RESEARCH ARTICLE

Methods for evaluating unsupervised vector representations of genomic regions

Guangtao Zheng[7], Julia Rymuza[1], Erfaneh Gharavi[1,2], Nathan J. LeRoy[1, 3], Aidong Zhang[2, 3, 7], and Nathan C. Sheffield[1-6,✉]

[1]Center for Public Health Genomics, School of Medicine, University of Virginia, 22908, Charlottesville VA
[2]School of Data Science, University of Virginia, Charlottesville VA 22904, Charlottesville VA
[3]Department of Biomedical Engineering, School of Medicine, University of Virginia, 22904, Charlottesville VA
[4]Department of Public Health Sciences, School of Medicine, University of Virginia, 22908, Charlottesville VA
[5]Department of Biochemistry and Molecular Genetics, School of Medicine, University of Virginia, 22908, Charlottesville VA
[6]Child Health Research Center, School of Medicine, University of Virginia, 22908, Charlottesville VA
[7]Department of Computer Science, School of Engineering, University of Virginia, 22908, Charlottesville VA
✉ Correspondence: nsheffield@virginia.edu

Background: Representation learning models have become a mainstay of modern genomics. These models are trained to yield vector representations, or embeddings, of various biological entities, such as cells, genes, individuals, or genomic regions. Recent applications of unsupervised embedding approaches have been shown to learn relationships among genomic regions that define functional elements in a genome. Unsupervised representation learning of genomic regions is free of the supervision from curated metadata and can condense rich biological knowledge from publicly available data to region embeddings. However, there exists no method for evaluating the quality of these embeddings in the absence of metadata, making it difficult to assess the reliability of analyses based on the embeddings, and to tune model training to yield optimal results. **Methods**: To bridge this gap, we propose four evaluation metrics: the cluster tendency test (CTT), the reconstruction test (RCT), the genome distance scaling test (GDST), and the neighborhood preserving test (NPT). The CTT and RCT are statistical methods that evaluate how well region embeddings can be clustered and how much the embeddings can preserve the information contained in training data. The GDST and NPT exploit the biological tendency of regions close in genomic space to have similar biological functions; they measure how much such information is captured by individual region embeddings and a set of region embeddings. **Results**: We demonstrate the utility of these statistical and biological tests for evaluating unsupervised genomic region embeddings and provide guidelines for learning reliable embeddings. **Availability:** Code is available at https://github.com/databio/geniml.

Introduction

Genomic regions, or intervals, define functional elements in the genome, such as enhancers, promoters, or transcription factor binding sites [1]. A region is represented as a pair of coordinates marking its location on the genome. A set of such regions, often stored in Browser Extensible Data (BED) format, can be used to represent an epigenomics experiment that identifies locations of interest produced by a biological experiment, such as ChIP-Seq [2] or ATAC-Seq [3, 4]. Through broad effort [1, 5], the amount of epigenome data has rapidly increased, and almost 100,000 BED files are now available on the NCBI Gene Expression Omnibus (GEO) [6–8]. This growing resource of BED files contains rich biological information and have been used to interpret the human genome and drive discovery in genetic variation and gene regulation [9–12]. However, the ever-increasing volume of genomic region data also makes region-based analyses computationally costly, since analysis often requires computing region overlaps among region sets [13–15].

We recently introduced Region-set2Vec [16], an unsupervised method to learn representations for genomic region sets. Region-set2Vec learns low-dimensional vectors called embeddings as representations for genomic region sets, using only the region co-occurrence information contained in region sets. With Region-set2Vec, we can leverage the publicly available genomic region data, even without annotations, and can replace the costly region overlap calculation with more efficient vector similarity measures. To create region set embeddings, Region-set2Vec first must learn region embeddings, which are then averaged to represent the region set. We demonstrated the value of region set embeddings, but we reasoned that the underlying region embeddings could be useful independently; for example, we can infer the function of an unknown region based on its closeness to other known regions in the embedding space, or use them for fast annotation and clustering on single-cell data [17]. To develop such methods and concepts further, we first sought a way to evaluate region embeddings independently and objectively. Our previous study evaluated region set embeddings, but no

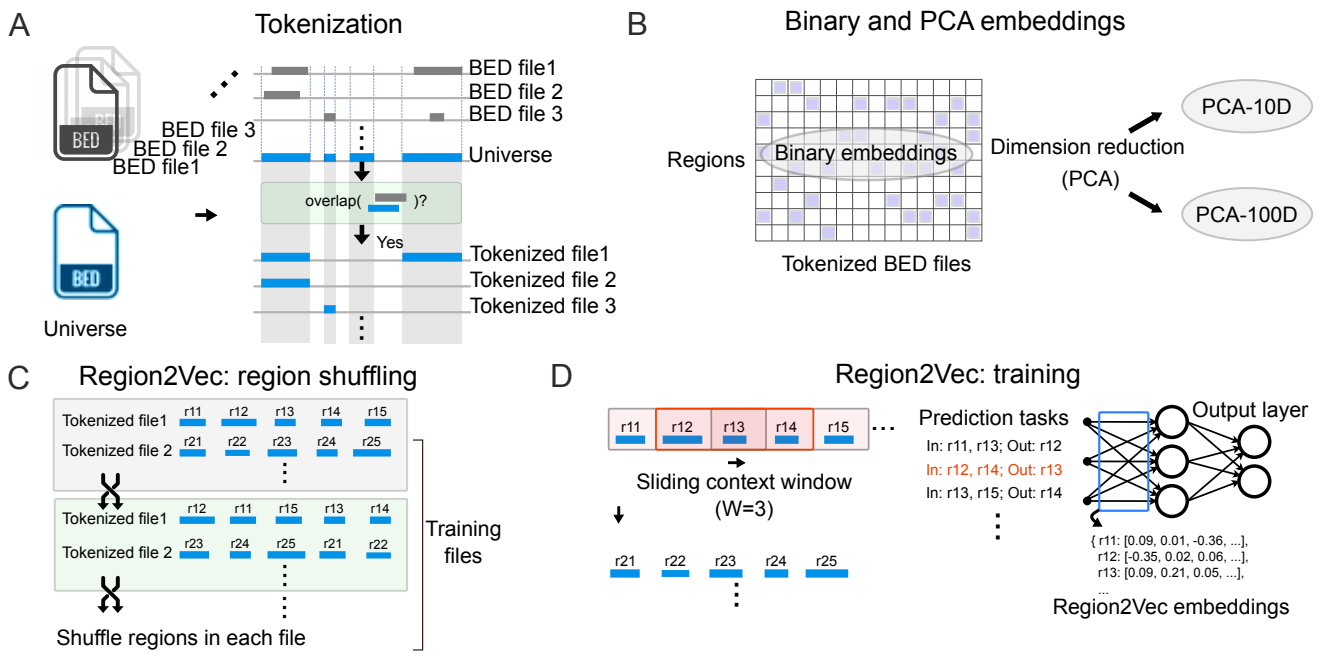

Figure 1: *Illustration of unsupervised region embedding generation.* ***A*** *Replace regions in BED files with overlapped regions in a universe. This is called tokenization.* ***B*** *Binary embeddings and the embeddings after keeping 10 (PAC-10D) and 100 (PCA-100D) principal components of the Binary embeddings.* ***C*** *Region2Vec shuffles regions in tokenized BED files to generate more training files.* ***D*** *Region2Vec trains the model using the prediction tasks created by a sliding context window with length W.*

method has yet been proposed to evaluate the quality of the underlying region embeddings.

To bridge this gap, we propose four novel statistical and biological tests to evaluate region embeddings. First, we propose two statistical tests that use only the training data and the region embeddings: 1) the cluster tendency test (CTT), which evaluates how well a set of region embeddings can form clusters; and 2) the reconstruction test (RCT), which evaluates information preserved in region embeddings by testing whether a region embedding can reconstruct the training data. Next, we propose two biologically-motivated tests that exploit the tendency of regions near one another on the genome to have similar biological functions. We reasoned that embedding distances that reflect functional similarity between regions should be biased toward smaller embedding distances for regions nearby in the genome. That is two embeddings that are close in latent space should be, on average, closer in genome space. To assess whether region embeddings capture this tendency, we propose the genome distance scaling test (GDST) and the neighborhood preserving test (NPT). The GDST calculates how the embedding distance between two regions scales with genome distance. We expect to obtain a positive and large scaling value if the embeddings recapitulate genome distance. The NPT calculates how much the neighborhood of a region in the genome space overlaps with the neighborhood of the same region in the embedding space.

These four tests are generally applicable to any set of region embeddings, including Region2Vec embeddings. To demonstrate, we applied the four tests to three different types of embeddings: 1) Binary embeddings; 2) dimensionality-reduced versions of Binary embeddings; and 3) Region2Vec embeddings trained with various learning parameters. Our results show the advantage of Region2Vec embeddings in capturing the desired biological knowledge with low dimensions, providing new evidence for the value of Region-set2Vec embeddings [16]. We also provide guidelines for training hyperparameters for learning reliable genomic region embeddings based on the results from the two tests.

