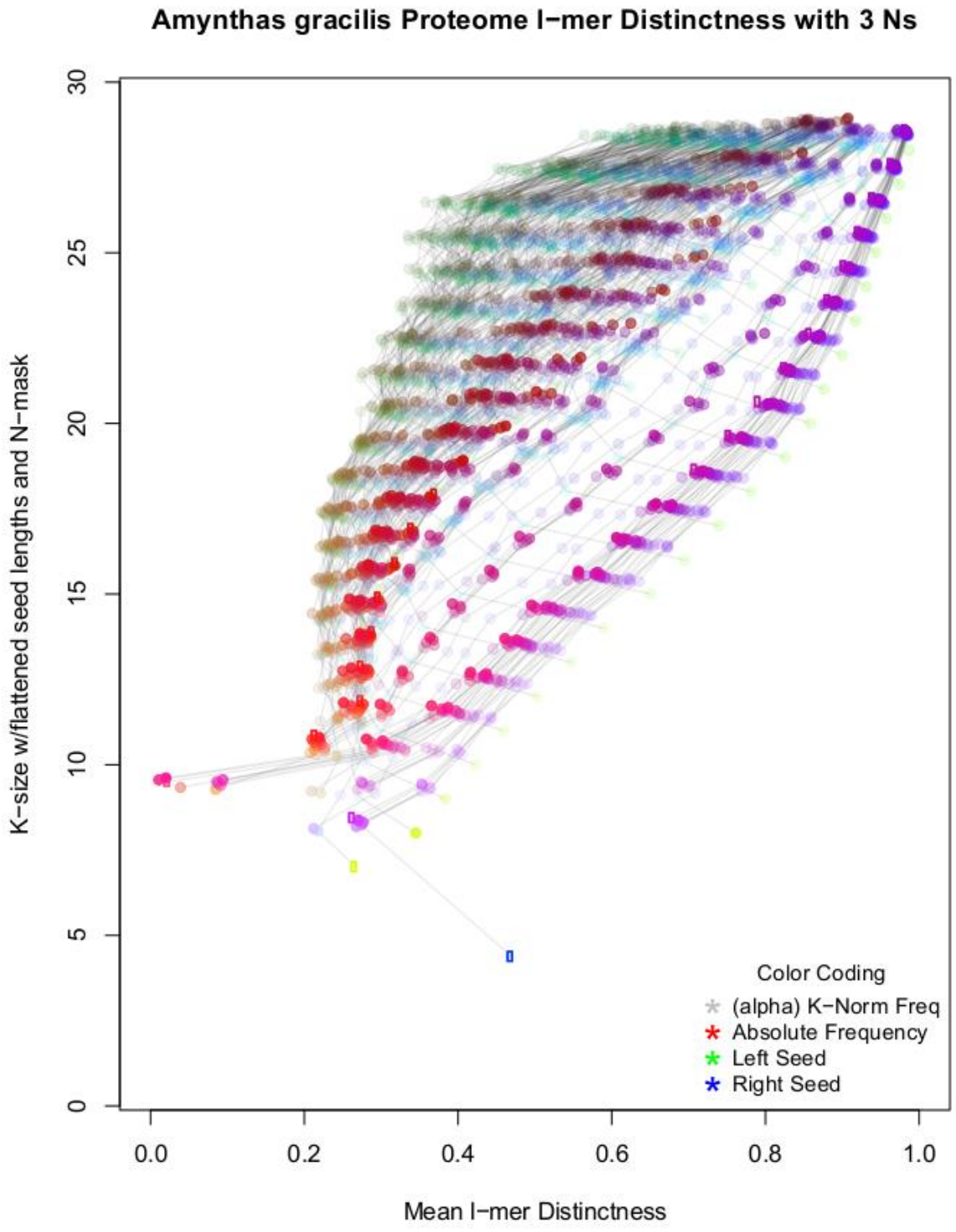

Figure 49. Amynthas gracilis proteome signature. N=3.

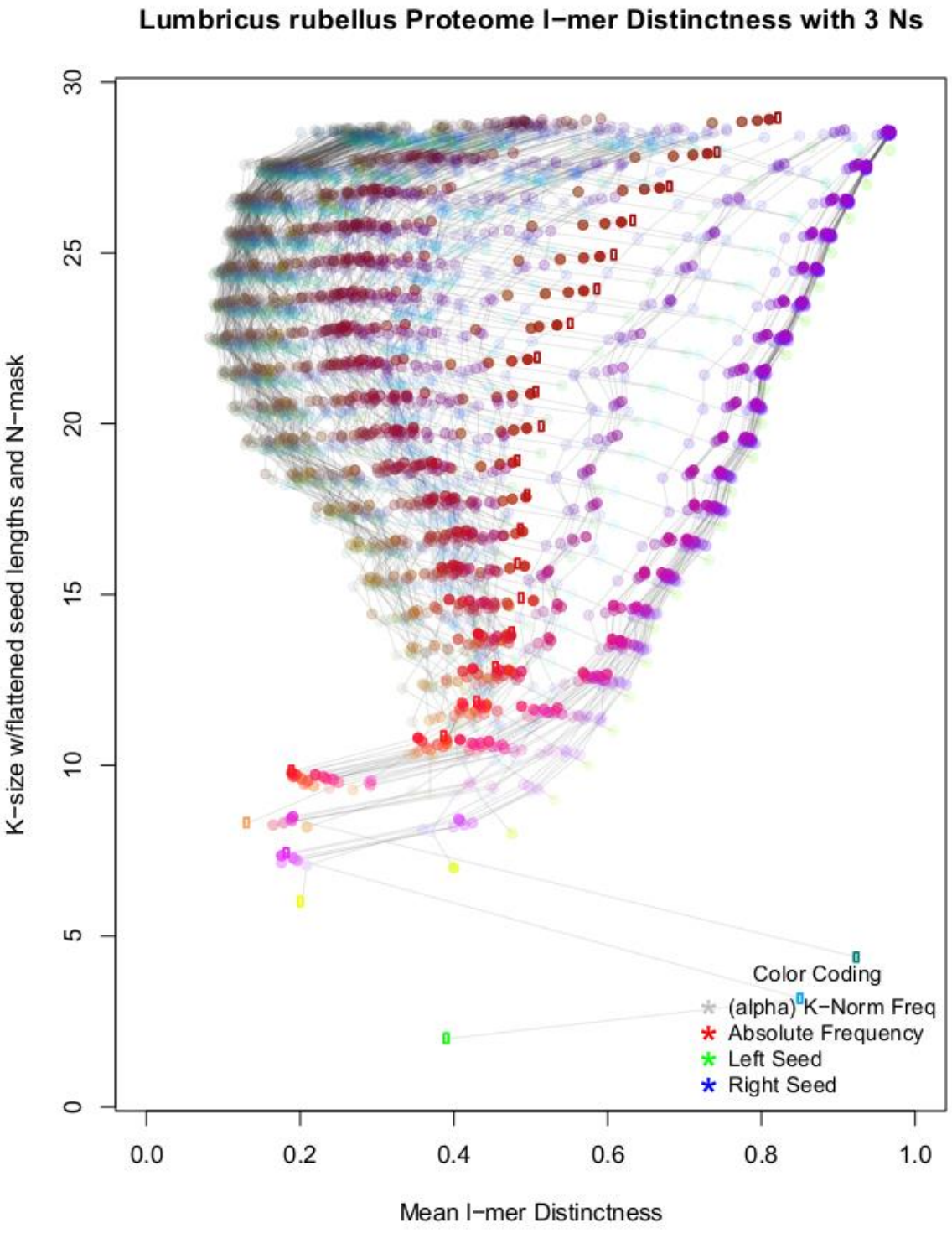

Figure 50. Lumbricus rubellus full proteome signature. N=3.

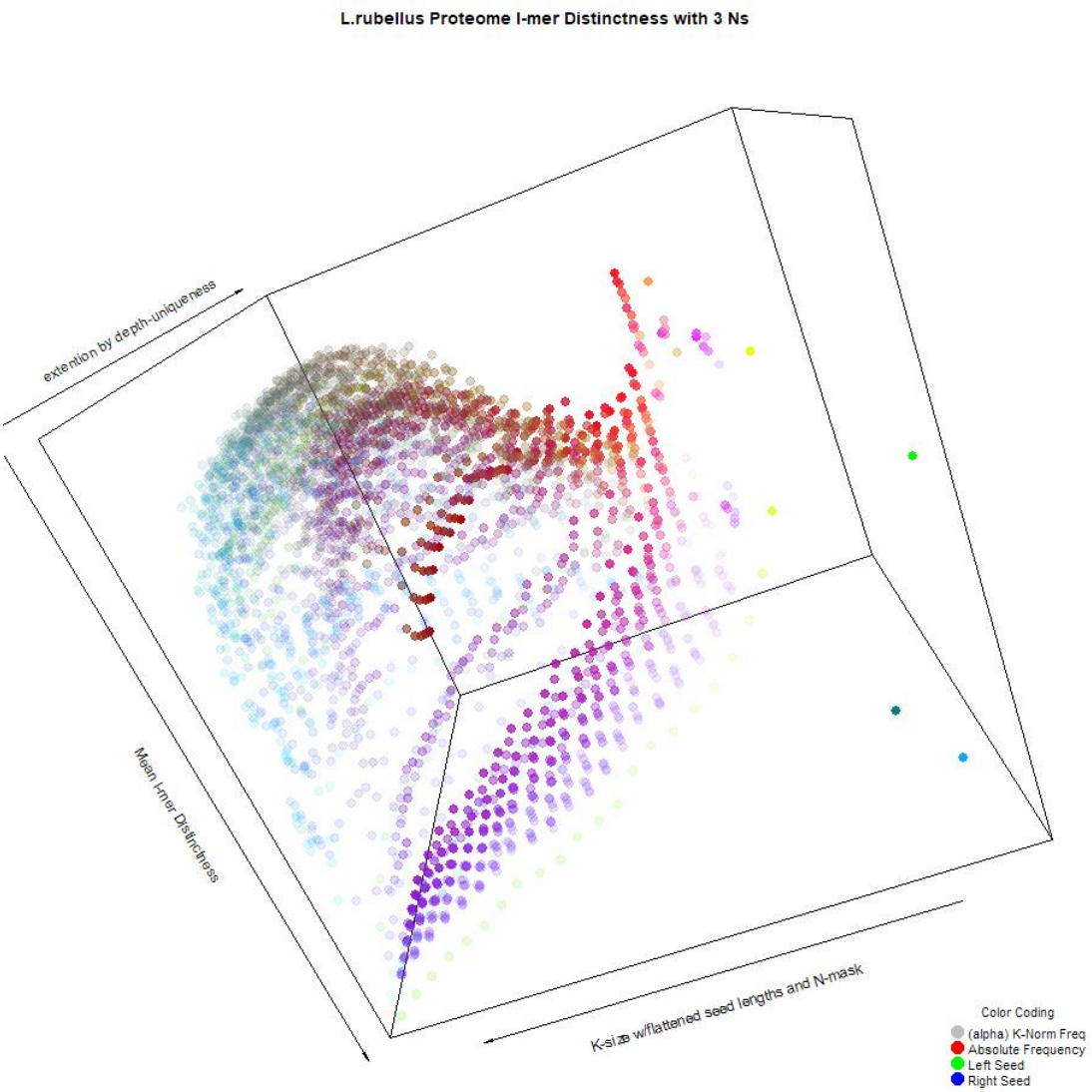

Figure 51. Lumbricus rubellus full proteome signature, alternative visualisation. N=3.

The 3D plots produced of these signatures are primarily illustrative of the depth of complexity discovered by this method. It is relatively difficult to compare between them due to the rotation-occlusion issue. Adding the thread lines, as in Figure 52, makes them particularly dense.

In summary, the proteome test set was able to demonstrate a wide variety of signatures, with key correlates between the singular tree summaries, and the patterns found in the signature graphs. The visualisation density issue is still a limiting factor on the user's interpretation, however there are also higher perspectives in the interpretation which don't always require the discernment of every single category's distinctness, or it's thread pattern.

