DSM: Deep Sequential Model for Complete Neuronal Morphology Representation and Feature Extraction

Hanchuan Peng (h@braintell.org)
Institute for Brain and Intelligence, Southeast University https://orcid.org/0000-0002-3478-3942
Feng Xiong
SEU-ALLEN Joint Center, Institute for Brain and Intelligence, Southeast University
Peng Xie
Institute for Brain and Intelligence, Southeast University
Zuo-Han Zhao
Institute for Brain and Intelligence, Southeast University https://orcid.org/0000-0003-1812-6237
Yiwei Li
Institute for Brain and Intelligence, Southeast University
Sujun Zhao
Institute for Brain and Intelligence, Southeast University
Lijuan Liu
SEU-ALLEN Joint Center, Institute for Brain and Intelligence, Southeast University

Article

Keywords:

Posted Date: June 29th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1627621/v1

DSM: Deep Sequential Model for Complete Neuronal Morphology Representation and Feature Extraction

Feng Xiong[1,2,#], Peng Xie[1,#], Zuohan Zhao[1,2], Yiwei Li[1,3] , Sujun Zhao[1,2], Lijuan Liu[1], Hanchuan Peng[1,*]

[1] SEU-ALLEN Joint Center, Institute for Brain and Intelligence, Southeast University, Nanjing, Jiangsu, China.
[2] School of Biological Science and Medical Engineering, Southeast University, Nanjing, Jiangsu, China.
[3] School of Computer Science and Engineering, Southeast University, Nanjing, Jiangsu, China.
[#] These authors contributed equally.
[*] Corresponding author: Hanchuan Peng <h@braintell.org>

Abstract

Full morphology of single neurons is indispensable to understand cell types, the basic building blocks in brains. It is critical to extract biologically relevant information from neuron morphologies. As projecting trajectories of neurons provide valuable information of both connectivity and cell identity, we developed an artificial intelligence method, Deep Sequential Model (DSM), to extract concise, cell-type defining features from projecting trajectories across brain regions. DSM achieves more than 90% accuracy in classifying twelve major neuron projection types, without compromising performance when spatial noises are present. Such remarkable robustness enabled us to efficiently manage and analyze two major full-morphology data sources. This study showcased those characteristic long projections can define cell identities.

Introduction

The classification of neuronal types is crucial to understanding the complex circuits of the brain. Neuronal classification requires comprehensive characterization at the levels including morphology, electrical property, transcriptomics, or a combination of them (Armañanzas et al., 2015; Gouwens et al., 2019). Among these, neuron morphology provides key implications for cellular identity and neuronal connectivity. But morphological studies have long been restricted to the soma-proximal areas, as limited by imaging technologies. Recently large-scale labeling, imaging, and reconstruction technologies have enabled the characterization of complete neuron morphologies at the whole-brain level (Winnubst et al., 2019; Peng et al., 2021). These studies showed cell-type diversity and sub-types observed in whole brain level provides new clues for neuronal circuits (Peng et al., 2021).

To untangle the complex neuron morphologies and to classify cell types, two key questions need to be addressed: feature extraction and quantitative comparison. Several feature extraction methods have been proposed, including vertex analysis (Sadler and Berry, 1983), fan-in analysis (Glaser and

McMullen, 1984), fractal analysis (Panico and Sterling, 1995), L-measure morphometrics (Guerra et al., 2011). The spatial distribution of dendritic arbors has also been demonstrated as a relevant feature (Sümbül et al., 2014). For quantitative comparison of morphological data, both supervised and unsupervised algorithms have been applied (Hosp et al., 2014; Lu et al., 2015). However, these approaches are designed for analyzing dendrites and local axons arbors and the extracted features may not be suitable for characterization of morphologies with long-range projections.

In recent years, several methods have been proposed for studying the full morphology of projecting neurons. Considering the spatial distribution of both local and distal neuronal branches, Costa et al (2014) introduced NBLAST for measuring pairwise neuronal similarity. Integrating topological branching patterns with spatial information, persistent homology was introduced to compare neuron structures and classify a large collection of neuron structures (Li et al., 2016; Kanari et al., 2017). BlastNeuron compares morphological similarities using a structural alignment approach (Wan et al., 2015). The accuracy and robustness of such approaches are considerably affected by within-type diversity and registration precision.