Results

Overview of analytical approach

To guide our embedding evaluation methods development, we first collected a representative collection of region sets consisting of 690 transcription factor binding BED files. Next, since our embedding approaches require defining the input data in terms of a universe of consensus regions, we selected 7 test universes; this allowed us to explore how our tests vary by universe. For each universe, we generated region embeddings by first tokenizing the region sets (Figure 1A) followed by using the tokenized region sets as training data to produce embeddings. We produced region embeddings using three unsupervised methods: 1) Binary embeddings, which are 690-dimensional vectors where each dimension represents presence of a region in one of the 690 BED files (Figure 1B); 2) principal component analysis (PCA) embeddings, constructed as the top 10 (PCA-10D) or 100 (PCA-100D) principal components of the binary embedding matrix (Figure 1B, [18]); and 3)

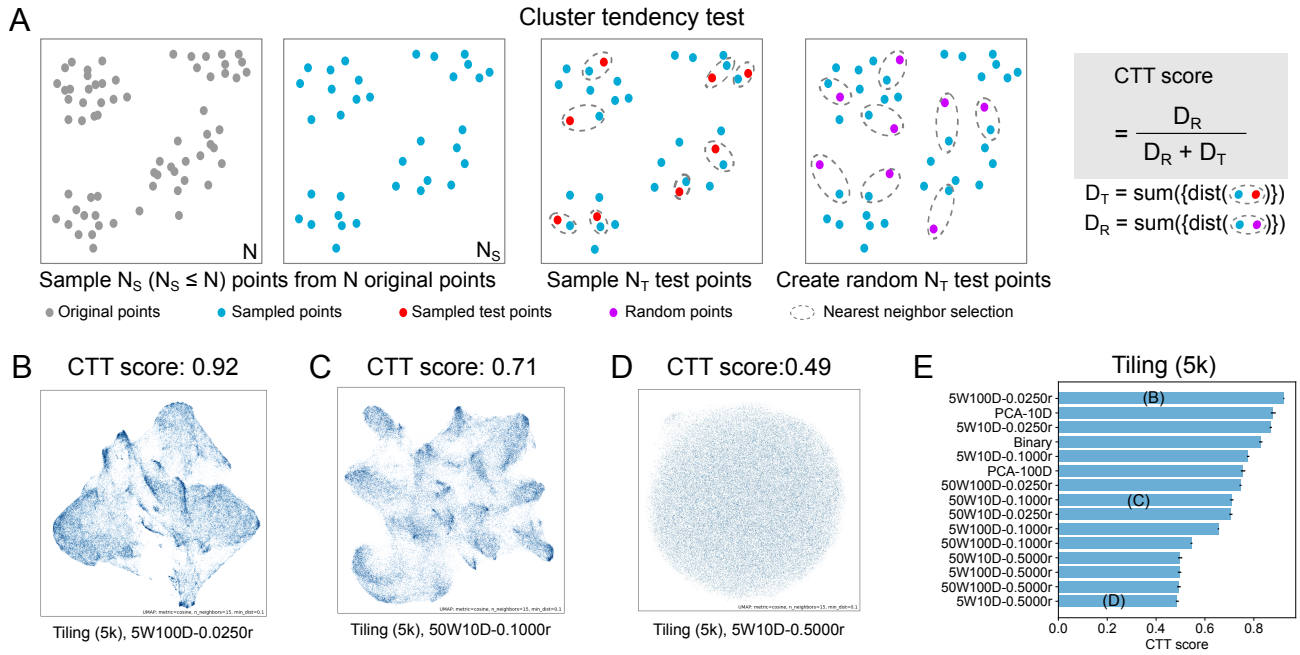

Figure 2: *Illustration of CTT and results for the Tiling (5k) universe.* ***A*** *Steps for calculating the CTT score.* ***B-D*** *UMAP visualizations of the 5W100D-0.025r, 50W100D-0.1000r, and 5W10D-0.5000r embeddings for regions in the Tiling (5k) universe.* ***E*** *CTT scores of different sets of embeddings for regions in the Tiling (5k) universe. Each blue horizontal bar and the associated black bar indicate the average and standard deviation of the CTT scores from 20 runs of the CTT for each set of region embeddings. W: context window size, D: embedding dimension, r: initial learning rate.*

Region2Vec embeddings, using 12 different parameter sets for training the Region2vec model (see Methods). Briefly, Region2Vec first randomly shuffles regions in each tokenized BED file (Figure 1C), then, similar to training word embeddings in Word2Vec [19], it uses a context window to create prediction tasks and updates Region2Vec embeddings (Figure 1D) [16]. For each of these 15 sets of embeddings (Table S1), we applied our proposed statistical and biological tests, allowing us to compare their performance.

Test 1: Cluster Tendency Test (CTT)

The Cluster Tendency Test (CTT) is a statistical test that analyzes whether region embeddings can be clustered. Our rationale is that embeddings that form clusters are more likely to be useful than embeddings that are diffuse. To calculate the CTT score, we first sample $N_S = \min(10^4, N)$ region embeddings from the original N embeddings to reduce the computational complexity. Then, from the N_S embeddings, we sample $N_T = 10\% \cdot N_S$ test points. For each of the N_T test points, we compute the distance to its nearest neighbor among the N_S sampled points excluding itself. We get the distance summation over the N_T test points as D_T. We do the same for N_T random points and get the distance summation as D_R (Figure 2A). The final CTT score is computed as $\frac{D_R}{D_R+D_T}$. If the embeddings being tested deviate from randomly distributed ones, then D_R will be large, and D_T will be relatively small; therefore, their CTT score will be large, and we conclude that the embeddings being tested tend to cluster (e.g., Figure 2B). In contrast, a lower CTT score indicates the embeddings are more dispersed throughout the embedding space (e.g., Figure 2C), with a score near 0.5 indicating near randomness (e.g., Figure 2D). The CTT score thus focuses on the spatial distribution of region embeddings and calculates how much those embeddings deviate from randomly distributed ones. To increase robustness, we computed the CTT score on 20 random repetitions and report the average.

We observed that the CTT score varied across embedding approaches, and that the CTT is robust to the choice of universes (Figure S1). Scores for the 5k tiling universe varied from 0.92 (5W100D-0.0250r) to 0.49 (5W10D-0.5000r) (Figure 2E). Interestingly, the PCA-10D embeddings have a higher score than the PCA-100D and Binary (Figure 2E), even though the PCA-10D retains less information from the training data than the PCA-100D and Binary. In other words, a high CTT score does not necessarily indicate better embeddings, since we can always generate a set of region embeddings with a very high score while being irrelevant to the training data. Instead, the CTT only evaluates the spatial distribution of region embeddings and how likely they can be clustered. For Region2Vec embeddings, we observed that small context windows (e.g., $W = 5$) achieved higher CTT scores than large context windows (e.g., $W = 50$). Moreover, using a large initial learning rate (e.g., $r = 0.5$) leads to training failures since the corresponding CTT scores are around 0.5, indicating that the embeddings are similar to uniformly random embeddings since $D_T \approx D_R$. Therefore, for training Region2Vec embeddings with good clusterability, avoid using a large r, and

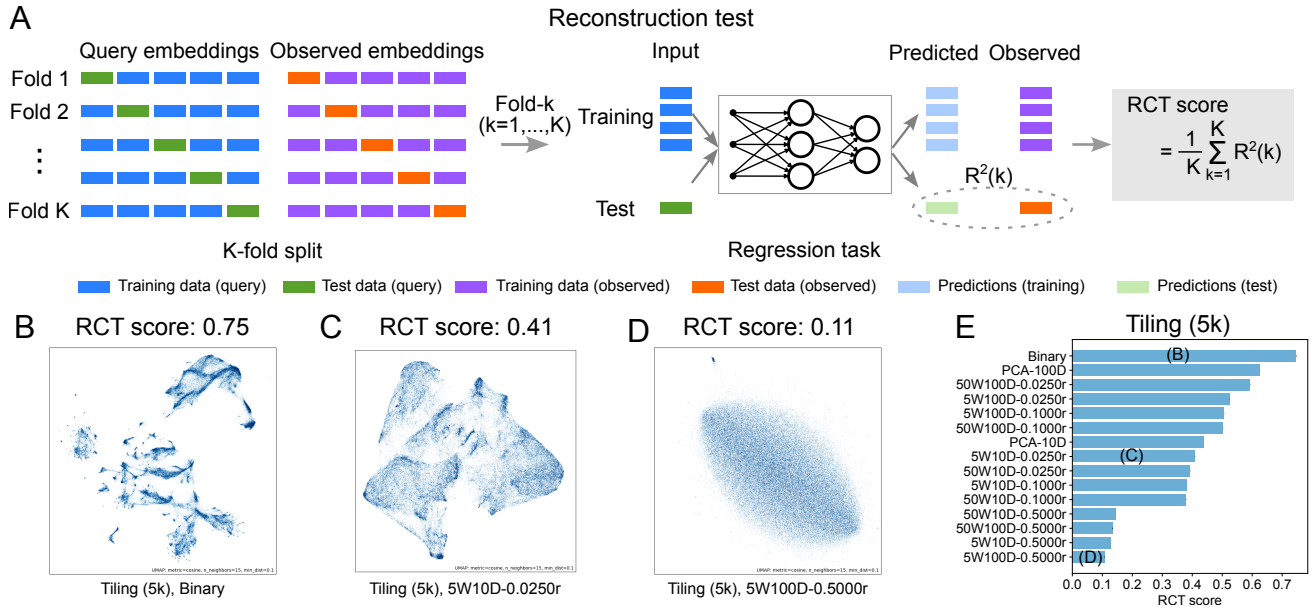

***Figure 3**: Illustration of the RCT and results for the Tiling (5k) universe. **A** Steps for calculating a RCT score. **B-D** UMAP visualizations of the Binary, 5W10D-0.0250r, and 5W100D-0.5000r embeddings for regions in the Tiling (5k) universe. **B** RCT scores of different sets of embeddings for regions in the Tiling (5k) universe. Each blue horizontal bar and the associated black bar indicate the average and standard deviation of the RCT scores from 5 runs of RCT for each set of embeddings. W: context window size, D: embedding dimension, r: initial learning rate.*

use a small W. We conclude that the CTT is useful to identify low quality region embeddings whose CTT score is near 0.5.

Test 2: Reconstruction Test (RCT)

One limitation of the CTT is that it evaluates only the spatial distribution of region embeddings, and does not assess the information contained in the embeddings. To address this, we sought a score that could evaluate embedding content more directly. To this end, we developed the Reconstruction Test (RCT), a statistical evaluation method that evaluates how much embeddings preserve the information about regions' occurrence among the training BED files. Intuitively, the RCT asks whether the embeddings can be used to reconstruct the full-dimensional input data. To compute the RCT score, we first define the Binary embeddings as "observed" embeddings, since they contain all the original data by definition. We define the trained embeddings as "query" embeddings. Each region thus has a "query" and an "observed" representation. We split the data into 5 folds (Figure 3A, K-fold split), then, use each fold to train a regression model that predicts the observed embeddings using the corresponding query embeddings (Figure 3A, Regression task). We use a three-layer feed-forward neural network with 200 hidden units and ReLU activations. The input dimension of the model depends on the given region embeddings, and the output dimension of the model is the number of BED files in the collection (690). The RCT score is the average R^2 score over the 5-fold cross-validation (see Methods). In theory, Binary embeddings should have the highest RCT score of 1; however, the model cannot perfectly be trained to be an identity function, so we consider the score of the Binary embeddings to be an upper bound. Embeddings with a high RCT score tend to have rich structures (e.g., Figure 3B and C); while those with a small RCT score tend to be randomly distributed (e.g., Figure 3D). This observation is similar to that in CTT and is further validated by the correlation between CTT and RCT scores (Figure S6). However, the RCT score favors Region2Vec embeddings trained with a large context window ($W = 50$), whereas the CTT score favors those trained with a small context window ($W = 5$).