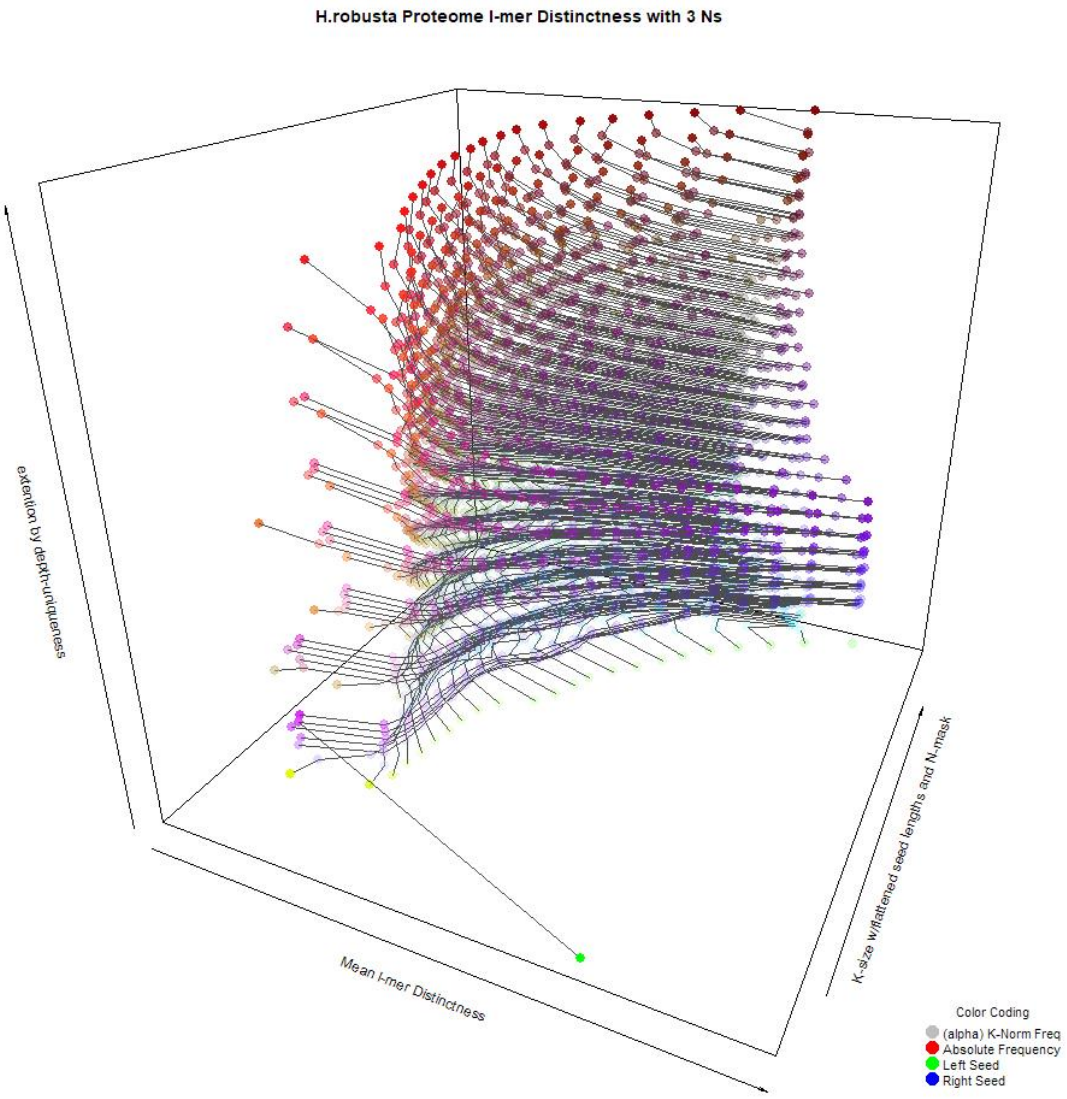

Figure 52. Helobdella robusta full proteome structure, alternative visualisation with threads. N=3.

3.4.5. Test Set: *E. coli* Genome Signatures

The second test set involves the signatures from the DNA of 18 *E. coli* genomes, retrieved from NCBI Genome database (NCBI 2016). The genomes were sampled from six of the seven major phylogroups as defined by the Clermont typing method (Clermont et al. 2013). Group C was only excluded due to data quality/availably issues. The phylogroup selection was applied with the intention of viewing the range of signatures across the broadest range of genomes available within the restriction of a single species. This serves as a counter-point to the previous test, which reached across hundreds of millions of years of evolutionary time. Here we investigate the variability of signatures within a tightly restricted set – to see if it might be informative, and to see the visual differences of with relatively small changes in input.

Table 6. E. coli, 18 genome structure summary scores.

E. coli Test

Strain	Phylogroup	Structure	Structure - Sub	Shape	Shape - Sub	Size (bp)	>K Frequencies (%)
S88	B2	0.1411	0.0378	1.290	3.453	5,166,121	1.85
LF82	B2	0.1400	0.0375	1.289	3.645	4,773,108	0.99
E2348/68	B2	0.1399	0.0369	1.289	3.635	5,069,678	2.67
SMS-3-5	F	0.1409	0.0373	1.286	3.692	5,215,377	1.53
IAI39	F	0.1405	0.0372	1.284	3.607	5,132,068	4.33
B093	F	0.1406	0.0367	1.280	3.804	5,205,351	0.99
TA280	D	0.1412	0.0371	1.284	3.721	5,296,938	1.32
H299	D	0.1417	0.0380	1.287	3.392	5,317,840	1.53
UMN026	D	0.1419	0.0380	1.287	3.491	5,202,090	1.73
ECOR31	E	0.1416	0.0371	1.273	3.661	5,443,045	3.20
B185	E	0.1415	0.0379	1.282	3.556	5,144,306	0.91
E101	E	0.1418	0.0375	1.283	3.499	5,181,904	1.53
_55989	B1	0.1403	0.0372	1.285	3.705	4,989,876	0.45
IAI1	B1	0.1399	0.0379	1.298	3.486	4,700,560	1.81
O111	B1	0.1424	0.0386	1.289	3.301	5,284,381	4.41
HS	A	0.1398	0.0380	1.299	3.434	4,643,538	2.13
ATCC-8739	A	0.1409	0.0389	1.297	3.272	4,746,218	1.99
TA007	A	0.1417	0.0378	1.286	3.364	5,299,319	2.36

Table 6, like Table 5, shows the range of structures and shapes across the test set. Perhaps as expected the structure and shape scores for all entries are highly consistent. More interestingly, the escaped frequency rate remains quite variable from 0.9-4.4%. The phylogroup categories did not have any significant correlation with any of the scores. This could be indicative of the substantial genomic variation present within phylogroups. Additionally, the signature and shape scores are intended as indicators of sequence set structure and complexity rather than evolutionary distance. Any distance between genomes describable by these scores could be thought of more as a biological architecture distance, which is only tangentially related to evolutionary time.

All the signatures generated by this test are available in Appendix 2.2 in the file E.coli_signatures, and additionally presented in series as a short .gif as ANIMATION1, in Appendix 2.4.

To demonstrate the effects of applying the local-null correction to the signature Figures 53 and 54 were created, pre- and post-correction. The main difference is the removal of most of the pre-saturation (~11.6) points. This shows that there was no over-abundance of shorter oligomers which wasn't also emergent in the random-shuffled input.

If there is a trend which Figures 54-57 follow, it is one of similarity to the DNA null-curve. The aspects of the small subset set of *E. coli* DNA which began to emerge as different to the null-curve are exaggerated in scale, but the transformation of signature shape is nowhere near as dramatic as in the proteomes. Additionally, whilst the structure scores in the Table 3 are significantly higher than Table 2, they are not directly comparable, as structure is always measured relative to sequence space occupation only equivalent alphabets may be compared numerically without additional transformation. The coherence to the null curve suggests that these DNA inputs were highly complex, and relatively low in highly duplicated structures. The corrected distribution shapes were also generally higher than most of the proteomes, except *Apis mellifera*, which also bore the most similarity to the small input subset, and to the peptide null curve more generally.

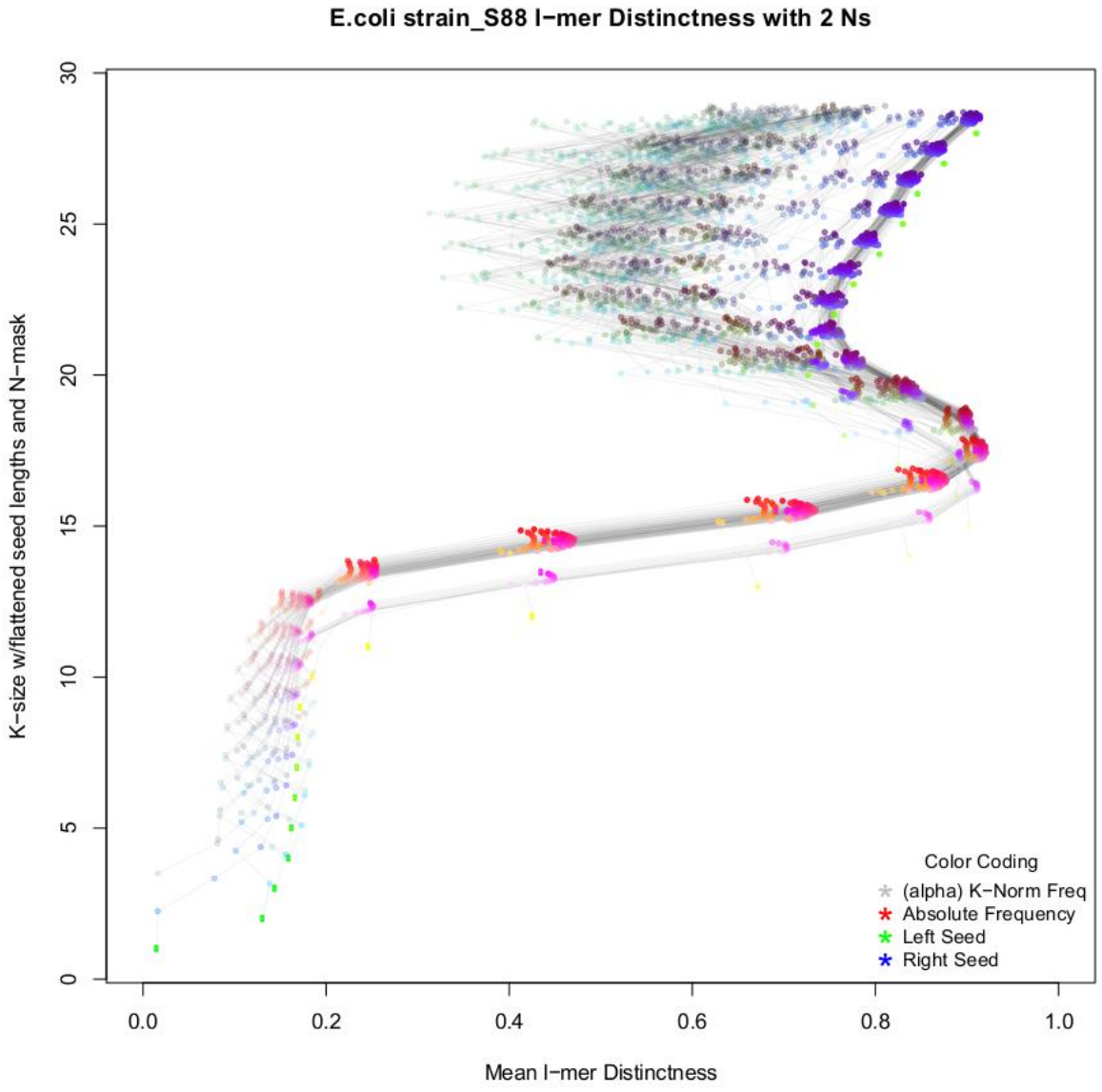

Figure 53. E. coli strain: S88 full genome signature. N=2. Without local-null subtraction.

This relatively low structural scale could be reasonably expected in bacterial genomes that usually have fairly small gene families, with many genes being unique single copies (Pushker et al. 2004). Still there are differences between the strains which may highlight their evolutionary behaviours. For example, despite being very closely related in structure scores, S88 and B185 (Figures 54 and 55) have quite a marked difference in the frequency densities of short left and right seed N-masks across the *l*=20-30 range, suggesting a pattern of motifs with multiple mutations separated by ~15+ bases are far more common in B185.

A similar comparison is possible between strains HS and O111 (Figures 56 and 57), the highest and least structured entries in the table respectively. However, in this case we can see the scale of the structures reflected also in the convexity of the curve of the rightmost band. Interestingly part of the

signature of higher overall structure manifests as less distinct primary sequence threads in these cases.