The long-range projection path is an biologically relevant feature for defining cell types (Peng et al., 2021), which is neglected by the abovementioned approaches. Here, we propose a novel strategy to encode the projection path, tree-like structure of 3D coordinates in the brain space, as a computable characteristic for quantitative analysis. We applied a sequentialization strategy of neuron morphology which was used for identification of structural motifs (Ascoli et al., 2015). The sequence structure provides a natural representation of the projection orders and allows for the application of a series of deep sequential models (DSM), which are widely used in the natural language processing (NLP) field. For cell-type classification tasks, we implemented and trained a Hierarchical Attention Network (Seo et al., 2016) model (DSM-HAN) and demonstrated its outstanding performance. For the measurement of cell-cell similarity, we trained a sequential autoencoder model (DSM-AE) to give each cell a concise representation, which encoded information of both projection strength and orders. We showed the usage of DSM-AE in unsupervised clustering and automated cell-type annotation of large datasets. With DSM-AE feature encodings, we built a database and provided an online service for fast retrieval of neuron morphologies and cell-type annotation (available via http://101.43.104.173:8501/).

Results

Overview of the model structure, datasets, and applications

To extract features and characterize neuronal morphologies, we constructed several deep sequential models, serving for multiple applications (Figure 1).

We built a pipeline of preprocess and feature extraction for neuron reconstruction (Figure 1A). The original input is the digital reconstruction of neuronal morphology, especially the complete neuron morphology of long-range projection. Reconstructed neurons are registered to CCF reference space (Wang et al., 2020) and each segment of the reconstruction belongs to a certain brain region according to its 3D location in the space. We applied a depth-first traversal algorithm to convert the

tree structure of the morphologically reconstructed neurons into a sequence of brain regions (see Methods), where each node in the sequence is represented as a one-hot encoding. The designed traversal order makes child branches close to their parent branches, allowing for more compact sequentialization of the local morphologies along the projection path. Here, a node is assigned to one of 316 manually-curated non-overlapping brain areas (Harris et al., 2019) with highly variable spatial proximity and functional similarities.

To reduce redundancy of the one-hot encoding vectors, we introduce the word2vec (W2V) module (see Methods) for feature compression and extraction. We applied this model to over 1000 neurons of multiple types and found that the W2V can not only convert the 316-dimension vectors into dense vectors, but also enhance the robustness of feature representation. The word2vec module provides the input for both supervised classification and unsupervised representation (Figure 1B).

The deep sequential model (DSM) consists of two downstream models, DSM-HAN and DSM-AE, serving for supervised classification and unsupervised representation separately. In the supervised module, a hierarchical attention network is adopted as DSM-HAN for cell-type classification. DSM-HAN reviews neuron morphologies in a hierarchical manner including node level, segment level, and the neuron level, providing both segment level encodings and neuron level encodings. The model shows its interpretability by encoding neighboring segments into similar vectors (see Supplementary Figure S3). We trained the DSM-HAN model with 1,282 neurons that belong to 12 cell types (CP_SNr, CP_GPe, VPM, ET_SS, IT_SS, VPL, LGd, MG, IT_VIS, IT_MO, RT, and ET_MO) from SEU morphology dataset.

In the unsupervised module, a sequential autoencoder (DSM-AE) model was trained for data exploration, including unsupervised clustering and retrieval of similar morphologies. Details of W2V model and DSM are introduced in the following sections.