Computing the RCT scores across our embedding sets, we observed that the RCT results are generally robust to the choice of universe (Figure S2). As expected, Binary embeddings achieve the highest RCT score (Figure 3E). The PCA embeddings have decreasing RCT scores as dimensionality is decreased, consistent with fewer dimensions retaining less information about the original data (Figure 3E). Among the Region2Vec embeddings, across universes, 50W100D-0.0250r achieves the highest RCT score and 5W100D-0.5000r achieves the lowest; in general, using a large context window (e.g., $W = 50$), a large embedding dimension (e.g., $D = 100$), and a small initial learning rate (e.g., $r = 0.025$) scores higher, indicating that these training settings best preserve the region occurrence information contained in the training data (e.g., Figure 3E).

Test 3: Genome Distance Scaling Test (GDST)

The CCT and RCT scores provide useful insight into the clusterability and reconstruction ability of embeddings;

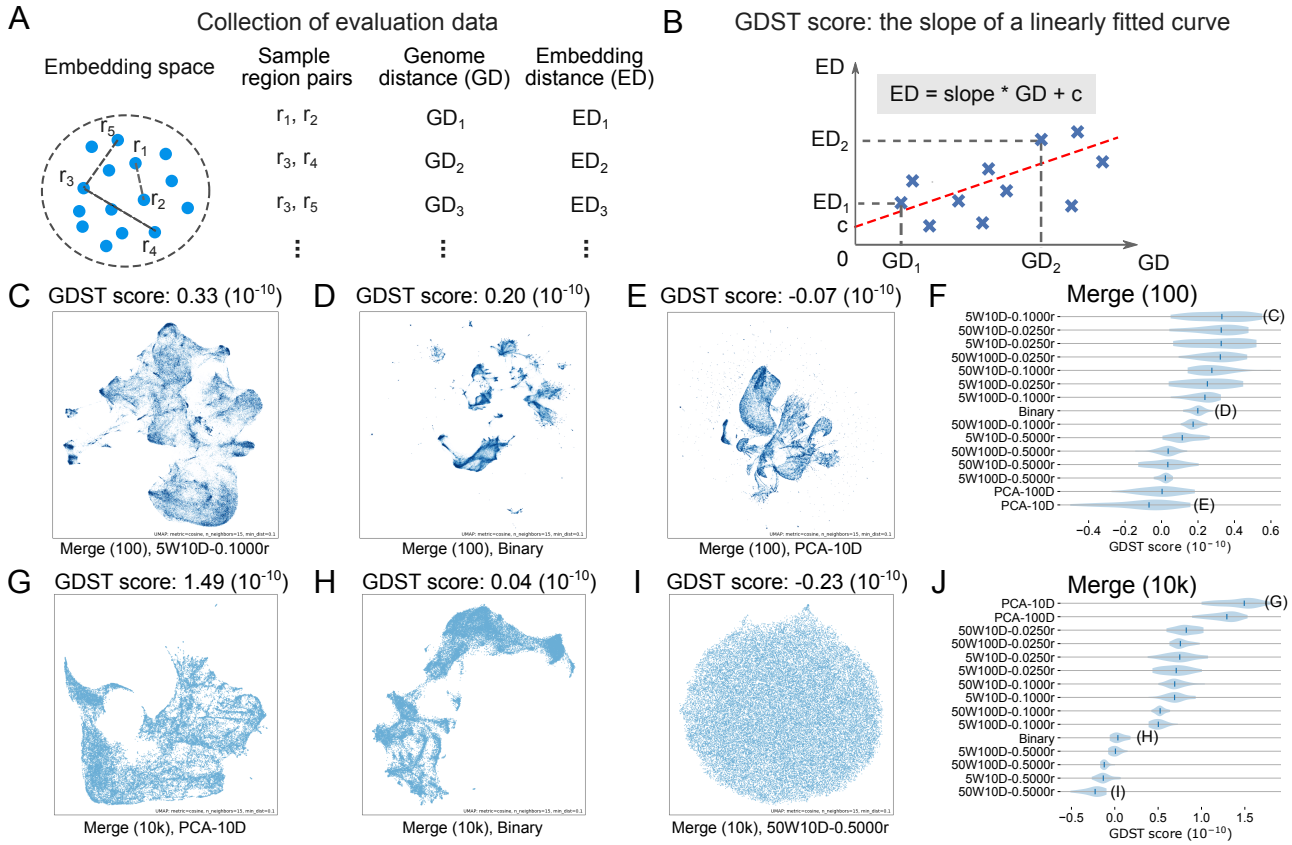

Figure 4: *Illustration of GDST and results for the Merge (100) and Merge (10k) universes.* ***A*** *Sample region pairs from a universe and calculate ED and GD for each pair.* ***B*** *Calculate the GDST score as the slope of the linear curve fitted on (ED, GD) pairs.* ***C-E*** *UMAP visualizations of the 5W10D-0.1000r, Binary, and PCA-10D embeddings for the Merge (100) universe.* ***G-I*** *UMAP visualizations of the PCA-10D, Binary, and 50W10D-0.5000r embeddings for the Merge (10k) universe.* ***F,J*** *GDST scores for different sets of region embeddings for the Merge (100) and Merge (10k) universe, respectively. Each shaded area and blue vertical bar indicate the distribution and median of GDST scores over 20 runs of GDST for each set of embeddings. W: context window size, D: embedding dimension, r: initial learning rate.*

however, neither of them considers biological knowledge. Thus, we next sought to devise evaluations that take into account the biology. To this end, we developed the Genome Distance Scaling Test (GDST), which analyzes how the embedding distances vary with the corresponding genome distances for a set of region embeddings. In the GDST, we first randomly sample 100k pairs of regions and calculate the embedding distance (ED) and the genome distance (GD) for each region pair (Figure 4A). Then, we compute the line of best fit over the 100k (GD, ED) points. The GDST score is the slope of the fitted line showing the global trend of ED as GD increases (Figure 4B). A positive score indicates that the embeddings capture the biological knowledge that regions near one another on the linear genome tend to have similar biological functions. The magnitude of the score indicates the degree of relationship between the embedding distance and genome distance.

Computing the GDST for each set of embeddings in Table S1 across universes, we observed that unlike the statistical tests, the GDST rankings vary substantially by universe. For example, in the `Merge (100)` universe, the PCA embeddings have the lowest GDST scores (Figure 4C), whereas, for the `Merge (10k)` universe, the PCA embeddings have the highest GDST scores (Figure 4G). We next sought to investigate why the universe selection has such a drastic effect on GDST. We reasoned that the tokenization under a universe could mask the shared function of nearby regulatory elements. For example, after tokenization, nearby regions are merged into large consensus regions, or they are completely missed due to the lack of coverage of the universe [20]. Then, the shared function signature could disappear, leading to lower GDST scores. To test this, we sought to determine the "fit" of each universe to the input data. We computed the F_{10} scores (see Methods), a score we developed recently [20] to measure how well a chosen universe covers the regions in the provided BED files (before tokenization). Briefly, the F_{10} score is computed as a weighted harmonic mean of the precision and recall when considering a query region set as a positive file with a universe as a prediction of that file. Larger F_{10} scores indicate the universe is a better fit for the regions in the collection. Computing F_{10} on the seven universes, we found that the Merge (100) universe has a higher F_{10} score ($F_{10} = 0.6281$) than the Merge (10k) universe ($F_{10} = 0.1990$), indicating it is a better fit to the input collection. This may explain why the embeddings tok-

enized with different universes have such different rankings; using the `Merge (100)` universe for tokenization can preserve more information contained in raw BED files and is less noisy than tokenizing with Merge (10k), for which simply using PCA to reduce noise works better than using Region2Vec to learn from very noisy data.

We next show how the GDST score varies visually. We observed that GDST score is higher for embeddings with low-density clusters. For example, we can observe large and sparse clusters from the high-scoring embeddings 5W10D-0.1000r for Merge (100) (Figure 4C) and PCA-10D for Merge (10k) (Figure 4G). In contrast, we can observe dense clusters from the low-scoring embeddings Binary for Merge (100) (Figure 4D), PCA-10D for Merge (100) (Figure 4E), and Binary for Merge (10k) (Figure 4H). For almost random embeddings, there is no clear cluster (e.g., Figure 4I). Similarly, Tilling (25k) has a low F_{10} score ($F_{10} = 0.1591$), and PCA embeddings perform the best in GDST (Figure S3F). Interestingly, Region2Vec embeddings perform best when the universe is a good fit to the BED files (Figure S3A,B,D,E,G). This indicates that Region2Vec is capturing biological knowledge in the embeddings. We conclude that using a universe with a high F_{10} score is important for Region2Vec. The context window size W depends on the size of the universe. Using $W = 5$ is preferred for a large universe (Figure S3A,D,G), while using $W = 50$ is preferred for a relatively small universe (Figure S3B,C,E,F). Avoid using a very large initial learning rate (e.g., $r = 0.5$) since it leads to very low GDST scores. In most cases, using a small embedding dimension (e.g., $D = 10$) is beneficial for achieving a high GDST score.