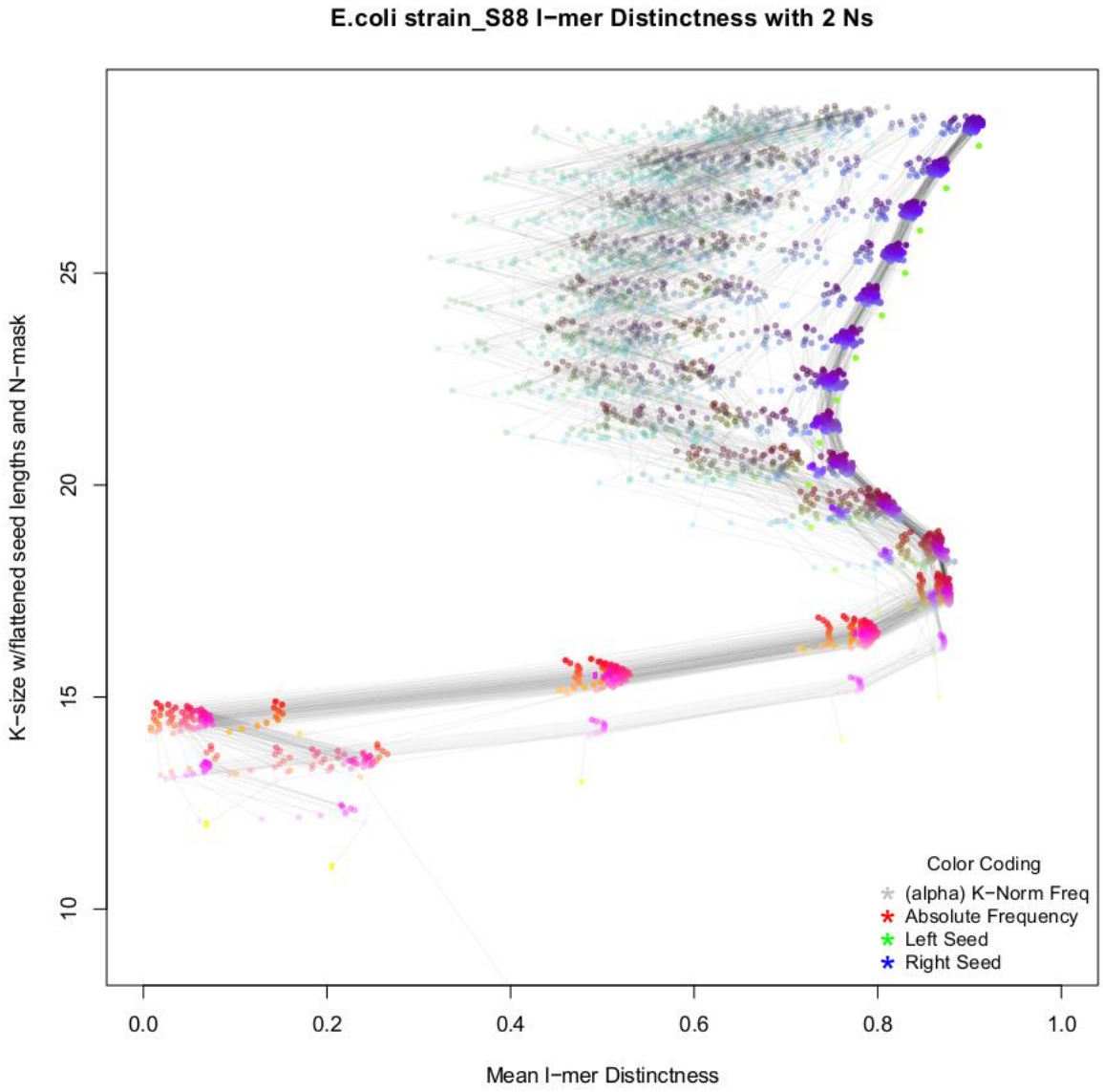

Figure 54. E. coli strain: S88 full genome signature. N=2.

The original conception of the *k*-mer tree signature method was to describe the structures in large and complex genomes, however given the current memory and performance limitations of the software, smaller bacterial genomes were chosen. When the signatures are expanded in 3D plots (see Figure 58), the patterns do not expand to a greater depth of complexity and remain very similar at different depths of the z-axis. Further performance gains must therefore be made before the DNA tree signatures can be suitably refined for the intended 0.5 - 1Gb genome inputs.

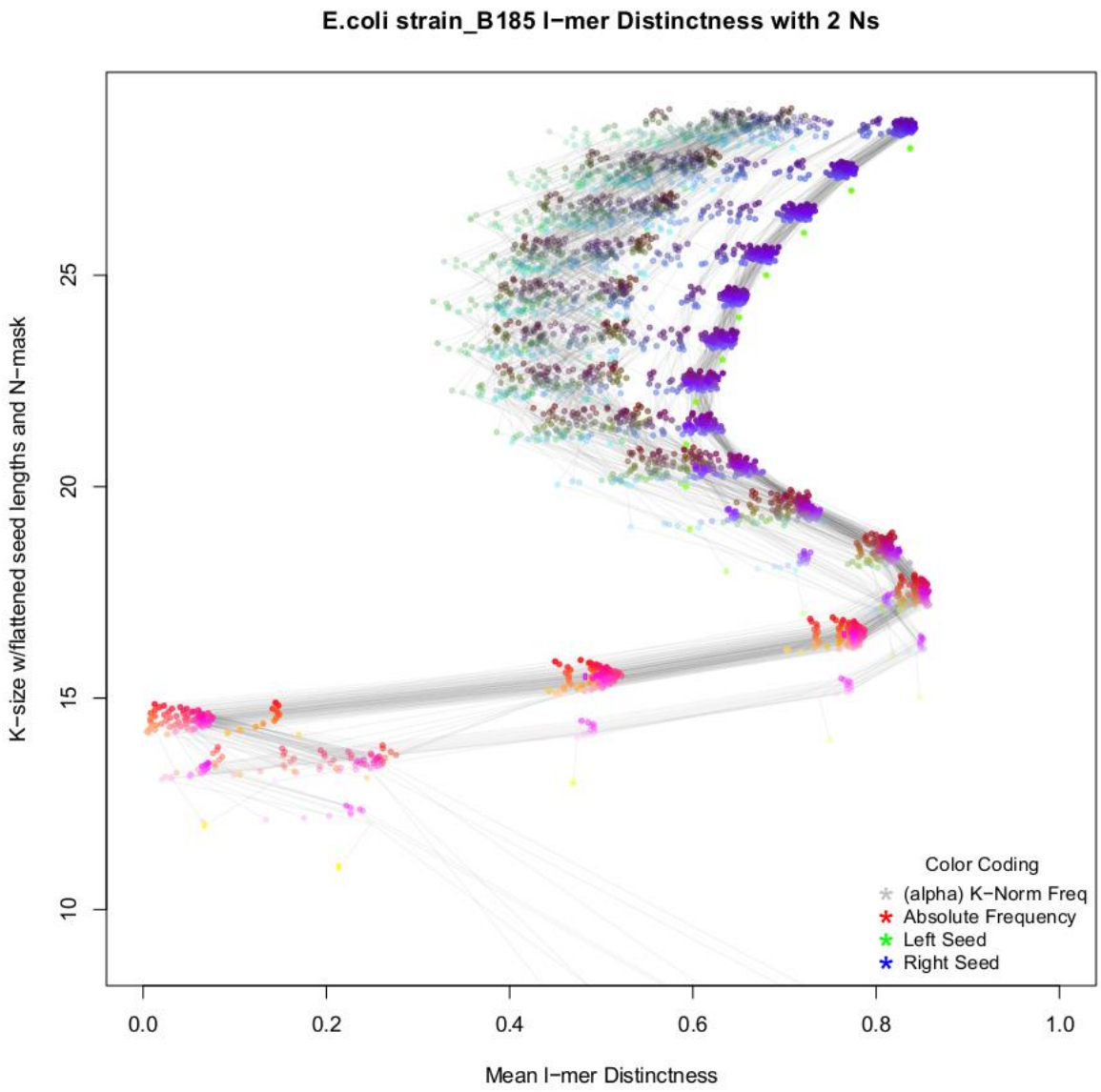

Figure 55. E. coli strain: B185 full genome signature. N=2.

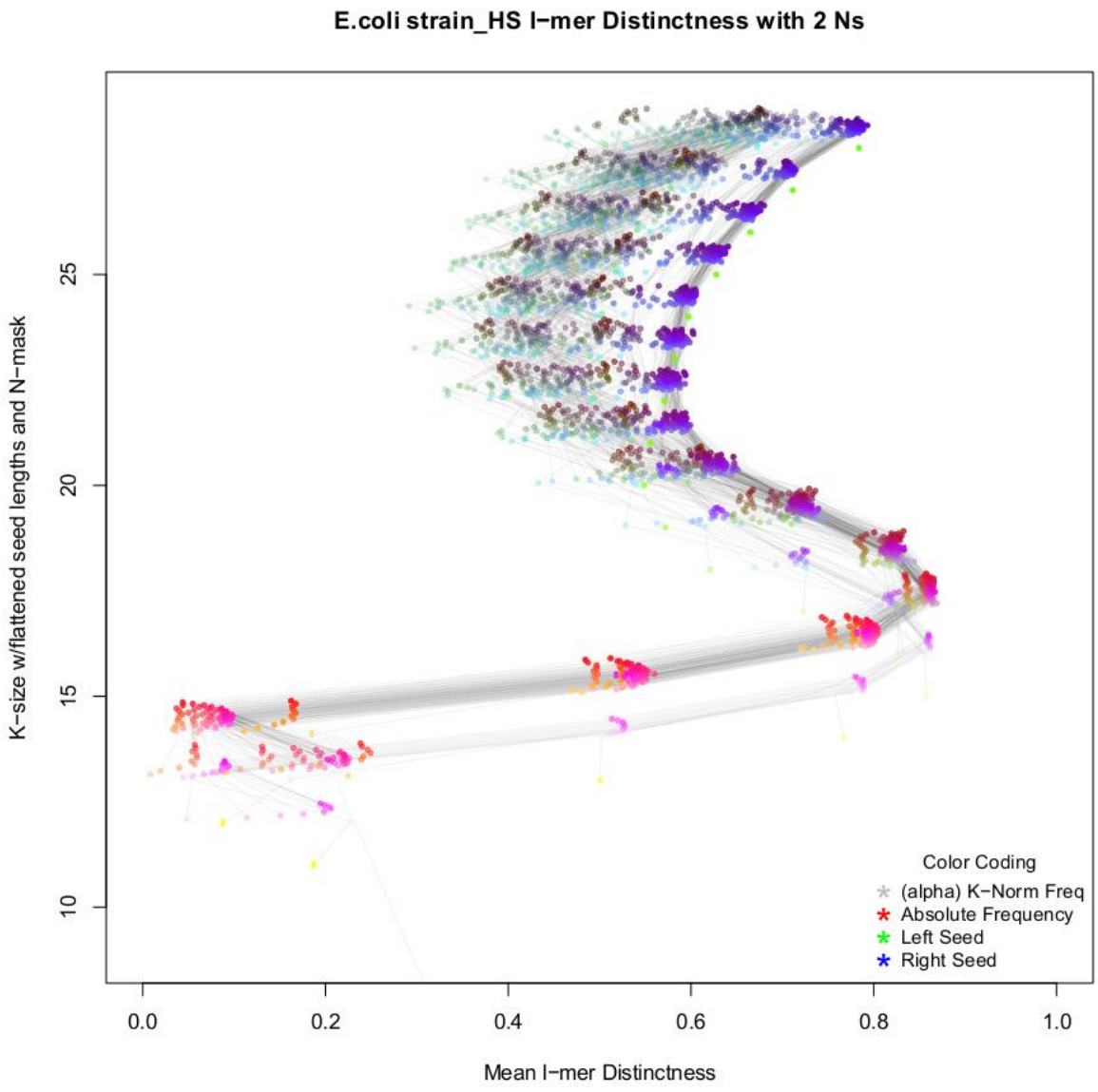

Figure 56. E. coli strain: HS full genome signature. N=2.

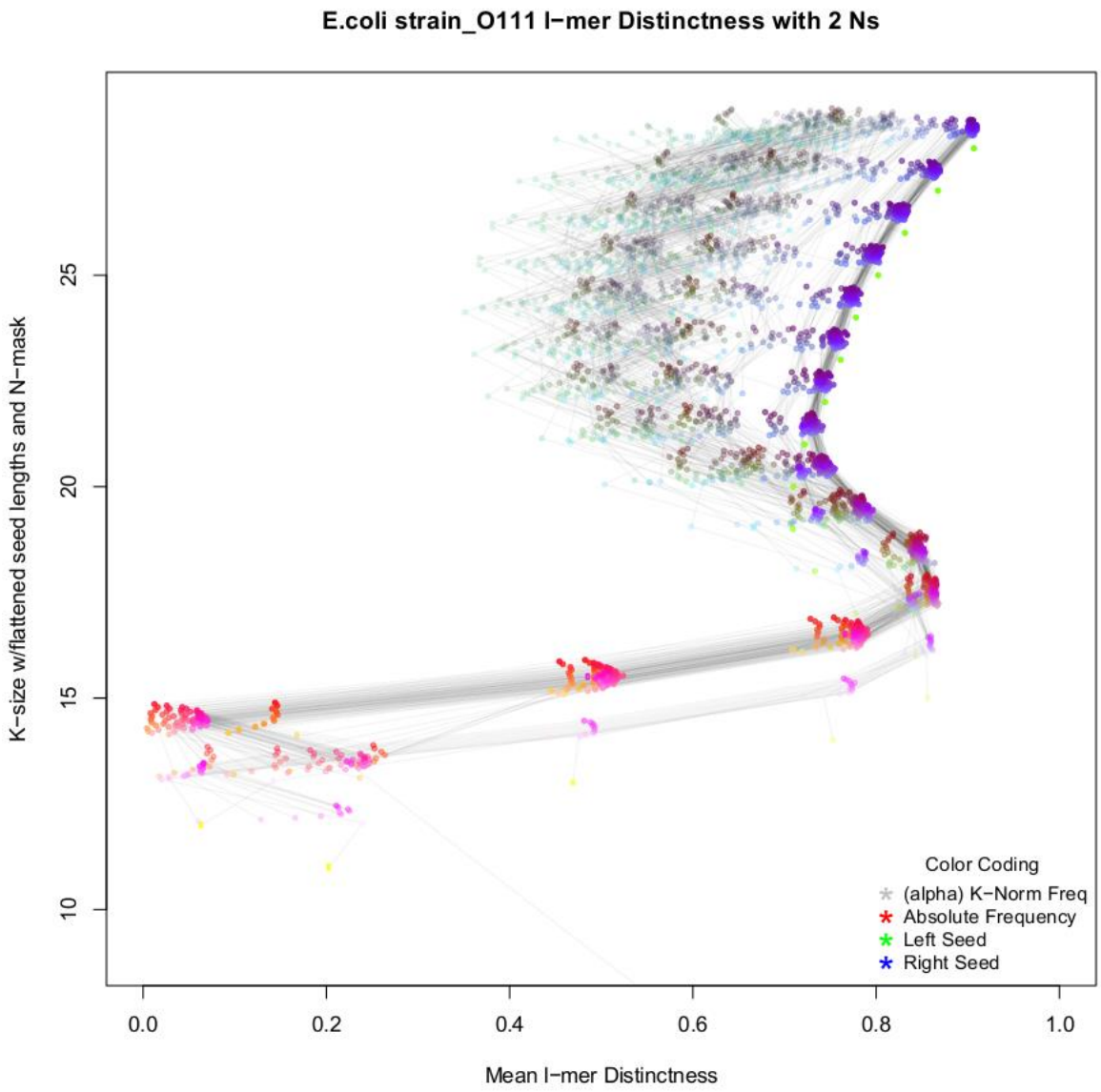

Figure 57. E. coli strain: O111 full genome signature. N=2.