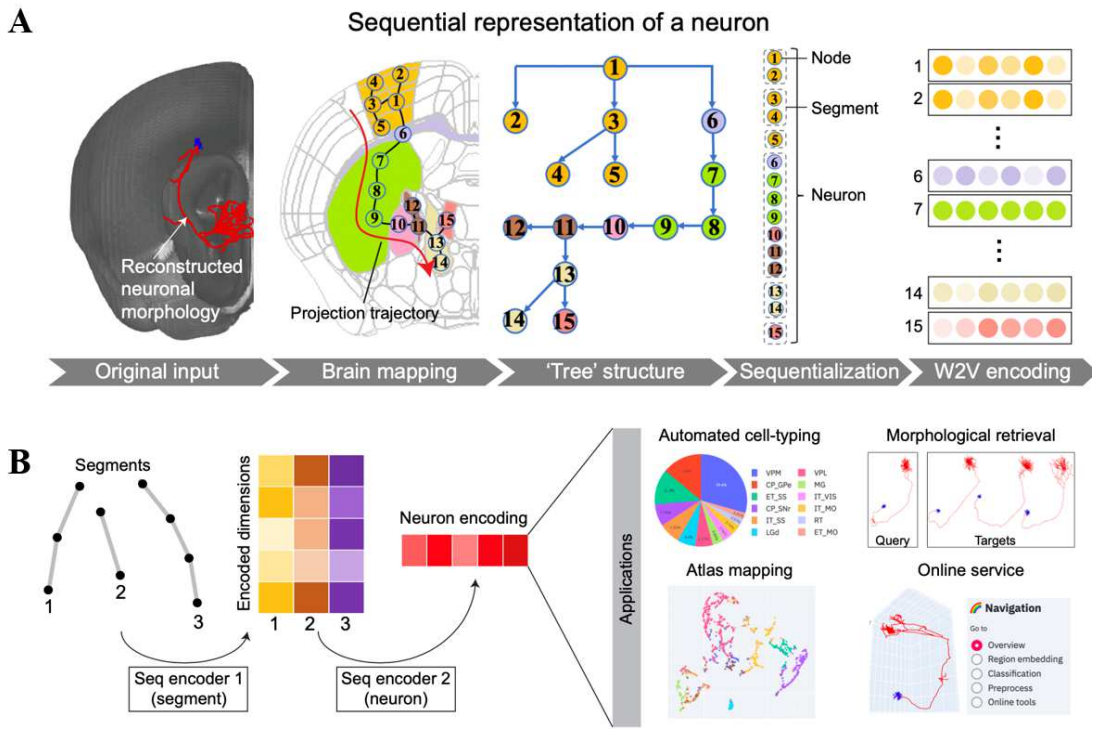

Fig. 1 | Overview of the sequential model and its applications. A. Process of raw data transformation including: mapping the nodes of reconstructed neurons to brain regions; tree-structure representation; depth-first search which disassembles the tree-structure as segment sequences; Word2vec (W2V) encoding that represents each node as the embedding of its brain region. B. Process of neuron encoding. A neuron is encoded by two steps of recursive neural network transformation (RNN). This process encodes neurons with variable sizes as a vector of the same dimensions, which is applied to tasks of automated cell-type classification, unsupervised atlas mapping, morphological retrieval.

Word2vec: distributed vectors of brain regions

Sub-regions within one major brain region, such as the somatosensory cortex, may share functional, spatial and developmental similarity. Thus, neurons projecting to these closely located sub-regions might belong to the same cell type (e.g. VPM neurons may project SSp-m or SSp-bfd). But for determining the projection regions, the registration of neuron reconstructions from different brains can also introduce certain levels of error. We addressed these concerns by introducing the word2vec (W2V) algorithm, a classic method for feature extraction and compression in NLP tasks. W2V assigns each word with a unique encoding (fixed-length vectors) indicating its semantics such that words with similar semantics or synonymous have closer encodings. By applying W2V, we were able to encode brain regions as dense vectors, the similarity of which reflects their functional and spatial similarity.

Our W2V neural network is composed of 3 layers, including an input layer, a hidden layer, and an output layer, as shown in Figure 2A. The input and output data are one-hot encodings of brain regions. Fed with a continuous sub-sequence (the neighboring nodes) as input, the model predicts the one-hot encoding of its center node. And the dense encoding of center node comes from the average hidden layer representation of input nodes (see Methods). The training task minimizes the difference between a word and its context words (see model parameters in Supplementary Table 1). To enhance the robustness of the W2V, we augmented our reconstruction dataset by introducing Gaussian noise to node coordinates (see Methods). The augmented dataset contains 5370 cells from 6 major brain regions: CNU (cerebral nuclei), CTX (cerebral cortex), TH (thalamus), MB (midbrain), HY (hypothalamus), and HB (hindbrain). After training, each brain area was encoded as a 6-dimension embedding vector (Supplementary Table 2). Dimension reduction (t-SNE) shows that the sub-regions within the same major brain regions are clustered together (Figure 2B).