Test 4: Neighborhood Preserving Test (NPT)

Having proven that region embeddings, when trained on well-fit universes, can reflect genome distance, we next sought to extend the GDST beyond individual region pair distances to consider local neighborhood structure. We thus developed the Neighborhood Preserving Test (NPT), which finds whether the regions neighboring on the genome are also close in the embedding space. Like the GDST, the NPT relies on the general tendency of biological regions to cluster by function. To achieve this, NPT calculates the overlap between a region's neighboring set in the genome space and the corresponding set in the embedding space. Specifically, we randomly sample 10^4 regions, and for each sampled region, we obtain its sets of K-nearest neighbor regions in the genome space ($\mathcal{B}$) and its K-nearest neighbor regions in the embedding space ($\mathcal{T}$; Figure 5A). Then, we calculate the overlap ratio for each sampled region (Figure 5A). The neighborhood preserving ratio (NPR) is the average of the overlap ratios for the 10^4 sampled regions. To correct for random overlaps, we measure the significance of the NPR for the query embeddings over random embeddings by calculating the SNPR, the log-ratio between the NPR for the query embeddings, qNPR, and the NPR for the random embeddings, rNPR (Figure 5B). We expect the NPT score to decrease as K increases since the sharing in biological function decreases as the neighboring

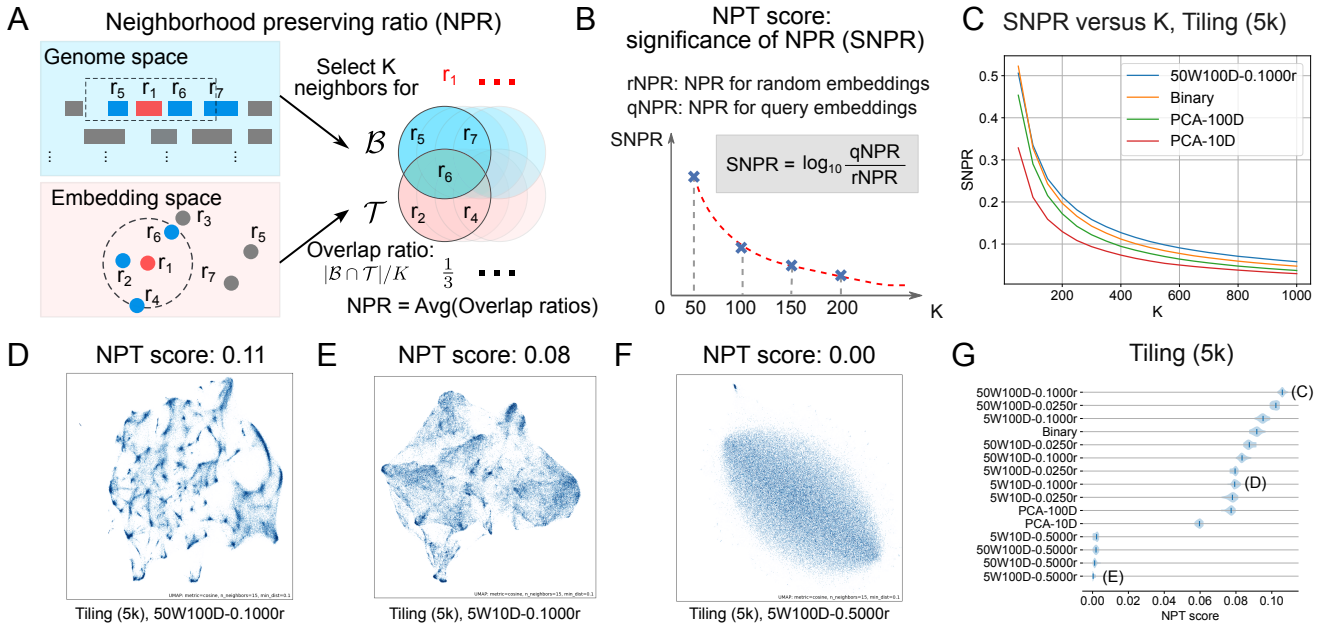

Figure 5: *Illustration of NPT and results for the Tiling (5k) universe.* ***A*** *Calculate NPR with a neighborhood size K as the average of the overlap ratios for all sampled regions.* ***B*** *Calculate the NPT score measuring how significant the NPR of a set of query embeddings is over a set of random embeddings.* ***C*** *SNPR versus K curves for the selected embeddings for the Tiling (5k) universe.* ***D-F*** *UMAP visualizations of the 50W100D-0.1000r, 5W10D-0.1000r, and 5W100D-0.5000r embeddings for regions in the Tiling (5k) universe.* ***G*** *Apply NPT with $K = 500$ to different sets of region embeddings for the seven universes, respectively. Each shaded area and blue vertical bar indicate the distribution and median of NPT scores over 20 runs of NPT for each set of embeddings. W: context window size, D: embedding dimension, r: initial learning rate.*

regions get further away in genome space. For each universe, we computed NPT scores for each of the 15 sets of generated region embeddings.

As K increases, we observe decrease in NPT scores (Figure 5C), which is expected. For comparison using NPT scores, we choose a large K, i.e., $K = 500$, since we are more interested in testing whether a set of region embeddings captures rich biological information which is often contained in a large neighborhood of regions. Although we choose $K = 500$ in Figure 5D-G, the ranks of different sets of region embeddings are stable under different Ks (Supplemental Figure S5A-G). An exception happens when K is small. For example, when $K = 50$, among different sets of region embeddings for the Merge (100) universe, Binary has the highest NPT score (Supplemental Figure S5A).

To achieve a high NPT score, the region's neighboring regions in the embedding space should be as close as possible so that they will not be included in other region's neighborhood set and decrease the overlap ratio for that region. Therefore, for a set of region embeddings with a high NPT score, we will observe many small and dense clusters. For example, we can observe small and dense clusters from the high-scoring embeddings 50W100D-0.1000r for Tiling (5k) (Figure 5D). We observe relatively sparse clusters for embeddings with smaller NPT scores (Figure 5E and F). Among different kinds of Region2Vec embeddings, the ones trained with the largest initial learning rate $r = 0.5$ have very small NPT scores, and the ones with the large context window size $W = 50$ and the large embedding dimension $D = 100$ have large NPT scores (Figure 5G). This observation is consistent across the seven universes (Figure S4). Moreover, for each of the seven universes, Binary, PCA-100D, and PCA-10D have decreasing NPT scores. As in other tests, using large initial learning rate ($r = 0.5$) leads to ineffective training of region embeddings and small NPT scores. High NPT scores are best achieved with a large context window (e.g., $W = 50$), a large embedding dimension (e.g., $D = 100$), and a small initial learning rate (e.g., $r = 0.025$).

Discussion and conclusion

In this paper, we proposed two statistical tests, CTT and RCT, and two biological tests, GDST and NPT, to evaluate the quality of genomic region embeddings learned from unsupervised methods. Each test evaluates different perspectives of region embeddings. The CTT tests how well region embeddings can be clustered, and the RCT tests how much original training information is preserved by region embeddings. The GDST calculates the degree to which region embedding distances scale with genomic region distance in the embedding space, while the NPT calculates how well region embeddings preserve a region's neighborhood in genome. These four tests provide complimentary perspectives that, when taken together, can provide a comprehensive evaluation of the statistical and biological value of region embeddings in an unsupervised way.

The tests also vary in terms of input requirements: CTT only needs region embeddings for evaluation, whereas the RCT also needs access to the training data (the tokenized BED files). The biological GDST and NPT do not need the training data, but they do need the genomic coordinates of regions in order to calculate genome distance. We have found the four tests useful in tandem to ensure region embeddings meets our needs. For example, we can first apply RCT to several candidate sets of region embeddings to filter out low-scoring sets that lose too much training information. Then, to ensure biological signatures are preserved, we can apply GDST or NPT to further select region embedding sets. Finally, we can apply CTT to find an easily-clustered region embedding set to facilitate downstream analyses.

Scores of the same test are not directly comparable across universes, since different universes have different sets of regions, resulting in different training data across universes. However, all the tests, except the GDST, yielded similar rankings regardless of universe. In fact, the GDST is the only one that was sensitive to the choice of universe. Although evaluating the quality of a universe is not our goal in this paper, we found a correlation between GDST score and F_{10} score, a measure of universe quality. Therefore, we propose that the GDST may be useful as a way to determine whether a universe is a poor fit, which coincides to the case when PCA embeddings outperform Binary embeddings, providing a way to assess a universe without the need to access the original BED files (before tokenization).

Overall, based on the results from the four tests, we found that with proper settings, embeddings learned from Region2Vec, are top performers in general. More specifically, the Region2Vec embeddings outperformed the binary and PCA embeddings in CTT, GDST and NPT. We notice that Region2Vec embeddings had lower RCT scores than PCA embeddings. We reason that Region2Vec exploits regions' co-occurrence information in the training data for dimensionality reduction with a finite-sized context window. It may lose certain region co-occurrence information contained in a long region sequence. Compared to Region2Vec, PCA performs relatively simple linear dimensionality reduction on the original data. Therefore, it is easier for a neural network to learn an inverse function for PCA embeddings to reconstruct the original data (hence, higher RCT scores) than for Region2Vec embeddings. Nevertheless, Region2Vec embeddings have demonstrated the ability to capture the simple biology that close regions on the genome have similar biological functions. Our tests also provide parameter selection guidelines for learning good Reigon2Vec embeddings under each test. We

expect that these scores will provide a useful basis for evaluation of unsupervised region embeddings, and also anticipate that they can form the basis of evaluation approaches for upcoming supervised region embedding methods as well [21].

Methods

Data overview

We collected 690 BED files from the ENCODE Uniform TFBS composite track[1] as a region set collection to generate and evaluate region embeddings.

To test how our proposed evaluation metrics are affected by region size, number of elements, and fit of the universe, we built seven universes with different numbers of regions and tokenized the 690 BED files into each (Table S2). These include 3 "Merge" universes, which we built by merging all regions in the collection using different distance thresholds. For example, For the "Merge (100)" universe, we merged all regions that were less than 100 bp apart. The other 4 universes are not derived from the BED file collection; the "Tiling" universes are created by simple tiling of fixed-size windows across the genome, and the DHS universe is an external universe defined as regulatory regions from DNase I hypersensitive sites (DHSs) with similar tissue specificity of DNase-seq signal patterns @[sheffield2013patterns]. For these non-data-driven universes, we filtered out any regions that do not appear in the tokenized BED file collection.