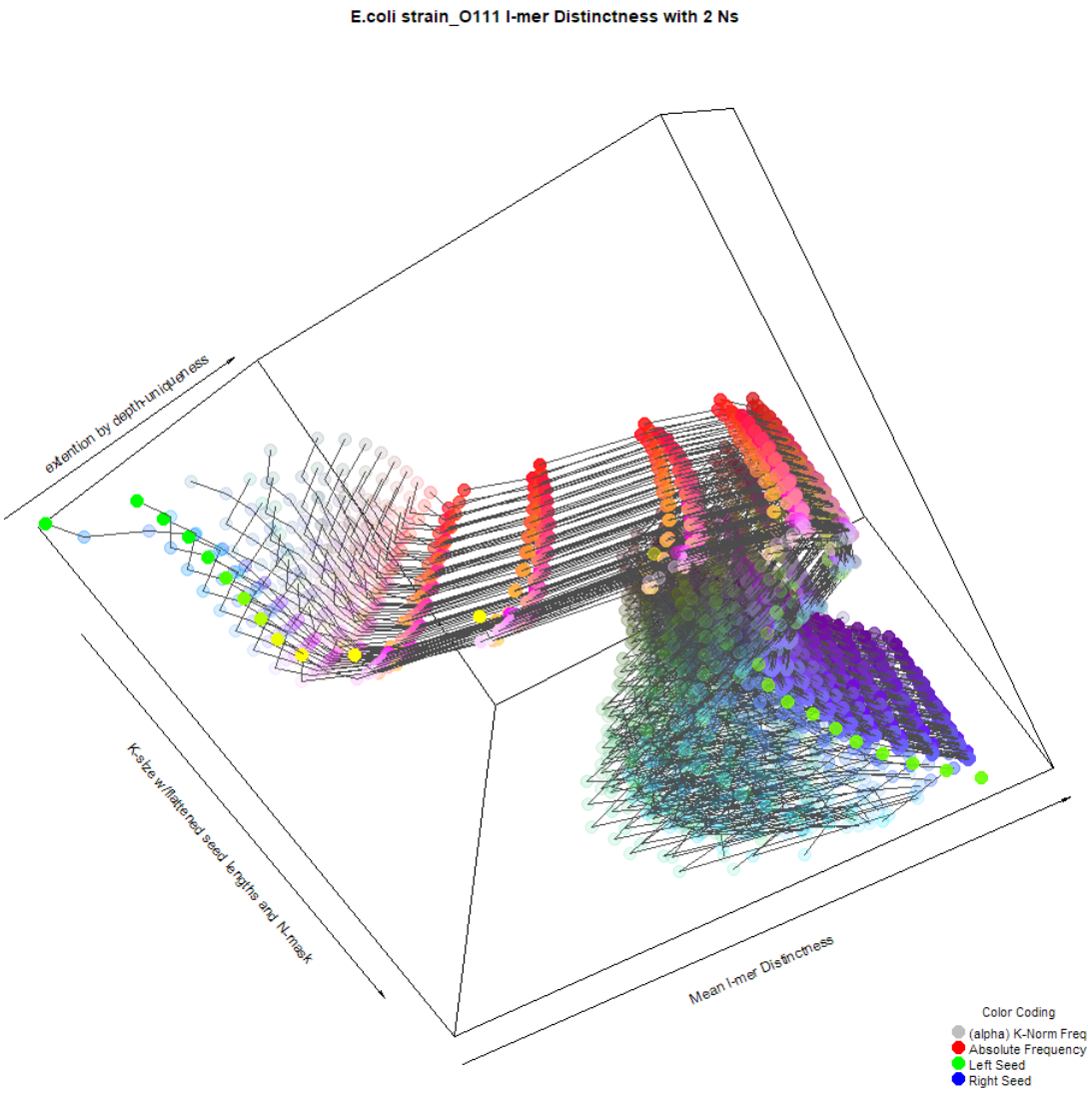

Figure 58. E. coli strain: O111 full genome signature. N=2. 3D visualisation.

3.4.6. Test Set: Protein Families

The third test set was intended to test the program's ability to describe smaller, yet highly structured datasets. Protein families were expected to satisfy this criteria due to the anticipated number of repeated domains in the input set. The input sets were retrieved from PFAM (Lee et al. 2015) ftp server. Due to the extreme size range in the protein family reference sets (100K – 5M), some files were limited to the top 10K lines. To maximise the utility of this test, the protein families selected were identical to the six highly allelically divergent environmentally adaptive families identified in both *Lingula anatina* and *Lumbricus rubellus* in Data Chapter 1. Of interest is Data Chapter 1, Figure 21, which shows the variable rates and distributions of allelic divergence amongst them. The hypothesis being that the rates of evolutionary divergence between alleles may have some correlate in the signatures.

Mucin-like glycoproteins were identified as being the more divergent group, followed by ZIP metal transporters. Interestingly, it appears that in Table 7 these two also the highest structure scores, with mucins coming out as the most by far the most structured. The two least divergent families were Glucuronosyltransferase and GPCR Chemoreceptors, and again the extremes align in reverse, with GPCRs achieving the lowest structure. Although this is not a statistically valid proof of allelic divergence and structure correlation more broadly, it does appear to have some interesting intersection in this case.

Table 7. Protein Families Structure Summaries

Protein Family	Structure	Struct. - Sub	Shape	Shape - Sub	Size (aa)	>K Freq (%)
Epithelial Sodium Channels	0.0461	0.0393	1.765	1.858	423,109	7.14
Glucuronosyl-transferase	0.0719	0.0636	1.606	1.754	384,828	2.12
GPCR Chemoreceptors	0.0504	0.0391	1.625	1.888	301,048	6.20
Laminins	0.0494	0.0414	1.630	1.922	259,950	3.01
Mucin-like Glycoprotein	0.1914	0.1774	1.561	1.690	119,725	11.33
ZIP Metal Transporters	0.0886	0.0751	1.625	1.776	380,509	6.38

This test set will also offer the chance to demonstrate the first derived measurement type, the WSD of category distributions (see 3.2.5.). Weighted deviations are more descriptive in the case that a specific set of structures are in question, as the can reveal the extent to which a category represents a singular feature of the protein family. Here the inverse scale of the WSD has been coded to the size

of the points used to show each category. The WSD scale has also been normalised to the maximum per depth, this ought to help combat the effect of WSD always shrinking towards distinctness boundaries, however this effect does persist. To be clear, the smaller the WSD, the narrower the distribution, the larger the point will be drawn on the plots in Figures 59-66, on a linear scale.

The information which can be gleaned simply from the WSD component of the signature is demonstrated by several comparisons within these images. Firstly, looking at the differences between epithelial sodium channels and Glucuronosyltransferase (Figures 59 and 60), there is only particular WSD pattern which stands out. This is the ~0.3 distinctness 'backbone' band between depths 9 and 16 found in Figure 60. Whilst both plots show a typical pattern of high deviation throughout the middle of the plot, suggesting most component signatures found in this range are a diverse structural mix, the 0.3 band feature in Figure 60 suggests a specific consistency to the dispersal patterns within that range, perhaps indicative of a conserved active domain, or conserved motifs within them. Figure 59 by contrast has far more 'distinctness outliers' generated by low-frequency N-masks with relatively large left and right seeds – these being depth-specific points which to not cohere to banding patterns. This suggests the presence of rarer variants present within motifs already typified by more regular variation patterns; smaller groups of domains which break away from the main set in an unusual manner. Given the breadth of the Epithelial Sodium Channel family, and the variety of sub-groups within it, this could be expected (Hanukoglu & Hanukoglu 2016).

All the protein family signatures generated by this test are also available as 3D visualisations in Appendix 2.3.

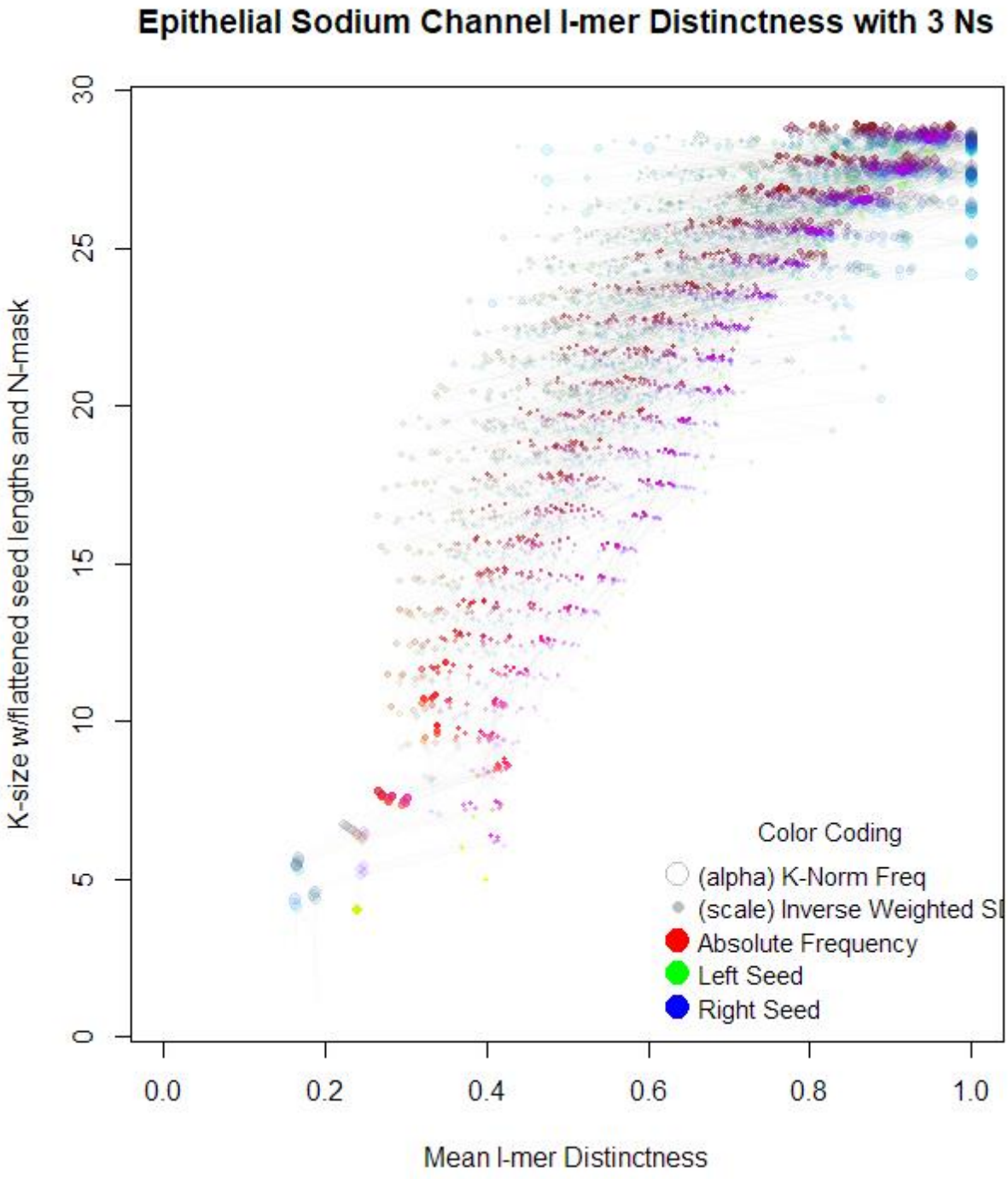

Figure 59. Epithelial Sodium Channel, (PFAM) Protein Family, Signature with WSD. N=3.