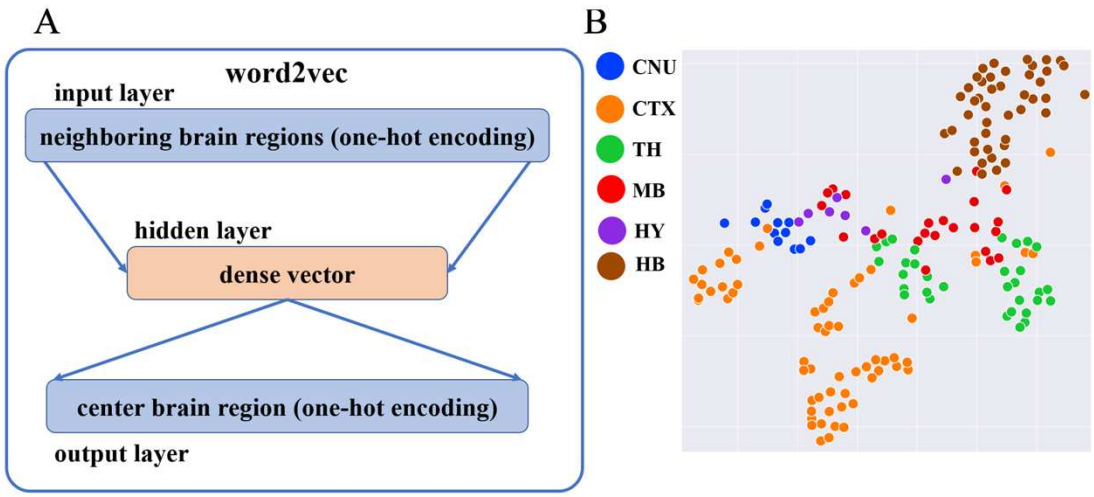

Fig. 2 | A. Word2vec network structure. B. Distribution of trained brain regions. To display the training result, sub-regions are separated into six major regions: CNU, CTX, TH, MB, HY, HB. Structures belonging to the same major region are decorated in the same color.

Hierarchical attention network: supervised cell-type classification

For supervised morphological classification, we adopted hierarchical attention network as DSM-HAN model, which is originally used for document classification. The intuition is to treat a neuron sequence (a sequence of W2V encoded brain regions) as a document (a sequence of words), and treat neuron segments (sub-sequences) as sentences. The DSM-HAN architecture reviews the sequence and extract features following the order of nodes to segments and segments to neurons.

W2V encodings of nodes in the same segment are seen as independent sub-sequence, representing local morphologies. The node level network (GRU (gated recurrent unit) layers and attention layers) takes these word2vec encodings as input, integrating the information within the sub-sequence. Its output, the node-wise averaged attention encodings, referred to as segment encodings, are passed to the segment level network. A similar structure was implemented in segment level network, which integrates segment level encodings. At the whole neuron level, a fully connected layer is implemented to output the probability estimation of cell types (Figure 3A, see Methods and Supplementary Table 1).

The DSM-HAN model was trained and tested on a dataset with 1,282 neuron cells of 12 manually defined morphological types based on their soma locations and projecting brain structures (Supplementary Table 3). We used 80% of the data (1,025 cells) for training and the remaining 20% (257 cells) for testing. Through 30 independent training sessions (random sampling of training and testing data), the model achieved 92.76% average testing accuracy. The high class-wise AUC (areas under the curve) scores of the Receiver Operating Characteristic (ROC) suggest high robustness (Figure 3B).

The information of the input neuron sequences is encoded from two aspects: the residing brain regions of nodes and their sequential orders. We performed tests to examine whether DSM-HAN

was able to learn sequential information, which was independent from brain regions. At the segment level, we performed hierarchical clustering for the segment encodings and identified the top 2 clusters for each cell (Supplementary Figure S3). Segments with close sequential orders were clustered together regardless of the residing brain regions, even for neurons with multiple target regions (e.g., ET_MO neurons). At the neuron level, we shuffled sequence orders for each cell and performed clustering with the original sequences (Supplementary Figure S4). For most cell-types, the shuffled sequences form distinct clusters, with the exception of IT neurons where the segments' residing regions are highly invariable. These results indicate that DSM-HAN utilizes the sequential information of projection orders, which further influence the neuron encodings.