Tokenization of BED files into universe regions

To compare diverse region sets, many epigenome analysis methods, including our embedding methods, require first re-defining the raw regions into consensus regions. We refer to the consensus region set as a "universe," and the process of re-defining the raw regions as "tokenization." Tokenization allows us to use a shared vocabulary of genomic regions. This process is also required for NLP applications and methods. We tokenize by representing each original region as any universe regions that it overlaps. Original regions that do not overlap any universe region are discarded. Thus, after tokenization, tokenized BED files contain only unique universe regions. As a result, we need only consider learning embeddings for regions in the universe instead of original regions. Tokenization is the first step for any region embedding learning methods considered in this paper.

Binary and PCA embeddings

Binary region embeddings are parameter-free and are directly obtained from tokenized BED files. We represent these embeddings as a matrix $\mathbf{B} \in \mathbb{R}^{L_U \times N}$, where N is the number of tokenized BED files, and L_U is the size of the universe. If the lth region exists in the nth tokenized BED file, then the entry at the lth row and nth column is 1; otherwise the value is 0. The Binary embeddings can be very high-dimensional when the number of tokenized BED files N is large. To obtain lower dimensional embeddings, we apply PCA [18] to reduce the dimension of the Binary embeddings from N ($N = 690$ in our analysis) to 10 (PCA-10D) or 100 (PCA-100D).

Overview of Region2Vec method

Region2Vec embeddings are derived from the first step of Region-set2Vec [16], which first creates region embeddings, and then averages these to create region set embeddings. Region2Vec maps genomic regions to D-dimensional vectors. Region2Vec treats each region as a word, and treats regions in the same tokenized BED file as a sentence. In natural language, words in a sentence often serve to convey a single idea. Similarly, regions in the same BED file share a similar biological context and can be considered as words in a sentence. Then, Region2Vec leverages techniques in natural language processing (NLP) [22] to produce region embeddings. We use the gensim [23] implementation of Word2Vec to implement Region2Vec.

There are two steps in the Region2Vec training pipeline: region shuffling and training.

Region shuffling. Treating regions in a BED file as a sentence allows Region2Vec to exploit techniques [22] from NLP to produce region embeddings. However, these models are built for natural language, which imposes a specific order between regions. Regions in a BED file do not have any order and can be considered as neighbors of each other since they share the same biological context. To break such order, we randomly shuffle the regions in each region sentence obtained by reading regions in order from a tokenized BED file.

Region2Vec training. Region2Vec training uses the region co-occurrence information contained in region sentences to generate region embeddings. Specifically, for each region sentence, Region2Vec moves a context window with length W from the beginning to the end of the sentence. At each position of the window, the W regions covered by the window are used to update region embeddings via a prediction task. The task asks a Region2Vec model to predict the remaining region given $W - 1$ regions in the context window. We design the Region2Vec model as a three-layer neural network with weights $\mathbf{W}_I \in \mathbb{R}^{L_U \times D}$ and $\mathbf{W}_O \in \mathbb{R}^{L_U \times D}$ for the first and second half of the network, respectively, where L_U is the size of the universe used in tokenization, and D is the embedding dimension. The input for the model is a binary vector $\mathbf{a} = [0, \ldots, 1, 0, 1, \ldots, 0]$ of length L_U. The ones in $\mathbf{a}$ correspond to $W - 1$ regions in a context window. Then, the model shows how likely each

[1] http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeAwgTfbsUniform/

region will be in the context window with a probability distribution $\mathbf{p} \in \mathbb{R}^{L_U \times 1}$ over all the universe regions. Region2Vec aims to maximize the probability on the remaining region by optimizing $\mathbf{W}_I$ and $\mathbf{W}_O$. We move the context window and continue the above prediction and optimization procedures until all region sentences have been consumed. The obtained weight matrix $\mathbf{W}_I$ are the embeddings for all the universe regions.

Region2Vec embeddings training

We consider three important hyperparameters in Region2Vec: context window size W, embedding dimension D, and initial learning rate r. The context window size determines how many region embeddings are jointly updated each time. The embedding dimension specifies the number of vector dimensions used to represent a region. The initial learning rate controls the update magnitude for region embeddings. To explore how these hyperparameters affect the quality of the embeddings, we did a grid search, taking different values of each of these parameters in combination. We take W from $\{5, 50\}$, D from $\{10, 100\}$, and r from $\{0.025, 0.1, 0.5\}$ to generate 12 sets of region embeddings for each universe. During Region2Vec training, we shuffle the whole set of tokenized BED files 100 times. After each shuffling, the model is trained on the shuffled files, and after that, the learning rate is linearly decayed until reaching the minimum 10^{-4}.

Statistical Test: Cluster Tendency Test (CTT)

We use the idea from the Hopkins test [24] to design our test. Given a set of N region embeddings, we first sample N_S embeddings $\mathcal{Q} = \{\mathbf{q}_i | i = 1, \ldots, N_S\}$ from the given set to reduce the computational complexity, where $\mathbf{q}_i$ is a region embedding. This step is optional if the time and computational resources can support the evaluation with all N region embeddings. We found that using $N_S = 10^4$ is enough to produce stable results.

Then, from the N_S embeddings, we further subsample N_T embeddings $\mathcal{T} = \{\mathbf{q}_i^* | i = 1, \ldots, N_T\}$ as test points, where $\mathbf{q}_i^* \in \mathcal{Q}$. For each test point $\mathbf{q}_i^*$, we find its nearest neighbor $\hat{\mathbf{q}}_i$ in $\mathcal{Q} \backslash \{\mathbf{q}_i^*\}$, i.e., $\hat{\mathbf{q}}_i \in \mathcal{Q}$ and $\hat{\mathbf{q}}_i \neq \mathbf{q}_i^*$. We calculate D_T, the summation of all the distances between $\mathbf{q}_i^*$ and $\hat{\mathbf{q}}_i$, as $D_T = \sum_{i=1}^{N_T} d(\mathbf{q}_i^*, \hat{\mathbf{q}}_i)$. We define the distance function $d(\cdot, \cdot)$ as the squared Euclidean distance.

Next, we generate N_T random points $\mathcal{U} = \{\mathbf{u}_i | i = 1, \ldots, N_T\}$, that are uniformly distributed in the same area where the embeddings in $\mathcal{Q}$ reside. To do this, we uniformly sample $\mathbf{u}_i$ between $\mathbf{q}_{\min}$ and $\mathbf{q}_{\max}$, where $\mathbf{q}_{\min}$ and $\mathbf{q}_{\max}$ are vectors. Each dimension of $\mathbf{q}_{\min}$ ($\mathbf{q}_{\max}$) has the minimum (maximum) value of the corresponding dimension of the embeddings in $\mathcal{Q}$. Then, we calculate D_R, the summation of all the distances between $\mathbf{u}_i$ and its nearest neighbor $\tilde{\mathbf{q}}_i$ in $\mathcal{Q}$, as $D_R = \sum_{i=1}^{N_T} d(\mathbf{u}_i, \tilde{\mathbf{q}}_i)$. If there are clusters in $\mathcal{Q}$, then the random points should have large distances to their nearest neighbors in $\mathcal{Q}$.

We define the CTT score as $D_R/(D_R + D_T)$, a normalized ratio between D_R and D_T, ranging from 0 to 1. A larger CTT score indicates a large D_R and a greater tendency for the embeddings being evaluated to have clusters. When the embeddings are uniformly distributed, we have $D_R \approx D_T$, and the score is approximately 0.5. For evenly spaced embeddings, we have $D_T \gg D_R$, and the score approaches 0. Note that points in $\mathcal{U}$ lie in a hypercube and that the embeddings in $\mathcal{Q}$ may not distribute in that shape in practice. As a result, D_R may be overestimated, inflating the CTT score. For example, embeddings in $\mathcal{Q}$ can be uniformly distributed in a sphere within the hypercube, and sampling points in $\mathcal{U}$ at the corners of the hypercube will increase D_R, leading to a large CTT score even when there are no clusters in $\mathcal{Q}$. We address this by shrinking the hypercube a little bit, using 95% percentile and 5% percentile of each dimension data in $\mathcal{Q}$ to define $\mathbf{q}_{\max}$ and $\mathbf{q}_{\min}$, respectively.

Statistical Test: Reconstruction Test (RCT)

RCT evaluates how well an embedding of a region preserves the region's occurrence information in the training data. To measure how much such information is preserved in a region embedding, we create a regression task to predict the original input vectors, and evaluate the performance of region embeddings on this task.

Specifically, we use Binary embeddings (Binary and PCA embeddings in Methods) to represent regions' occurrences in the training data. These Binary embeddings $\mathbf{B}$ (row vectors) are the labels in the regression task. The query embeddings $\mathbf{Q}$ (row vectors), i.e., the region embeddings being evaluated, are the inputs in this task. We use a neural network f_θ with parameters θ to learn the mapping between the inputs and the labels. The performance of f_θ reflects quality of $\mathbf{Q}$ in this test.

To learn and evaluate the neural network f_θ, we adopt a K-fold cross-validation and split $\mathbf{B}$ and $\mathbf{Q}$ into K folds, $\mathbf{B} = [\mathbf{B}_1, \ldots, \mathbf{B}_K]$ and $\mathbf{Q} = [\mathbf{Q}_1, \ldots, \mathbf{Q}_K]$. At the kth step, we use $\mathbf{B}_k$ and $\mathbf{Q}_k$ for evaluation and the remaining folds for learning f_θ. We use mean-squared error as the training loss. During evaluation, we use $R^2(k)$ score for the kth fold defined as follows

$$R^2(k) = \frac{\sum_i \|\mathbf{b} * i^k - f * \theta^k(\mathbf{q}_i^k)\| * 2^2}{\sum * i \|\mathbf{b}_i^k - \bar{\mathbf{b}}^k\|_2^2}, \qquad (1)$$

where $\| \cdot \|_2^2$ denotes the squared 2-norm of a vector, $\mathbf{b}_i^k$ denotes the ith row of $\mathbf{B}_k$, $\mathbf{q}_i^k$ denotes the ith row of $\mathbf{Q}_k$, $\bar{\mathbf{b}}^k$ denotes the row average of $\mathbf{B}_k$, and θ^k denotes the parameters learned at the kth step. We repeat the above process process for K steps. Finally, the RCT score for $\mathbf{Q}$ equals $\frac{1}{K} \sum_{k=1}^{K} R^2(k)$. To make the RCT scores for different sets of embeddings comparable, we keep the architecture of the neural network f_θ fixed.