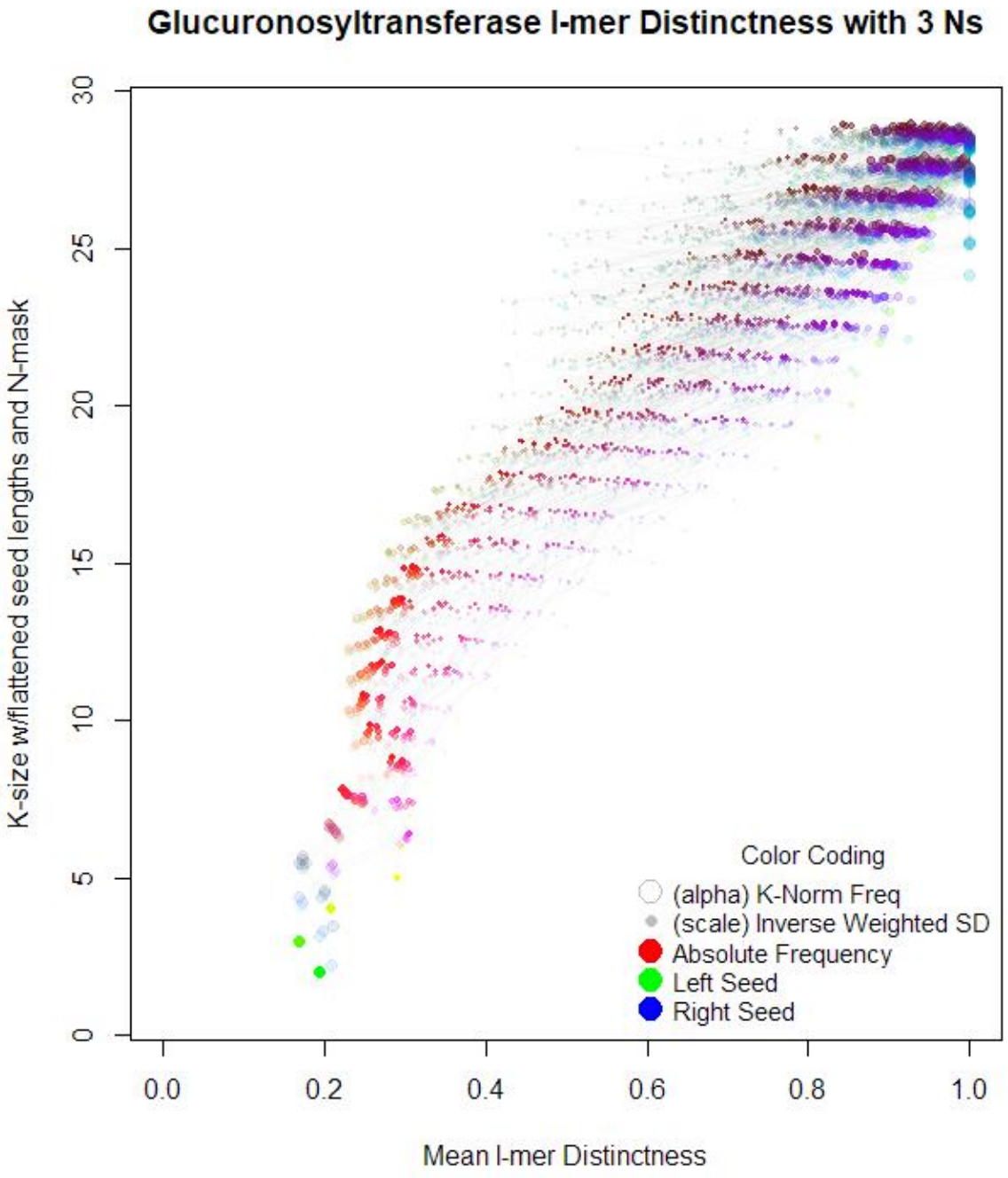

Figure 60. Glucuronosyltransferase, (PFAM) Protein Family, 3D Signature with WSD. N=3.

Comparison between Figures 60 and 61 may also be illustrative of the WSD signatures. It seems that the GPCR Chemoreceptor family, although considerably less structured than Glucuronosyltransferase in Table 7, has far narrower categorical distributions of dispersal type across the entire signature, although as in Figure 59, there are also many distinctness-outliers present. This may also be a commonality of membrane bound proteins with active sites, although it suggests GPCRs as a family have more homogeneity in their predominant variant patterning. A final point of comparison between them is the shape and distinctness position of the lower half of the signature. Although they both trend similarly, Figure 60 shows a more concave shape, whilst 61 is more convex. From this we can also infer than the flexible AA positions in GCPR Chemoreceptors may also be functionally more restricted to a certain set of replacements. Given that GCPRs are known to possess seven membrane spanning a-helices (Hollenstein et al. 2014), this could be a signature of the importance of hydrophilic/lipophilic AA restriction at regular helical sequence positions. Given that the slight convexity is also present in Figure 59, also representing a protein with membrane-spanning domains, this could be a more general signature of that attribute.

Returning to Data Chapter 1 Figure 21, and the mystery of the hyper divergent Mucins, we can now compare its signature (Figure 63) to the rest of the set. Remarkably, it is incredibly different. In addition to having a less structured tree summary, it also presents a signature far closer to the small subset curve than any of the other family signatures. Additionally, like Figure 60, although to a much greater degree there is a very low deviation dispersal pattern for very low left/right seeds along the leftmost band, reaching all the way up to depth ~25. This suggests a very large and consistently heterogenous variation pattern for most of the sequence content in these proteins. It has been observed that typically only the terminal domains in mucin-like proteins are conserved between species, whilst the central, threonine rich region, is made up of many tandem repeats whose primarily function appears to become highly glycosylated, thus creating the hydrophilic properties required to form gels or mucus (Acosta-Serrano et al. 2001). It seems that the signature of this large highly variable central domain is dominating the protein family signature and is responsible for the huge sequence variation seen in Data Chapter 1. We can also suggest that the propensity towards many repeats within the protein is the main constituent factor in the higher tree structure scores.

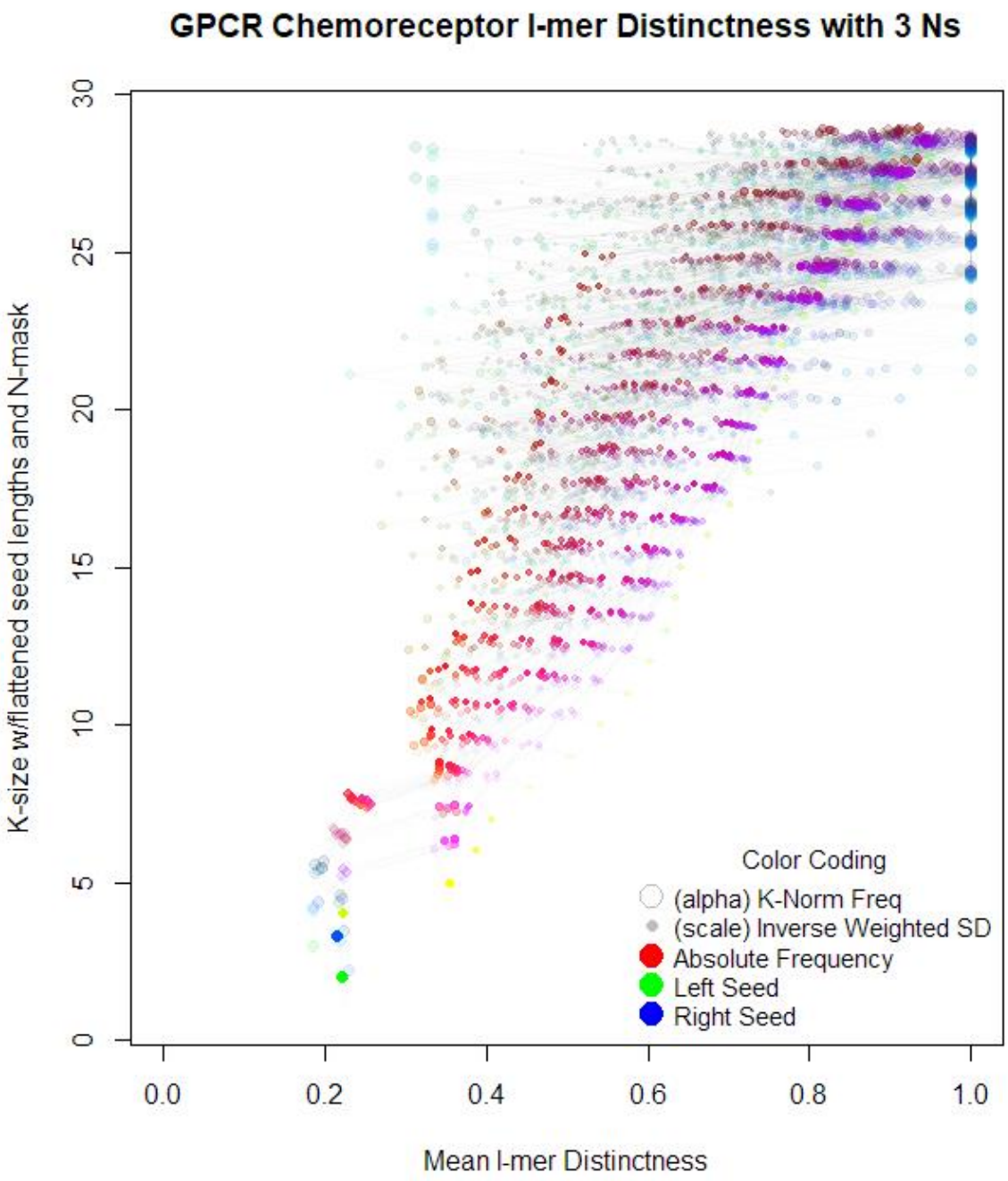

Figure 61. GPCR Chemoreceptors, (PFAM) Protein Family, 3D Signature with WSD. N=3.

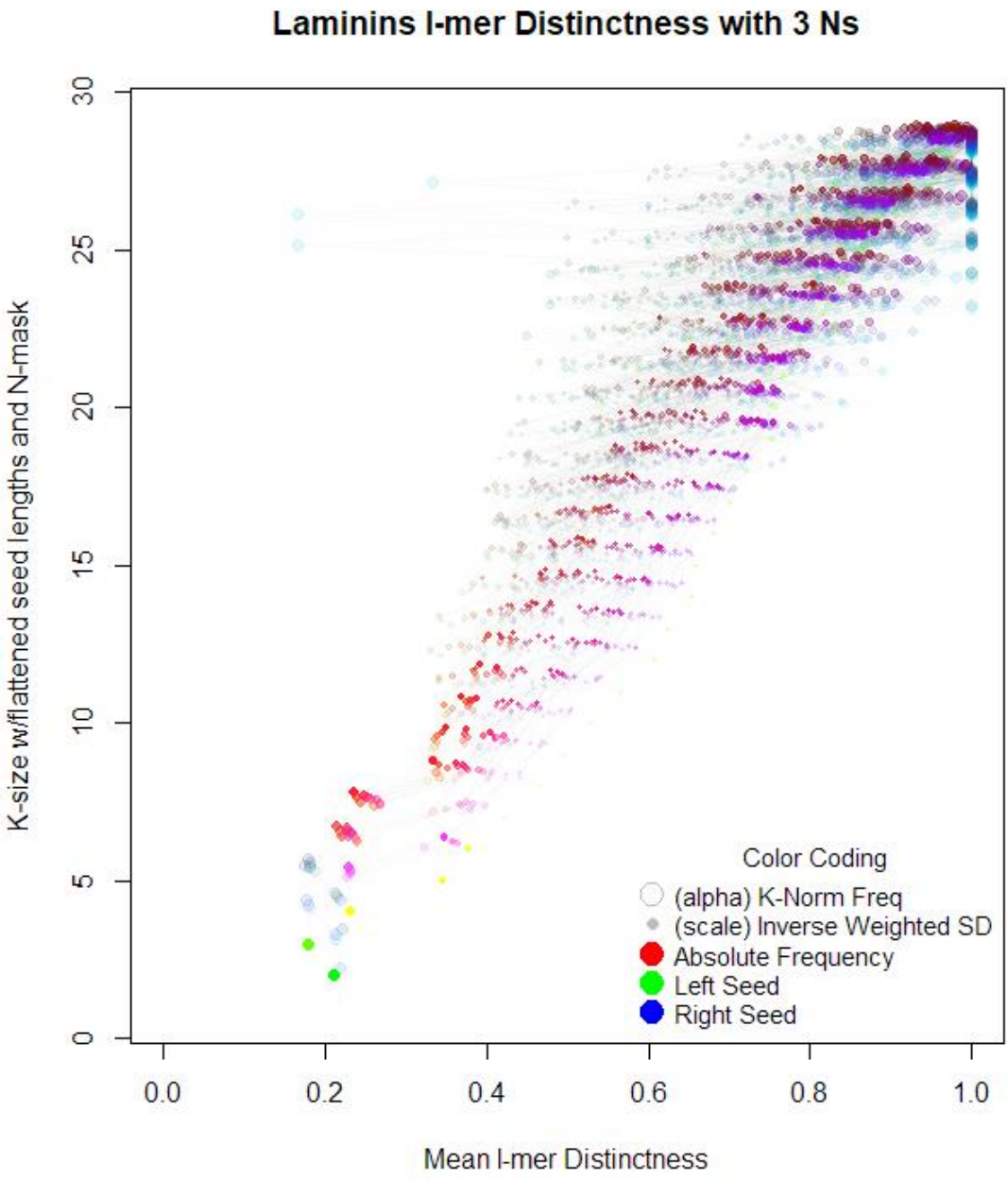

Figure 62. Laminins, (PFAM) Protein Family, 3D Signature with WSD. N=3.