We compared the model performance with other algorithms including TFIDF, NBLAST and persistent homology. TFIDF is a classic sequential model widely used for feature extraction in document classification (Salton and Yu, 1975). TFIDF was applied to extract features from neuron sequences, making the dataset a 'number of cells' by 'number of unique brain regions' matrix (see Methods). We divided training and testing datasets by the same strategy as for DSM-HAN and trained linear-kernel SVM models through 30 independent training sessions. The average testing accuracy is 89.02% (Figure 3C). NBLAST calculates the spatial proximity between neuronal segments through structural alignment (Costa et al, 2014). We used NBLAST similarity matrix as input features for several machine learning algorithms, among which linear-kernel SVM showed the best testing accuracy (70.89%, Figure 3C). The 'persistent homology' analysis couples morphology properties and neuron branching patterns (Li et al., 2016). The geodesic distance along neuron tree edges is chosen as the decision function to convert neuron structures to persistent diagrams. Wasserstein distance is calculated between the diagrams for their similarity scores, which are used as input features for machine learning classifiers. The best testing accuracy comes from linear-kernel SVM (64.82%, Figure 3C). In addition, the four approaches are applied on a smaller dataset (405 cells) of five classes, which have highly distinct morphologies and projection paths, including CP, VPM, LGd, MG, and SSp-L5. The average testing accuracy for each approach is DSM-HAN, 96.91%; TFIDF, 93.51%; NBLAST, 80.07%; persistent homology, 79.57%.

To test the robustness of the DSM-HAN model to the registration deviation, we introduced Gaussian noise to the node coordinates of the testing dataset (257 cells). This perturbation resulted in changes to the brain region assignment of many nodes. The noise level (the standard deviation of Gaussian distribution) was gradually increased from 10μm to 1000μm, with a 10μm step size (Figure 3D). Results show that the testing accuracy was stably high (>90%) with noise level <200 um. Empirically, the maximum registration deviation for mouse brains is 100μm. Thus, the DSM-HAN model is robust to variations introduced by registration and brain region assignment.