We preprocessed both the query and observed embeddings so that they have zero mean and unit variance. We used the implementation from scikit-learn [25] and applied the RCT to the region embeddings that we generated for the seven universes.

Biological Test: Genome Distance Scaling Test (GDST)

The GDST analyzes how the embedding distance (ED) between two regions changes with their genomic distance (GD). The ED and GD reflect two different relationships between two regions; the ED reflects their region co-occurrence in BED files, whereas the GD reflects their distance in genome space. Since regions that are close on the genome tend to have more similar biological functions than regions that are distant on the genome, we reasoned that, on average, the GD could be used as a noisy proxy for *some* shared biological function. Of course, regions that are distant from one another on the genome may be functionally similar, and may also be near one another in embedding space because of their functional similarity in biology. However, on average, we expect GD to be smaller between two regions that are similar than between two random regions. In training the embeddings, the location on the genome is not provided to the model. Therefore, any correlation between genomic distance and embedding distance reflects biological learning accomplished by the training process. We exploit this expectation to design the GDST, reasoning that embeddings that capture this biological signal will have GD positively correlated with ED.

Therefore, in GDST, we first randomly sample N_S region pairs from the universe and calculate their ED and GD values. The GD between two regions $GD(r, \tilde{r})$ is calculated as follows,

$$GD(r, \tilde{r}) = \begin{cases} \infty & c \neq \tilde{c} \\ \max(s - \tilde{e}, \tilde{s} - e, 0) & c = \tilde{c} \end{cases}, \quad (2)$$

where $r = (c, s, e)$ ($\tilde{r} = (\tilde{c}, \tilde{s}, \tilde{e})$), c ($\tilde{c}$) represents the chromosome index of r ($\tilde{r}$), s ($\tilde{s}$) and e ($\tilde{e}$) denote the start and end positions of r ($\tilde{r}$), respectively. Therefore, if two regions overlap, the GD is zero; if they are on different chromosomes, the GD is infinity; otherwise, the GD is the smallest number of base pairs connecting the two regions. The ED between the two regions $ED(r, \tilde{r})$ is the cosine distance between their region embeddings because the distance is immune to the artifacts caused by the magnitudes of region embeddings or the embedding dimensions. For example, the Euclidean distance between two region embeddings obtained from Region2Vec can be simply enlarged using a large learning rate or a large embedding dimension, leading to a large GDST score. However, using cosine distance does not have this issue since the distance is bounded from -1 to 1.

Finally, we conduct a linear curve fitting for the N_S ED and GD pairs and obtain the slope of the fitted curve as the GDST score for the set of region embeddings being evaluated. A positive score indicates the set of region embeddings captures the biological information that the closeness of regions in the genome reflects their functional similarity. The magnitude of the score indicates how strong such information is captured by the region embeddings. A large GDST score is often associated with embeddings with large and low-density clusters, where the embeddings tend to repel each other to have a large average ED.

The GDST is not a perfect test of biological information capture. In fact, a model that is provided with genomic locations in the training process could exploit the locations to place region embeddings near one another when the regions are near in genome space; however, such a model will not have learned anything useful. Nevertheless, for a model that is *not* provided with genomic locations in training, the ability to recapitulate genome distance to some degree provides evidence that the model has learned underlying biology.

Biological Test: Neighborhood Preserving Test (NPT)

The NPT evaluates how well the linear genomic neighborhood information of genomic regions is preserved in the embedding space. Compared with the GDST, the NPT does not evaluate individual region embeddings; instead, it focuses on local sets of region embeddings. It is based on the same assumption that a region's neighboring regions in the genome space tend to have similar biological functions, and whether the embeddings capture this relationships can be used as a measure of their quality. Hence, for good embeddings, we expect to observe a significant overlap between the neighboring region sets from the genome and the embedding space. NPT calculates SNPR to measure the significance of the overlap in average.

Specifically, NPT first calculates the overlap between the K-nearest neighbors of a region in genome space, $\mathcal{B}^K(r)$ and its K-nearest neighbors in embeddings space, $\mathcal{T}^K(r)$ of a region r. We use the GD defined in Eq. (2) and cosine distance as the ED. Then, NPT calculates the overlap ratio $\rho(r)$ for the region r as follows:

$$\rho(r) = \frac{|\mathcal{B}^K(r) \cap \mathcal{T}^K(r)|}{K}, \quad (3)$$

where $|\cdot|$ denotes the size of a set. Averaging over all $\rho(r)$ gives us the NPR for the set of region embeddings being evaluated. We denote such an NPR as qNPR. To measure how significant the NPR is, we calculate the NPR for a set of random region embeddings, denoted as rNPR. A random region embedding is created by sampling a value

from -0.5 to 0.5 for each embedding dimension. Finally, NPT calculates SNPR as

$$\text{SNPR} = \log_{10} \frac{\text{qNPR}}{\text{rNPR}}. \quad (4)$$

A large SNPR indicates the strong capability of the region embeddings to preserve the neighborhood information. When the value of SNPR is close to 0, the K-neighborhood preserving capability of the region embeddings is only as good as a set of random embeddings. Increasing K increases the distance of the regions on the genome and decreases the value of qNPR. Hence, SNPR will also decrease. In contrast, a very small K may lead to zero qNPR, making SNPR an ineffective measure of evaluating region embeddings. In our evaluation, we set K as a multiple of 50 so that we can effectively compare different sets of region embeddings using SNPR.

F_{10} score

We recently developed the F_{10} score as a way to assess fit of a universe to a collection [20]. Briefly, we first consider a genomic position covered by a raw region set in the collection and ask whether this position covered by the universe. We count the number of genomic positions that are covered in both as true positive (TP), the number covered only in the universe as false positive (FP), and the number covered only in the region set as false negative (FN). With TP, FP, and FN, we calculate precision and recall. Precision tells us the ratio of the universe covering the region set, and recall gives the ratio of the region set covering the universe. We then compute the F_{10} score, which is similar to the F score, and combines the precision and the recall with the harmonic mean, except we weigh the recall 10 times more weight than precision, due to the asymmetry of the relationship between universe and region set collection member. The asymmetry of the weights favors a universe that broader than the region set, since the universe needs to be more general to accommodate many different region sets. The F_{10} score ranges between 0 and 1, with 1 representing the perfect fit between the universe and the region set.

References

1. Consortium EP, others. An integrated encyclopedia of DNA elements in the human genome. Nature. 2012;489:57.

2. Furey TS. ChIP–seq and beyond: New and improved methodologies to detect and characterize protein–DNA interactions. Nature Reviews Genetics. 2012;13:840–52. doi:10.1038/nrg3306.

3. Buenrostro JD, Giresi PG, Zaba LC, Chang HY, Greenleaf WJ. Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position. Nature Methods. 2013;10:1213–8. doi:10.1038/nmeth.2688.

4. Smith JP, Sheffield NC. Analytical approaches for ATAC-seq data analysis. Current Protocols in Human Genetics. 2020;106:e101.

5. Research Network CGA (TCGA), others. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. Nature. 2008;455:1061.

6. Khoroshevskyi O, LeRoy N, Reuter VP, Sheffield NC. GEOfetch: a command-line tool for downloading data and standardized metadata from GEO and SRA. Bioinformatics. 2023;39:btad069. doi:10.1093/bioinformatics/btad069.

7. Barrett T, Wilhite SE, Ledoux P, Evangelista C, Kim IF, Tomashevsky M, et al. NCBI GEO: Archive for functional genomics data sets—update. Nucleic acids research. 2012;41:D991–5.

8. Xue B, Khoroshevskyi O, Gomez RA, Sheffield NC. Opportunities and challenges in sharing and reusing genomic interval data. Frontiers in Genetics. 2023;14. doi:10.3389/fgene.2023.1155809.

9. Portela A, Esteller M. Epigenetic modifications and human disease. Nature biotechnology. 2010;28:1057–68.

10. Maurano MT, Humbert R, Rynes E, Thurman RE, Haugen E, Wang H, et al. Systematic localization of common disease-associated variation in regulatory DNA. Science. 2012;337:1190–5.

11. Sheffield NC, Furey TS. Identifying and characterizing regulatory sequences in the human genome with chromatin accessibility assays. Genes. 2012;3:651–70.

12. Koch L. Non-coding mutations in the driver seat. Nature Reviews Genetics. 2014;15:575–5.

13. Sheffield NC, Bock C. LOLA: Enrichment analysis for genomic region sets and regulatory elements in r and bioconductor. Bioinformatics. 2016;32:587–9.

14. Quinlan AR, Hall IM. BEDTools: A flexible suite of utilities for comparing genomic features. Bioinformatics. 2010;26:841–2.

15. Dozmorov MG. Epigenomic annotation-based interpretation of genomic data: From enrichment analysis to machine learning. Bioinformatics. 2017;33:3323–30.

16. Gharavi E, Gu A, Zheng G, Smith JP, Cho HJ, Zhang A, et al. Embeddings of genomic region sets capture rich biological associations in lower dimensions. Bioinformatics. 2021;37:4299–306.

17. LeRoy NJ, Smith JP, Zheng G, Rymuza J, Gharavi E, Brown DE, et al. Fast clustering and cell-type annotation of scATAC data using pre-trained embeddings. 2023.

18. Abdi H, Williams LJ. Principal component analysis. Wiley interdisciplinary reviews: computational statistics. 2010;2:433–59.

19. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:13013781. 2013.

20. Rymuza J, Sun Y, Zheng G, LeRoy NJ, Murach M, Phan N, et al. Methods for constructing and evaluating consensus genomic interval sets. 2023.