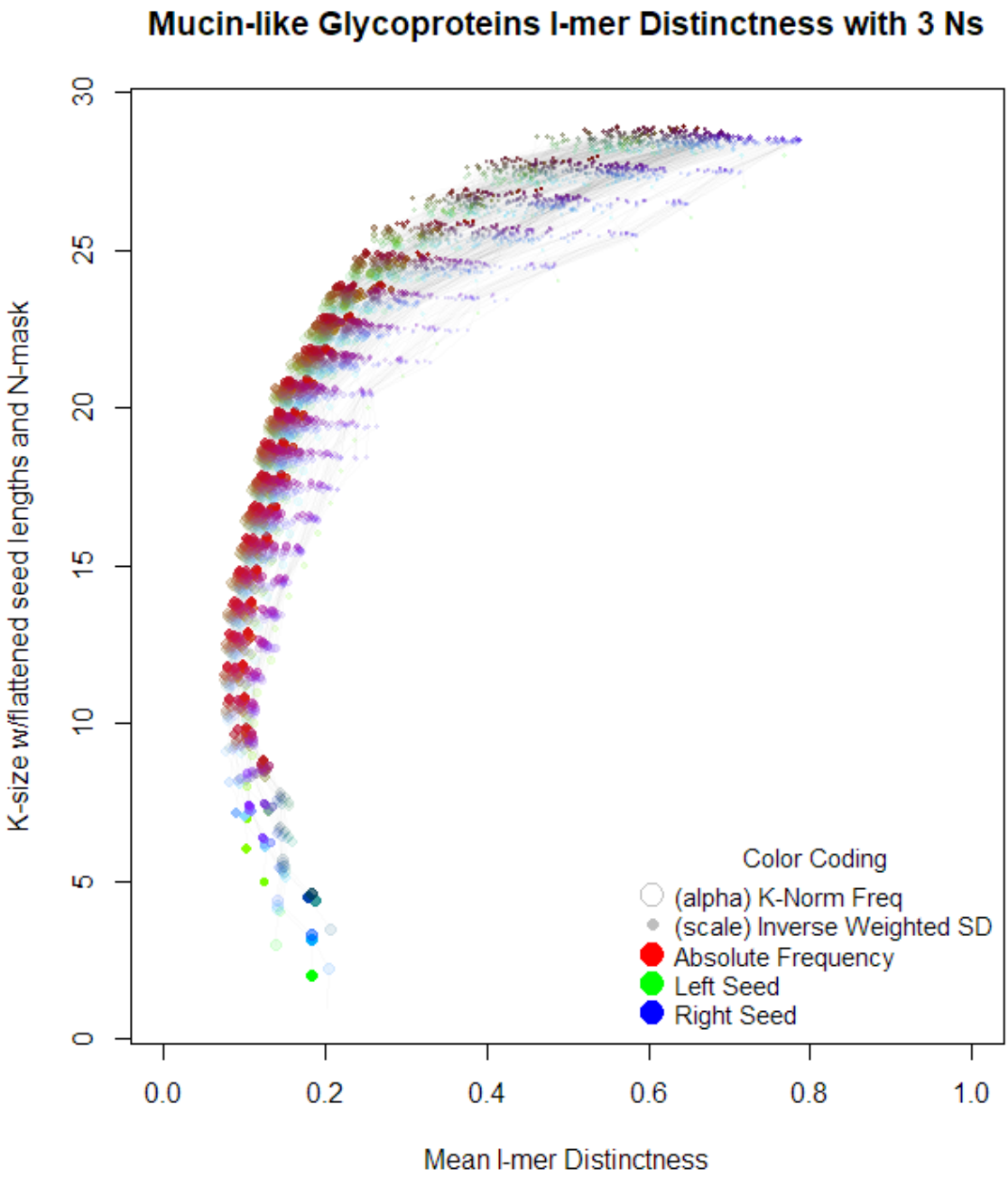

Figure 63. Mucin-like Glycoproteins, (PFAM) Protein Family, 3D Signature with WSD. N=3.

The Mucin-like family, with its high structure summary and low distinctness signature curve, shows us that both measurements must be read together to form a fuller understanding of the *k*-mer tree. From the signature and the summary, we can read that this is an example of abundant highly repetitive, yet highly flexible domains.

As the second most structured entry in Table 7 (although by a considerable margin), ZIP metal transporters. The signatures here show a slight backbone effect, and a concave low to middle distinctness curve, with broad motif distinctness distributions for almost all categories. These

transporter types have been shown to possess eight well conserved transmembrane domains with extra- and intra-cellular loops by contrast being highly divergent (Grotz et al. 1998)(Guerinot 2000). That there is a combination of two variant modalities could be the origin of the high deviation category distributions. The large number of conserved domains increasing overall structure, with the large number of highly variable loops reducing distinctness.

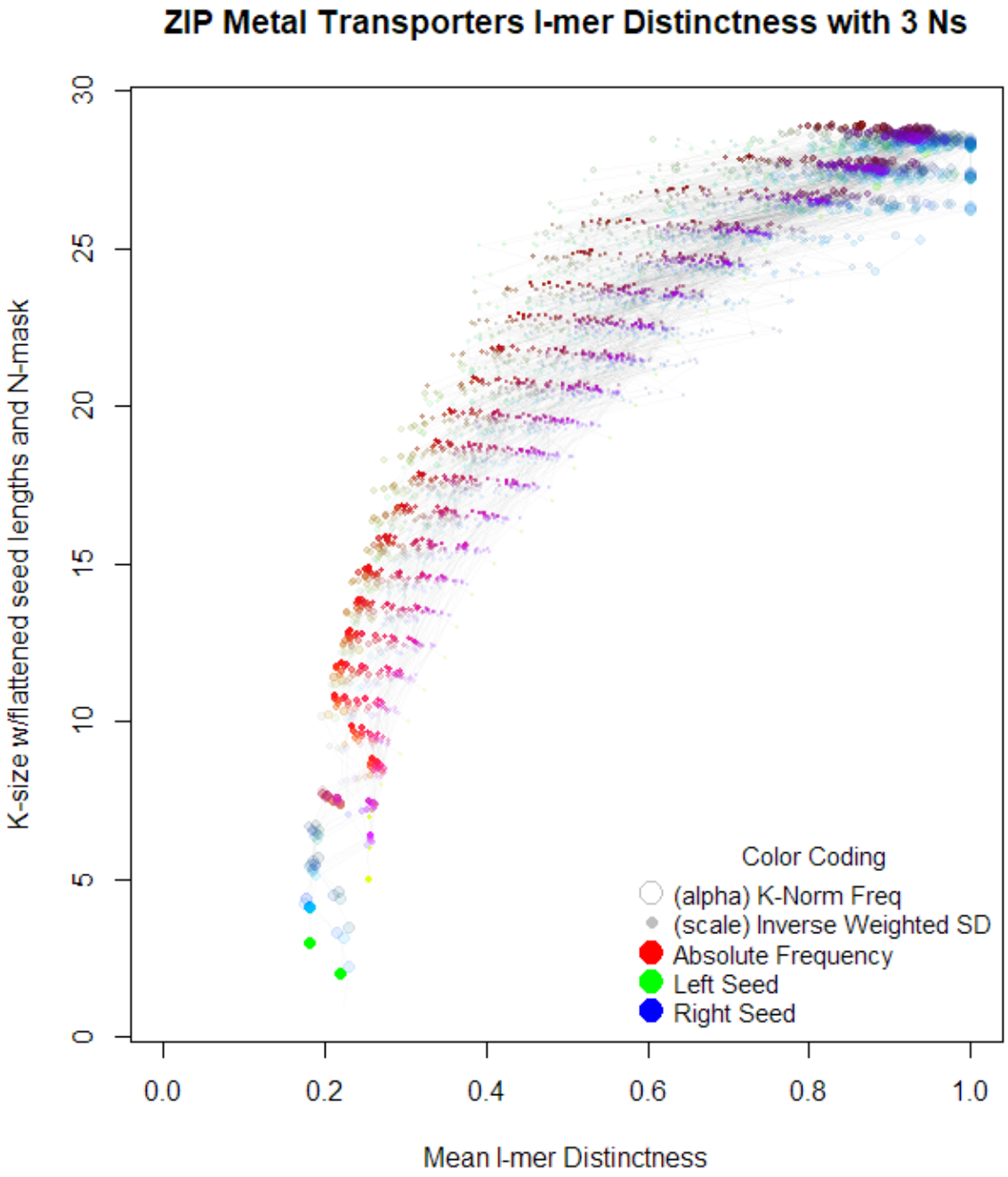

Figure 64. ZIP Metal Transporters, (PFAM) Protein Family, 3D Signature with WSD.

In summary of the family test set. The results combined with those in Chapter 1, Figure 21, suggest that protein families which have less duplicative structures may also be more resistant or sensitive to allelic variation, this could be possibly be described in the high-distinctness, low structure, high complexity category of sequence, and possibly high-fragility. The more duplicative and many-domain proteins typically are more structured in terms of sequence repetition, but with very indistinct variant patterning, and perhaps low overall complexity. These are of course early estimations of the possible set of relationships between signatures and family types. A full study of 1,000+ PFAM families via this interpretive method would be required to yield a stable reference set of guidelines for biological signature interpretation beyond the observations of pure entropic sequence descriptions.

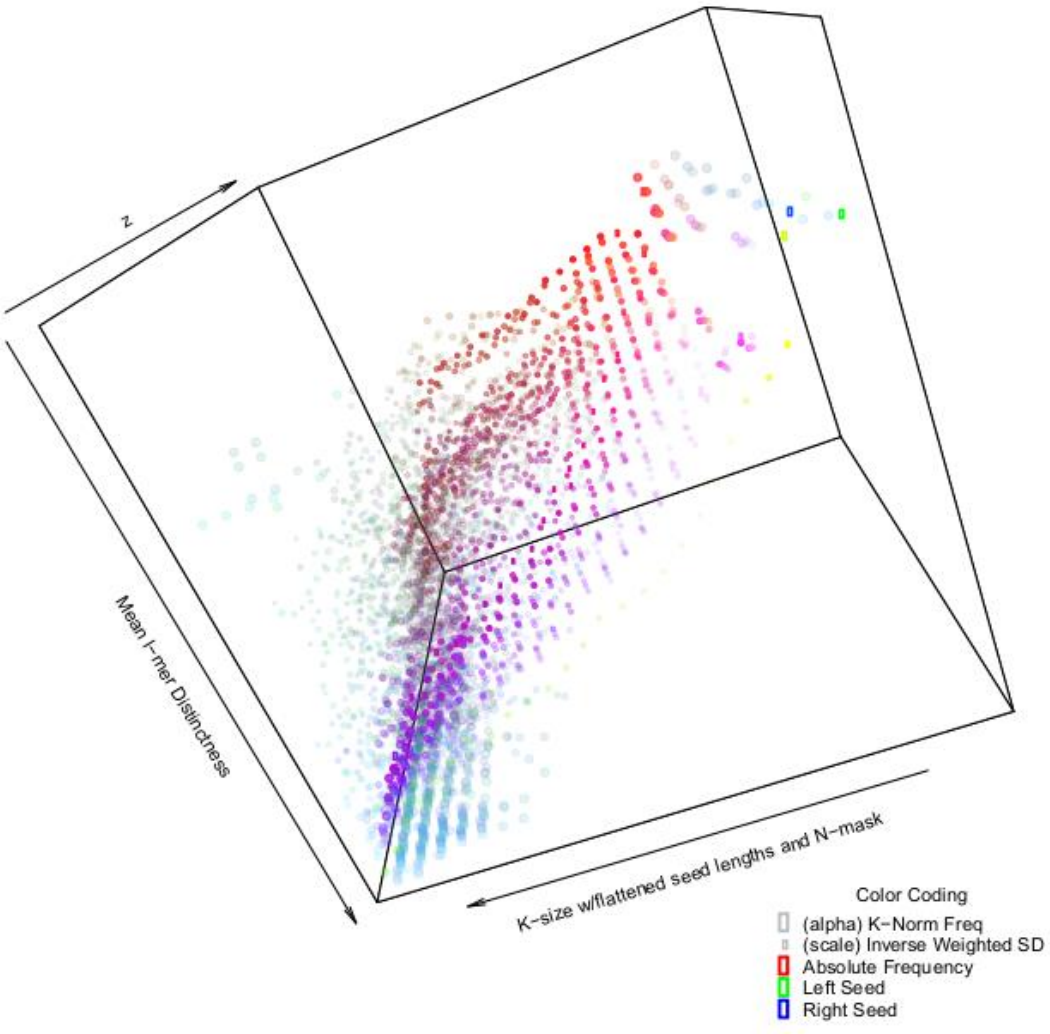

Figure 65. GPCR Chemoreceptors, (PFAM) Protein Family, 3D Signature with WSD, 3D Plot.