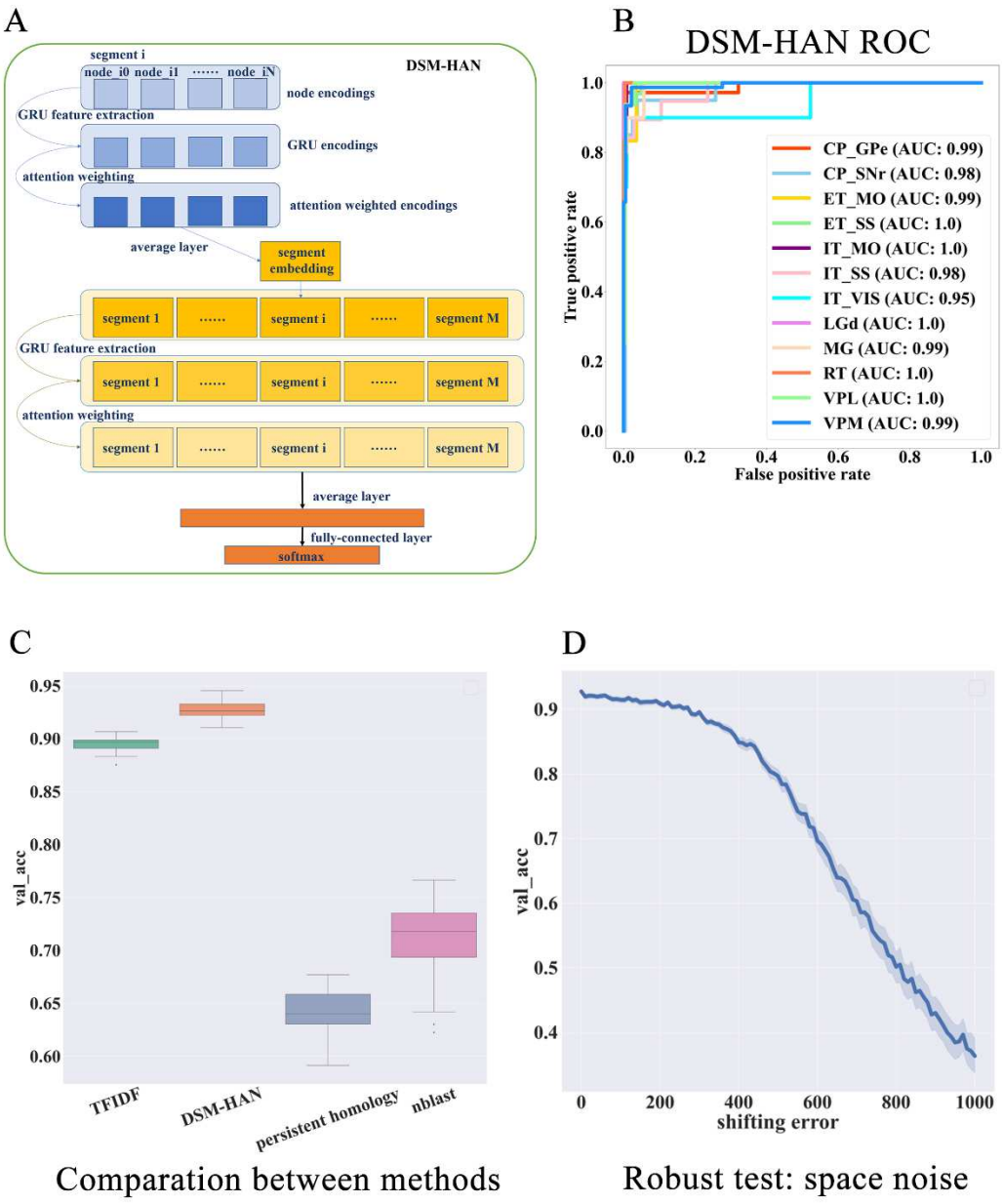

Fig. 3 | Hierarchical attention network. A. Hierarchical attention network structure. B. ROC curve and AUC of the DSM-HAN classifier. Taking multi-class classification as twelve binary classifications, we calculate false positive rate and true positive rate for each binary classification and plot the ROC curves. C. Comparison between methods. Each method was tested by 30 times of cross-validation. D. Robustness test: the accuracy curve for noise levels ranging from 10μm to 1,000μm.

Autoencoder: concise representation for exploratory studies and data retrieval

The direct cell-cell comparison is important for exploratory studies and data retrieval. As there is no exact node-to-node correspondence between two different neuron reconstructions, it becomes necessary to generate comparable and quantitative representations for them. Here, we adopted a sequential autoencoder as DSM-AE model. This model aims to learn a 32-dimension latent vector representing input neuron sequence, which can be used to well recover the input sequence (see

Methods). The training process optimizes parameters of the model to minimize the difference of recovery and input sequence. Thus, the latent vector representation significantly reduces the data dimensionality while retaining most of the information.

For applicability evaluation, we applied the model to the 1,282-cell dataset of 12 cell types (see Methods and materials) and generated their DSM-AE representations. We clustered these representations by DBSCAN and compared the correspondence between clusters and cell types (confusion matrix shown in Figure 4B, rand index=0.7116). 2D visualization of the DSM-AE encoding shows that cells from the twelve different classes form distinct sub-populations and the distribution of the clusters also match those sub-populations (Figure 4C).

For exploratory studies where the number of cell-types are unknown and where novel cell-types may exist, DSM-AE can be combined with DSM-HAN for automated dataset annotation. For identification of novel cell-types, we introduced an outlier detection module with 12 outlier detectors. For each known cell-type, a detector is a One-Class SVM model using DSM-AE encodings as input data (Methods). After a cell is assigned to a cell-type by DSM-HAN, the corresponding detector is applied, and changes its label to 'unknown' once the cell is determined as an outlier. As a proof of concept, we applied this approach to the annotation of the Janelia dataset (1,002 neuron cells, Winnubst et al., 2019), 61% cells of which belong to novel cell types (Supplementary Table 4). We evaluated the outlier detection performance by comparing its predictions with the manually-curated labeling. The median of F1-score was 0.54 (VPM=0.71; VPL=0.33; IT_MO=0.85; ET_MO=0.72; IT_VIS=0.55; ET_SS=0.07; IT_SS=0.38). We examined poorly performing detectors and found that most of the cells were assigned to cell types with similar morphology (e.g. VPM assigned as VPL, ET_MO assigned as ET_SS).