21. Gharavi E, LeRoy NJ, Zheng G, Zhang A, Brown DE, Sheffield NC. Joint representation learning for retrieval and annotation of genomic interval sets. bioRxiv. 2023.

22. Le Q, Mikolov T. Distributed representations of sentences and documents. In: International conference on machine learning. PMLR; 2014. p. 1188–96.

23. Řehůřek R, Sojka P. Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. Valletta, Malta: ELRA; 2010. p. 45–50.

24. HOPKINS B, SKELLAM JG. A New Method for determining the Type of Distribution of Plant Individuals. Annals of Botany. 1954;18:213–27. doi:10.1093/oxfordjournals.aob.a083391.

25. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research. 2011;12:2825–30.

Supplementary information

***Table S1**: Configurations of the 15 types of embeddings.*

Embedding method	Name	Dimensions (D)	Window size (W)	Learning rate (r)
Binary	Binary	690	N/A	N/A
PCA	PCA-10D	10	N/A	N/A
	PCA-100D	100	N/A	N/A
Region2Vec	5W10D-0.0250r	10	5	0.025
	5W10D-0.1000r	10	5	0.1
	5W10D-0.5000r	10	5	0.5
	5W100D-0.025r	100	5	0.025
	5W100D-0.1000r	100	5	0.1
	5W100D-0.5000r	100	5	0.5
	50W10D-0.0250r	10	50	0.025
	50W10D-0.1000r	10	50	0.1
	50W10D-0.5000r	10	50	0.5
	50W100D-0.0250r	100	50	0.025
	50W100D-0.1000r	100	50	0.1
	50W100D-0.5000r	100	50	0.5

***Table S2**: Universes used in the experiments.*

Universe	Number of regions	F_{10}	Description
Merge (100)	287,225	0.6281	Merge regions when distance $<$ 100 bps
Merge (1k)	232,626	0.5296	Merge regions when distance $<$ 1k bps
Merge (10k)	40,498	0.1990	Merge regions when distance $<$ 10k bps
Tiling (1k)	426,840	0.4937	Each region has 1k bps
Tiling (5k)	250,857	0.2677	Each region has 5k bps
Tiling (25k)	99,996	0.1591	Each region has 25k bps
DNAse Hypersensitive Sites	561,580	0.4513	External universe

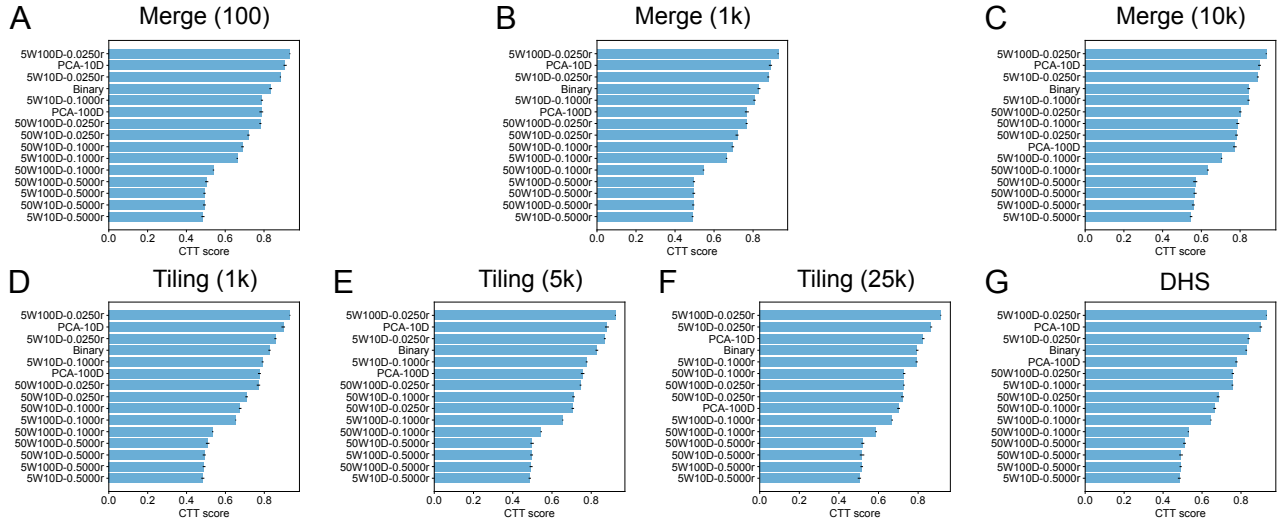

Figure S1: CTT scores of different sets of embeddings for regions in the seven universes. *Each blue horizontal bar and the associated black bar indicate the average and standard deviation of the CTT scores from 20 runs of the CTT for each set of region embeddings. W: context window size, D: embedding dimension, r: initial learning rate.*

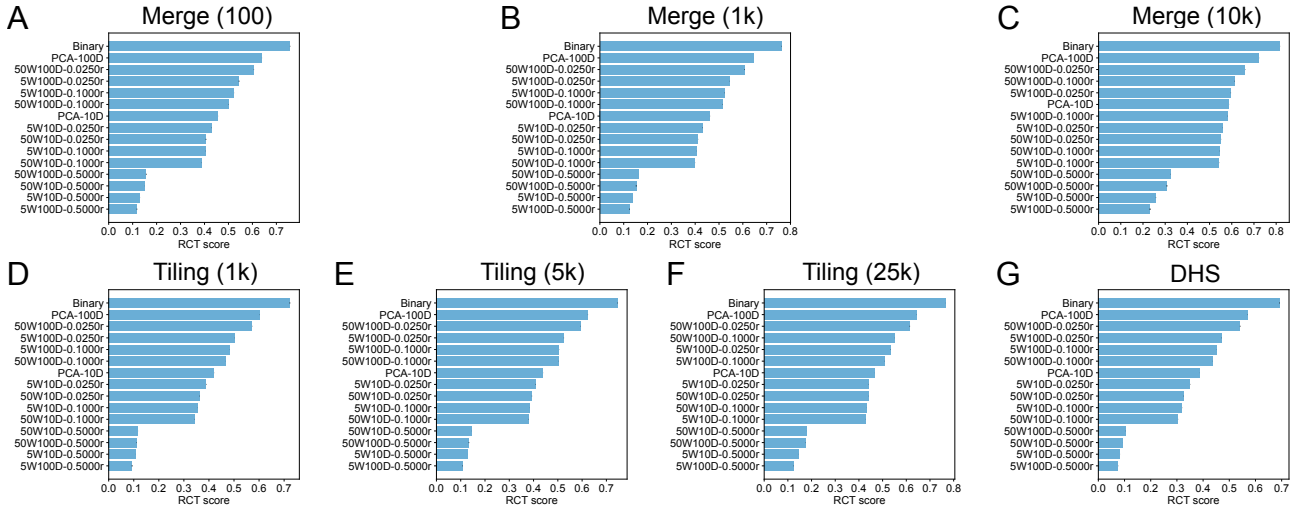

Figure S2: RCT scores of different sets of embeddings for regions in the seven universes. *Each blue horizontal bar and the associated black bar indicate the average and standard deviation of the RCT scores from 20 runs of the RCT for each set of region embeddings. W: context window size, D: embedding dimension, r: initial learning rate.*

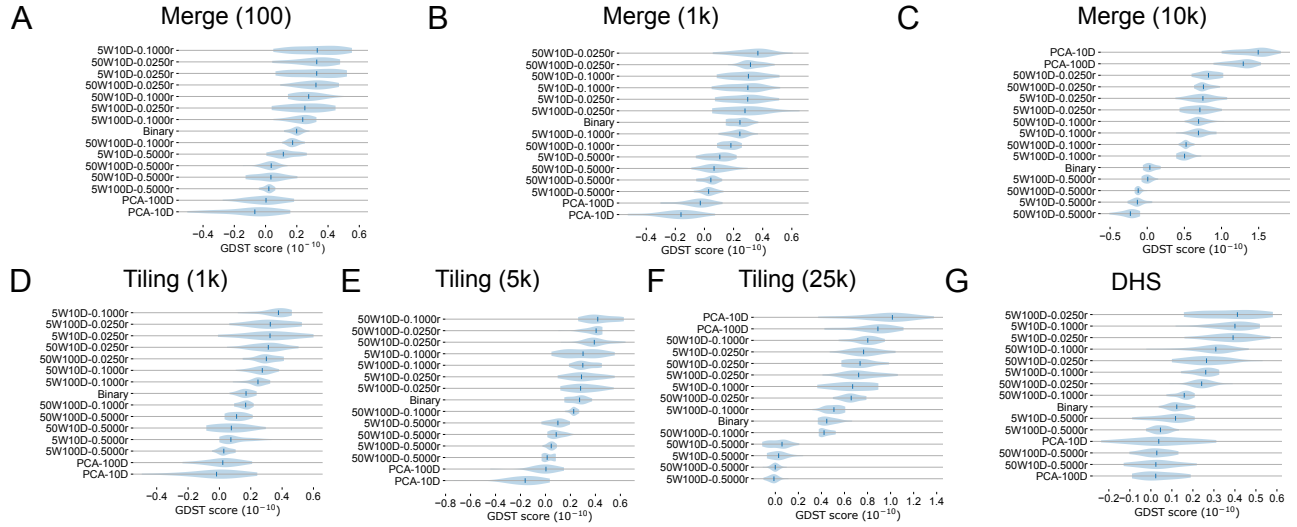

Figure S3: GDST scores of different sets of embeddings for regions in the seven universes. *Each blue horizontal bar and the associated black bar indicate the average and standard deviation of the RCT scores from 20 runs of the RCT for each set of region embeddings. W: context window size, D: embedding dimension, r: initial learning rate.*