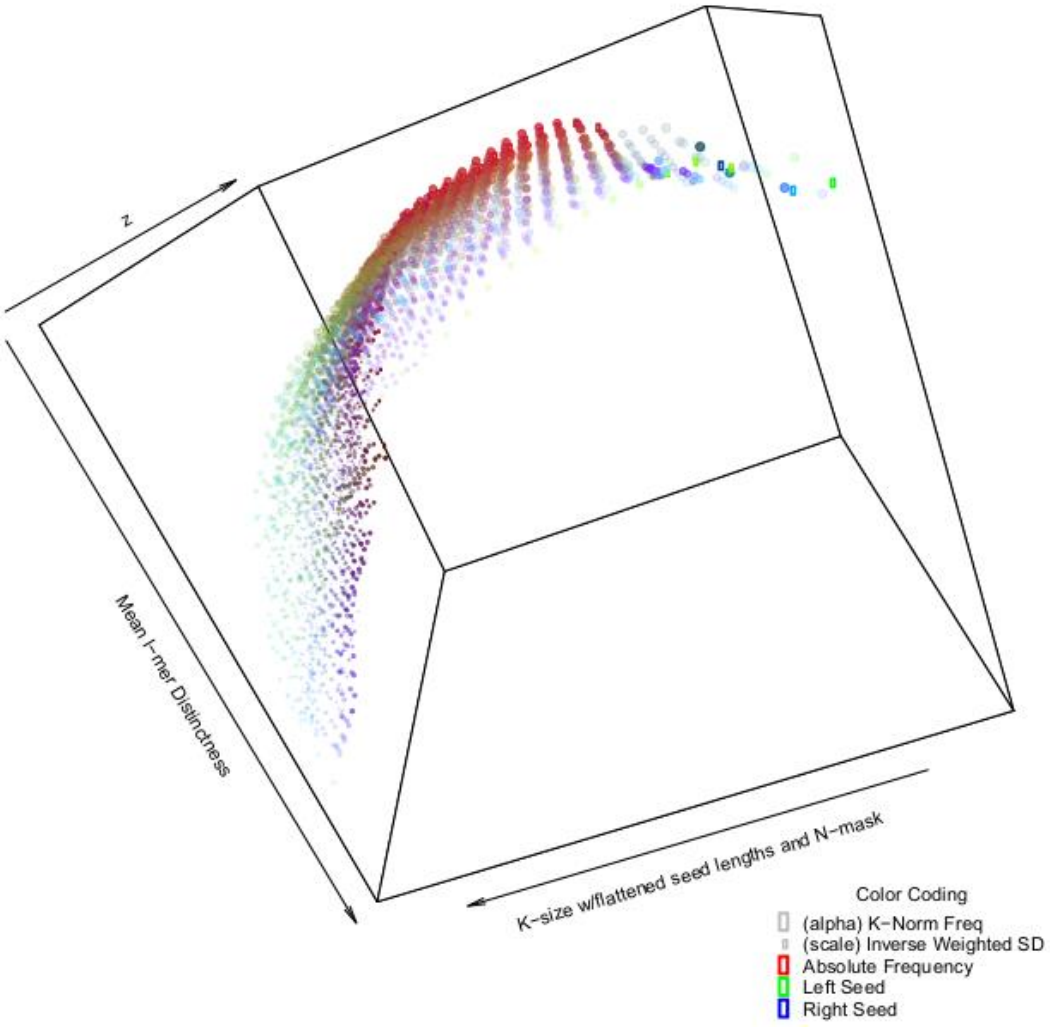

Figure 66. Mucin-like Glycoproteins, (PFAM) Protein Family, 3D Signature with WSD, 3D Plot.

Three-dimensional plots of GPCR chemoreceptors and Mucin-like glycoproteins have both been produced (Figures 65 and 66). These can further indicate visually the inner complexity of the low-structure GPCR family of sequences, and the contrast this presents to the low-complexity, yet highly structured sequence of Mucin domains.

3.4.7. The Pervasiveness of the Power-Law

The shape parameters generated for the uncorrected datasets are remarkably consistent. Figures 67-69 show that between the heterogenous data in the three test sets, the post saturation shape always ends up within the 1.5-2.0 range, with ~1.65 being the most common shape score. The distribution shape is discovered at all post saturation *l* values, and at all counts of *N.* There does not appear to be a correlative relationship (save for the saturation effect) between *l* and *a*, nor does there between *N* and *a*, nor does there between genomic and proteomic sequences.

For reference Figure 70 illustrates the steepness of the Pareto curve for the value of 1.6, in comparison to other values registered by some of the null-corrected summaries. The pervasiveness of this value of *a*, may also suggest to us that perhaps correcting shape parameters by local-nulls could be inadvisable, as it appears that the true biological power-law shape is manifest universally. Instead it could be suggested that the smaller variations within this 1.5-2 range ought to be considered with greater weight. In Table 7, the pre-corrected difference between GPCRs and Mucin-likes is between shapes of 1.625 and 1.561. Likely showing that the higher abundance of the most frequent motifs in Mucins due to the repeat-structured central domain creates the steeper Pareto distribution.

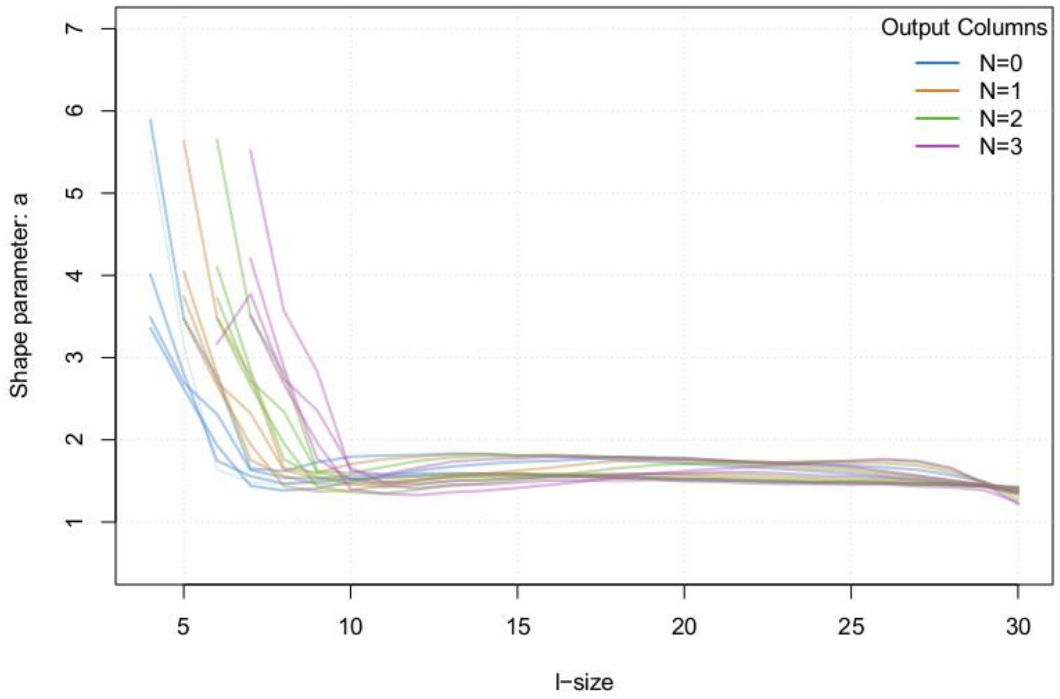

Figure 67. Pareto shape parameters distributed across l and N, Five Proteome Test Set (one line per proteome per N value).

The consistency observed here suggests that further work to establish the impact and value of the range of variation observed in uncorrected shape scores could be useful in maximising the information utility of the signatures.

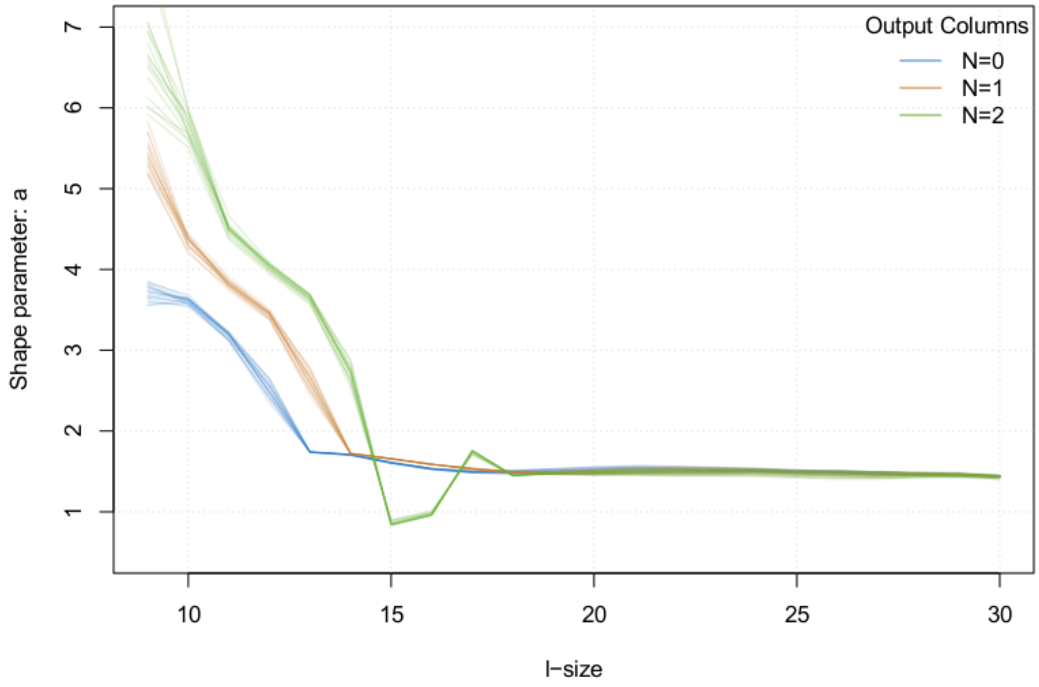

Figure 68. Pareto shape parameters distributed across l and N, E. coli Genome Test Set (one line per genome per N value).

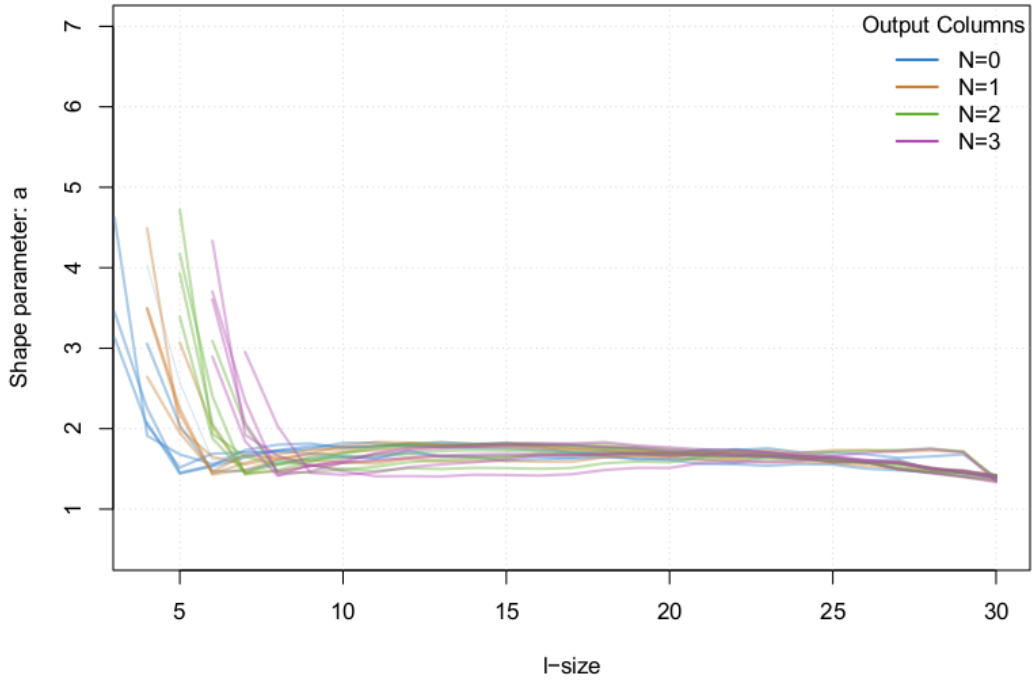

Figure 69. Pareto shape parameters distributed across l and N, Protein Family Test Set, (one line per protein family per N value)