The above-mentioned deep-learning models, a reference dataset with DSM-AE representation and morphological retrieval service have been available via our website http://101.43.104.173:8501/. The DSM-AE representations are further reduced to 2D using UMAP algorithm, resulting in a 2D reference atlas. User-uploaded morphological data (query data) are projected to the 2D reference atlas for determining cell types according to the similarity with reference data points. In addition, the most similar single cells (target data) and their horizontal/vertical views are reported, for visual inspection (Figure 4D). The website also provides a series of tools including interactive visualizations of word2vec brain region encodings, 2D reference data atlas, and 3D single-cell morphologies.

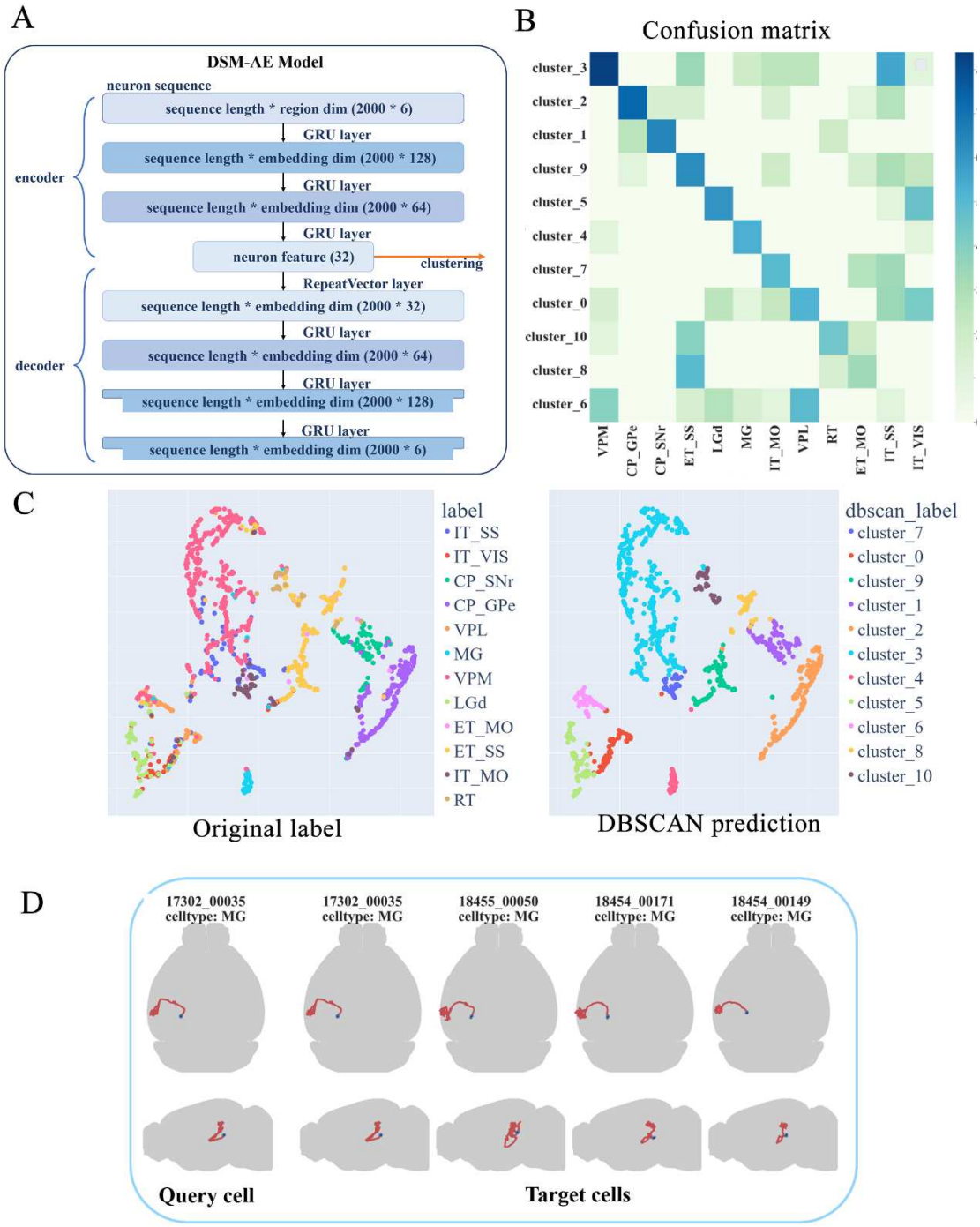

Fig. 4 | Autoencoder and cell type prediction. A. DSM-AE network structure. B. Confusion matrix of clusters and cell-types. The color indicates the 2-based logarithm of cell numbers. C. Data visualization of the DSM-AE encoding. The neuron sequences are encoded as 32-dimension vectors and projected to 2D space by the UMAP algorithm. Colors indicate cell types (left) or the DBSCAN cluster assignment (right). Only cells above DBSCAN confidence threshold are shown. D. Morphological data retrieval. The brain-level views of query cell and target cells are reported by the online service.

Discussion

Complete morphology of long-range projecting neurons is crucial for deciphering the diversity of cell types and for understanding organizational principles of brain connectivity. The unique and complex nature of morphological ~~data~~ structure brings on difficulties for data analysis. In this work, we introduce a sequential model for feature extraction from full neuronal morphology, which enables efficient cell-type classification, morphological clustering, and data retrieval. We provide a series of computational tools which outperform existing ones in accuracy and robustness. These tools are available as online services for researchers with or without computational backgrounds.

Compared with the transcriptome data based on gene sequencing, it is challenging to define morphological features that effectively represent cell-type related characteristics. Although some feature sets (e.g. L-measure) are more influential, there is limited consensus on a reasonable practice of feature definition. In the previous studies, we defined cell types by where their projection initiates and terminates (Peng et al., 2021). Such projections form directional trajectories in the brain, which turned out to be highly consistent within a cell type. The data structure of such trajectories shares a lot of similarities with texts, the basic structure of which is also stretches of nodes. For a paragraph of text, a node is a word. For the morphology of a neuron, a node is a point along its branch. Such stretches are logically (for texts) or physically (for neurons) connected, forming branching structures. In the field of natural language processing, a series of classical techniques enable tasks including texts understanding and generation. The similarity of data structure renders NLP techniques applicable for the study of neuronal projection paths.