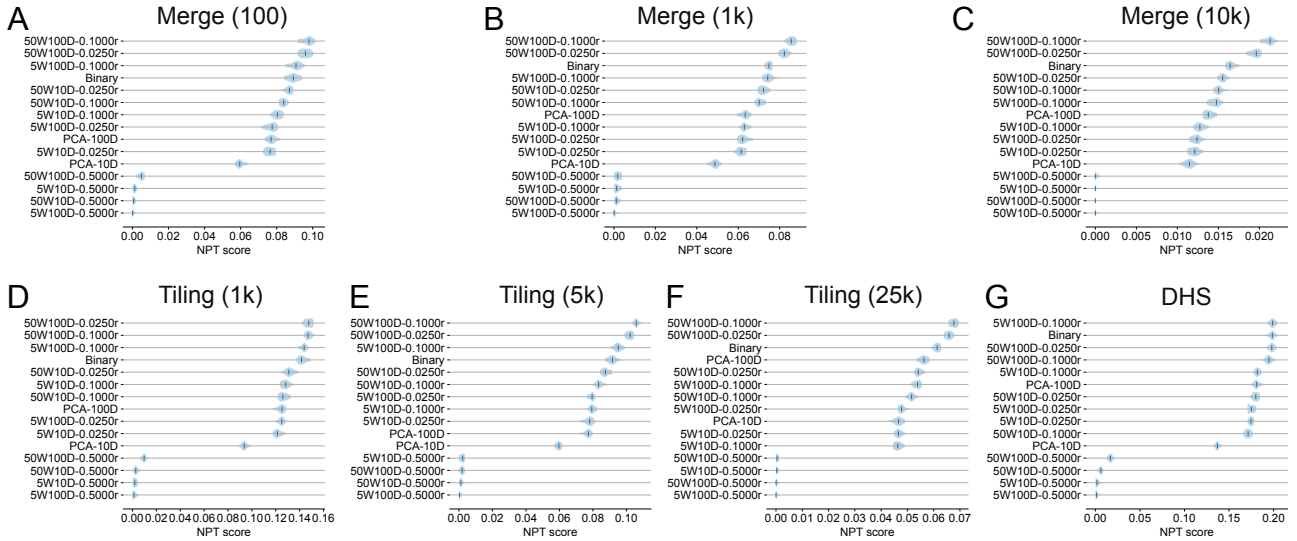

Figure S4: NPT scores of different sets of embeddings for regions in the seven universes. *Each blue horizontal bar and the associated black bar indicate the average and standard deviation of the RCT scores from 20 runs of the RCT for each set of region embeddings. W: context window size, D: embedding dimension, r: initial learning rate.*

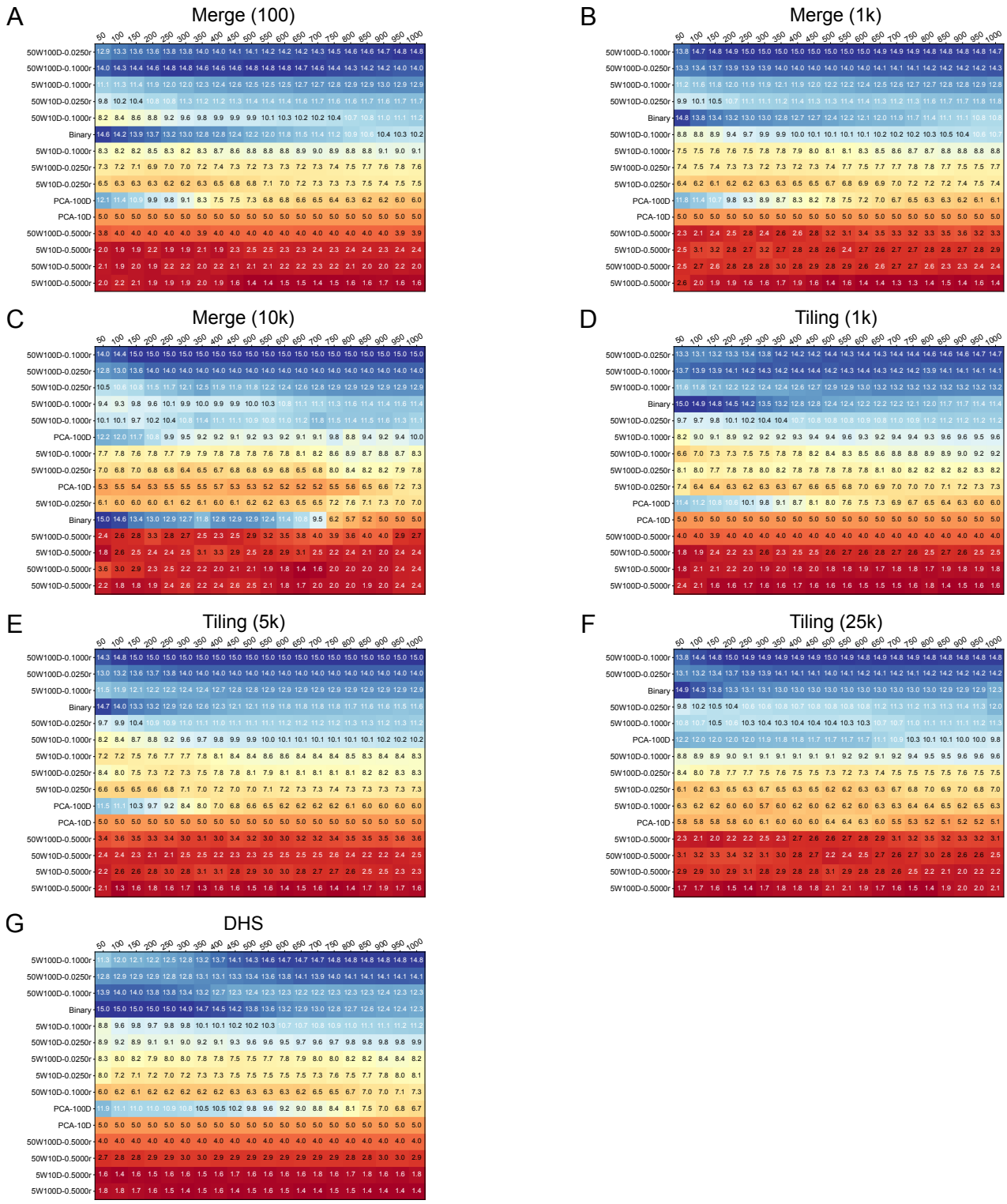

Figure S5: Average ranks of each set of region embeddings under different Ks over 20 runs of NPT. *A large rank indicates a high NPT score. W: context window size, D: embedding dimension, r: initial learning rate.*

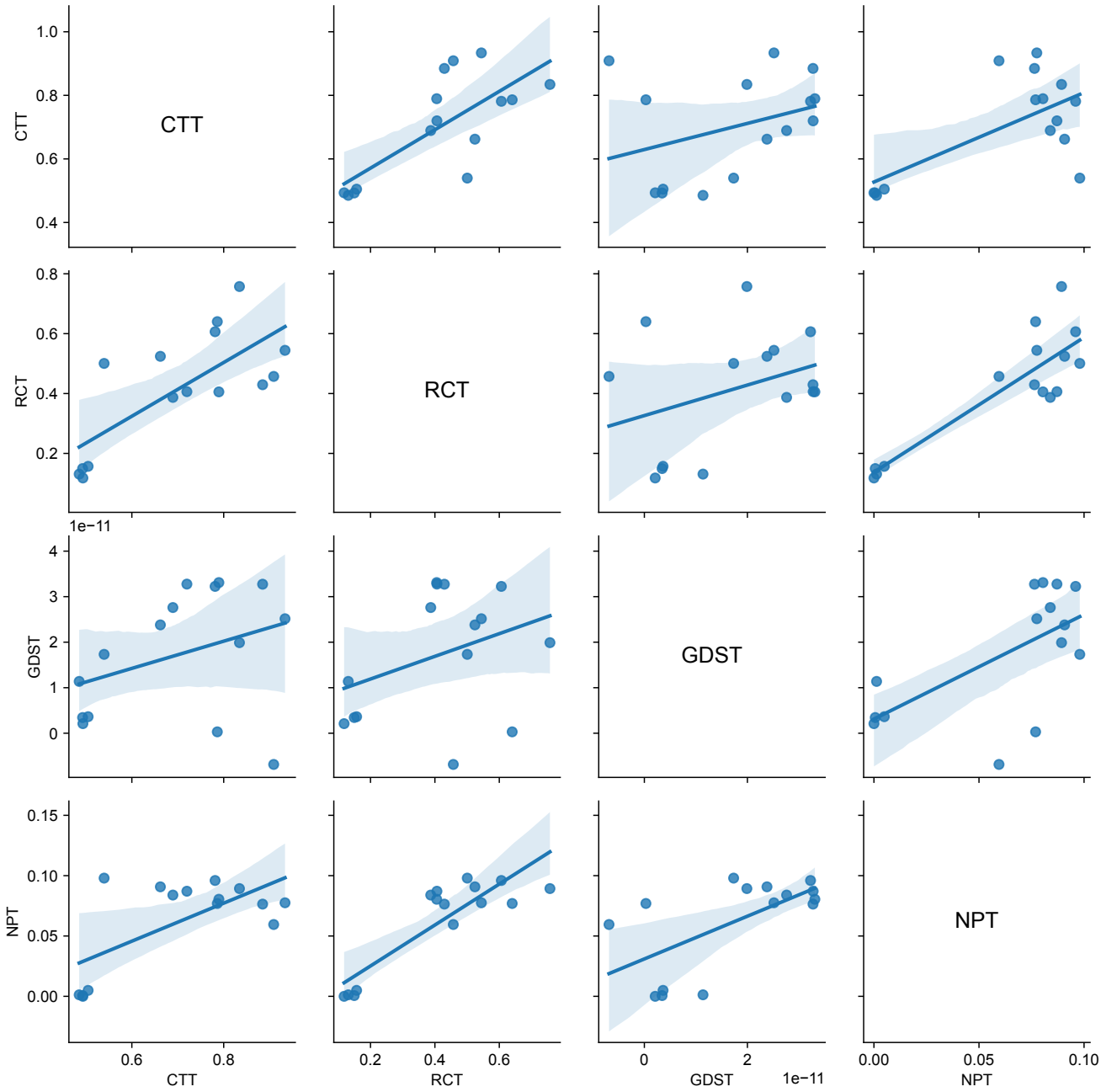

Figure S6: Pairwise correlations between the CTT, RCT, GDST, and NPT scores for the Merge (100) universe.