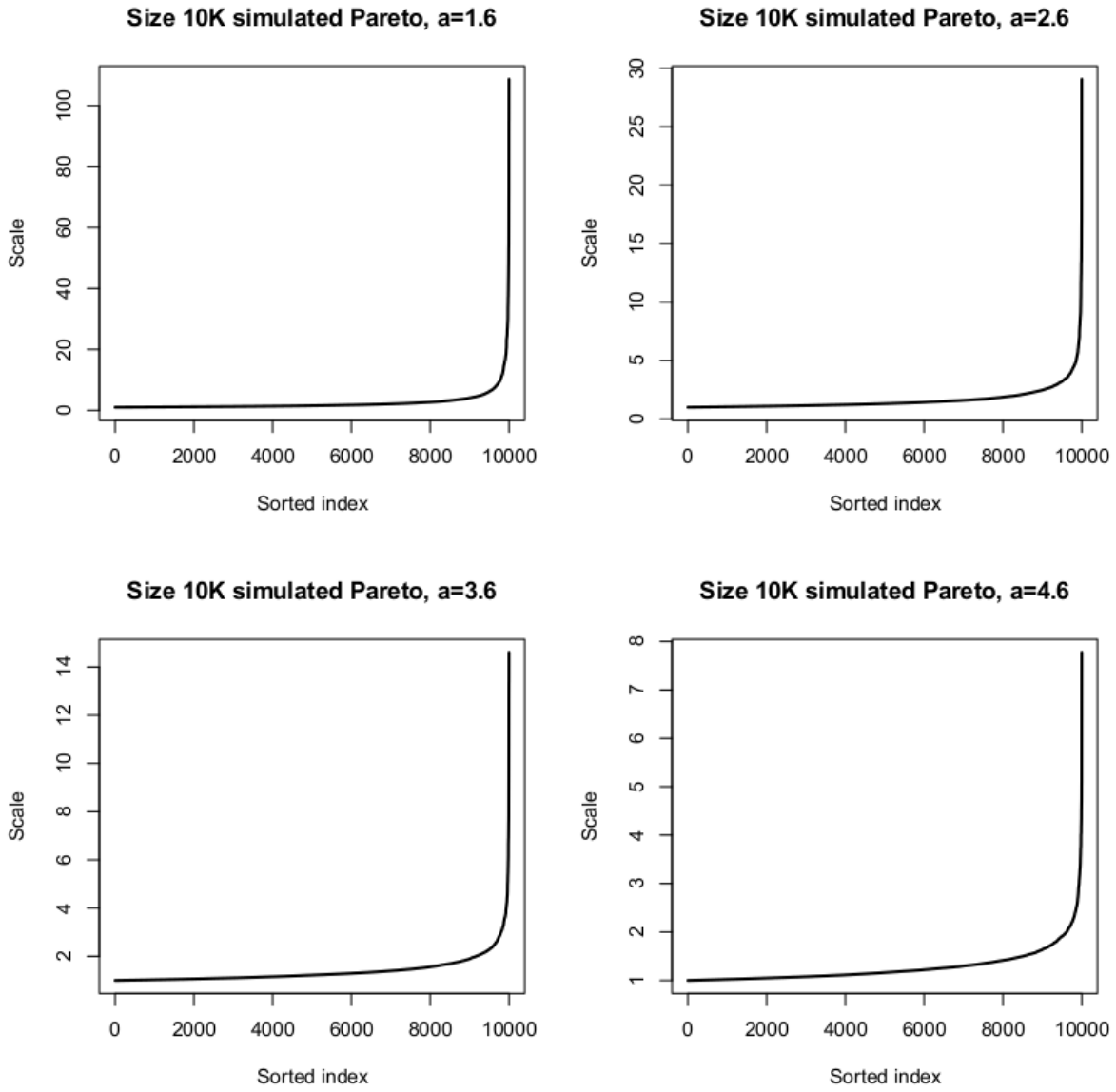

Figure 70. Example Pareto Distributions across shape parameters. Top-left: a=1.6, Top-right: a=2.6, Bottom-left: a=3.6, Bottom-right: a=4.6.

3.5. Discussion

3.5.1. Signature Performance

Having developed these aggregation, summarisation and visualisation methods for N-masked *k*-mer trees, we can begin to review their effectiveness as tools for the description of sequence sets. The objective of this research was to find methods of describing large self-contained sets of biological sequence in a manner which yielded an 'at a glance' impression of the content of that sequence, and that this signature ought to also be composed of datapoints which could be dissected into the points of biological origin that composed such an image.

One of aspects which this work has overlooked to some extent is the potential for further dissection of the signature categories. Any given signature category could, for example, have a paired set of annotation labels from the input set, and could be described in its proportional representation of those categories. This could be secondary structures, or domain types in protein families. It could be functional attributes of entire proteins (cell signalling, transmembrane, DNA binding, enzyme, matrix structure and so forth), in the case of proteomes, or it could be any prominent DNA annotation in the case of genomes (known binding motifs, intra/extra-genic DNA, intronic/exonic, LTRs, etc.).

Although a certain degree of human error might be introduced by adding annotation categories to signatures, this might also further the informativeness of the decomposition of a signature. For example, in the case of *L. rubellus* it could show at a glance which proteins were creating the huge separations in banding patterns, or the in the case of Mucins, the glycosylated repeated domains causing the total distinctness collapse. As every point in the *k*-mer tree has spatial sequence origins it would not be difficult program the propagation annotation categories throughout the tree in the same manner as basic frequencies. Through a hash-map of parallel storage variables in the Node class, an arbitrary dictionary of annotation types could be fit to the tree at run-time. This would however have a larger memory footprint. Another method of programming the annotation overlay with a lower memory footprint, but a higher time-cost, could be to generate the tree as many times as annotation categories are present, using only annotated sequence each subsequent tree. The final set of signatures could then be merged into a categorical ratio overlay.

Many of the observations made of signatures in this work have been post-hoc speculations on the origins of signature sub-formations, each of which would need further investigation to flesh out into suitably dependable theories that might be relied upon in future research. The alternative annotation methods for the tree might thus be a reverse search capacity whereby the user selects signature components of interest, and searches an annotated sequence set for its constitutive spatial information. This might be as simple as re-building the tree, and searching (&DFS subtree

merging) for only a specific set of N-mask patterns, and converting all discovered sequences with any degree of distinctness into a finite state automata, in a similar manner to BLAST (Camacho et al. 2009). This approach might have the advantage, compared to the above paragraph, of allowing the user to search a specific pattern type across a far wider array of annotations without the per-annotation time cost, and with a fixed peak memory footprint. The disadvantage being the signatures remain initially quite abstract.

At present the signature system appears to function quite well at processing entire proteomes, however the DNA processing capacities have limited its application in other ways. A point of interest might have been the *Lumbricus rubellus* and *Lingula anatina* genomic signatures in comparison to other less-divergent genomes in the same clade, or in comparison to the more well researched model species. However, further efficiency gains will need to be made before such a comparison is possible.

3.5.2. Experimental scope for performance gains

In the context of the possibility of major efficiency improvements, there are several parameters that sequences signatures could be expanded. These are 1) the dimensionality of the N-mask, 2) the depth of the tree, and 3) the size of the input sets.

3.5.2.1. N-Mask Dimensionality

Regarding the first parameter, N: It would be ideal to be able to expand the maximum number of Ns in each mask up to $k - log(f_r)$, meaning that all N-masked sequences in the tree would still reach null saturation by their leaves. A more comprehensive summary of the structure, and even more sensitive detection of sparse motifs would be also be achievable if techniques were developed to compute the complete set of all (2^n) N-masks per k-mer, however a more reasonably expectable near-term objective might be a moderate expansion of the N-count. One primary inefficiency of the current method are the excessive heap memory allocation and deallocation during the creation and destruction of subtrees in the merging function. This could be replaced by some fixed size pre-allocated working memory used for tree merges. Another bottleneck in terms of clock cycles is the repetitive paired-DFS function executions required to merge subtrees. It might be possible to by subtraction discover extra merged node combinations for 'free' by storing multiple different merged frequency scores in the same node.

3.5.2.2. Tree Depth Limitation

The depth of the tree, unlike the N-mask, does not have a theoretical complete solution. In terms of absolute utility, the depth of the tree could extend to the length of the longest string in the input set, however this is also wildly unfeasible. At present, the frequencies which 'escape' the tree ranged

between 1-4% of the total for *E. coli* DNA, and 2-11% for the protein families, and 1.5-18% for the proteomes. It would perhaps be wise to suggest that frequency escape isn't always a bad thing, or that is ought to be considered an effect which renders the analysis compromised. For example, if analysing both alleles of a highly allelically divergent genomes, one might expect 30-40% of frequencies to escape, and be backwards subtracted from the tree. A hyper-conserved protein family could be expected to give a similar reading. The real consideration with depth is whether it captures the breakdown of the high frequency structures which are expected to break down, and which are meaningfully interpretable when they do so. Still, it remains desirable yet to extend the tree's depth, if only to discover the point at which it becomes ineffective.

3.5.2.3. Size of Input Sets

The memory and processing time capacity for larger input sets would be very useful, as mentioned several times, for the sake of full genome signatures. It could also be utilised for other large sequence inputs, such as metagenomes, which can be hard to analyse and whose description might be facilitated by sequence signatures. Another area of large sequence set inputs is transcriptomics. Since frequency is currently coded to 1-per-k-mer, there would be no significant performance penalty for coding *x*-per-*k*-mer, where *x* is the normalised read count spatially resolved for a transcript from a given sample. There are many possible input configurations which could be explored with performance gains, and memory footprint reductions.

3.5.3. Potential Experimental Applications

There are many unexplored potential use-cases for the signatures developed in this research. Those that will be expanded up here include 1) Traits and phylogenetic association, 2) Stress/dose response signatures, 3) Reference signature database development.

3.5.3.1. Intersection with Traits and Phylogeny

Taxonomic classification and the prediction of evolutionary history has become a very powerful tool for understanding evolutionary biology, particularly since the advent of NGS technology. With functionally informative signature tools, there exists the possibility to describe the '-omic scale' architectures many types of lifeform, and to intersect these outputs with the structure of phylogenetic trees of various scales. This might concern a small monophyletic clade of species, or diverse samples separated by 100s of millions of years. This may lead to the discovery of 'typical' signatures for certain clades, or the possible association of signature types with other traits, such as environmental plasticity, k- or r-selection, life history, and other phenotypic qualities.

3.5.3.2. Stress/Dose Response Signatures

Stress responses in transcriptomes are a frequent research objective. Even the earlier transcriptomic studies revealed that organism stress responses can have huge impacts on the entire expression pattern of transcripts, for example, in 2002 *Arabidopsis thaliana* was found to have 30% of its transcripts in some way differentially regulated because of common stressors (Kreps 2002), and currently the results of transcriptome stress experiments have similar results across the board. For example, a 2017 study shows that in human mononuclear blood cells repress two thirds of their genes in response to heat stress, whilst up-regulating many others (Bouchama et al. 2017). Stress responses have the effect of drastically altering gene expression in most organisms. By encoding a k-tree with read-depth scaled frequency scores for all input k-mers, it would be possible to deploy a system of signature differentials for stress response, with the aim of qualifying the extent of a stress response in an organism which annotates very poorly when compared to the available references. There might also be a variety of signature differential types depending on whether stressors are singular or multiple, or based on their severity.

Another avenue of mathematical development which might aide this potential research direction would be the formalisation of distance metrics between signatures within the same set of samples. This would include estimation of null variance of category scores between replicates, and the testing of stress or other variable response samples against them. Distances would also be subject to the signatures form, with the possibility of 'distance signatures' displaying most prominently the range of distances exhibited by the categories or threads which separate the samples the most.

3.5.3.3. Reference Database Development

Although signatures are informative of sequence features by themselves, much of their value can come from comparison. The highly visual aspect of the output allows a researcher to very quickly see if one sequence set 'looks like' another one. By extension, the notion that digital means to quantitatively assess which signatures look-like which seems sensibly forthcoming. In terms of comparing between a small number of pre-calculated outputs this might be trivial, however much of the modern quest for biological insight comes in the form of queries directed at massive data banks via sophisticated search tools. Should enough signatures be generated, it would be useful for a signature-specific distance metric heuristic search function to be developed, such that a user might query a database of signatures with their own outputs to see which other organisms manifest similar sequence variation and structure patterns, in a manner that is liberated from comparisons of homology. The type of sequence data in this use case is not limited to the types of any of the test sets used in this work. However, a relatively low-computational cost first endeavour might be to

generated signatures for all 1000+ PFAM protein families, and to deploy a PFAM-indexed search function, as an adjunct the current protein family knowledge base.

3.6. Conclusion

A biological sequence signature creation tool was developed. The tool was applied to 5 invertebrate proteomes, 6 protein families, and 18 *E. coli* genomes, as test sets. The results showed that the peptide trees are capable generating varied and informative signatures of the sequence inputs, which relate directly to the biology of the sample sources. The DNA trees have yet to achieve the computational performance required to perform large scale analysis on genomic scales. Multiple aggregation methods were proposed for tree aggregation. The research then focused on two aggregation methods that were most suitable for the trees generated, given the performance boundaries of its parameters. Visualisation methods were developed for the outputs, with infographics also created to aide in the interpretation of the 'signature' plots. This work shows that it is possible to create dense and complex signatures of sequence structure, without sacrificing their biological utility, when highly optimised navigations of high dimensional space are deployed instead of generalisations which first seek to reduce it.