Although our sequential representation and deep learning models showed high performance in the morphological classification tasks, several limitations and future improvements are worth noting. First, the method is specifically designed for long-range projecting neurons. It's not applicable for the classification of neurons without long-range projections (e.g. most interneurons). Second, our model neglects some local morphological features (e.g. segment curvature, branching angle, etc.), which also potentially bear information of cell identity.

There are several future directions related to our model. First, motif analysis is a classical approach in the study of DNA sequences. It is interesting to identify repetitive and characteristic sub-structures of a cell type which might be related to both neuronal identity and development history. Second, the deep sequential model enables the possibility of cross-species comparison. A metaphor for this is comparing articles written in multiple languages. The word2vec model enables identification of paralogs of brain regions and the projection path analysis enables comparison of cell-type paralogs. This will be made possible with the availability of full morphology data from other species, the most possible one being the monkey in the future.

Methods and materials

Transform neuron topological structure to sequence

Tree-like structure can be transformed into sequence through tree traversal algorithms. Here, to make the local morphologies a continuous sub-sequence and unfold these local morphologies along the neuron projection (from soma to distal arbor), we apply a depth-first traversal algorithm with designed traversal order.

While the soma node is the only node that could have more than two child nodes, to avoid multifurcations, the soma node is discarded from the neuronal structure in advance, and its removal converts the neuron structure into several binary trees. Afterwards, we apply depth-first traversal strategy on each tree to generate the corresponded sequence and concatenate them into one sequence representing neuron structure. This depth-first search strategy keeps adjacent connected nodes in the tree structure as close as possible in the sequence representation. The rules for traversal were set as follows: sub-trees with fewer leaf nodes and shorter path are visited first. Upon iterative traversal through the tree nodes, we append each visited node to the sequence. Finally, instead of using a series of spatial coordinates directly, we encode each node with its brain region in the Common Coordinate Framework (CCF; Wang et al., 2020) 3D reference space (resolution: 25μm).

Preparation and pre-processing of dataset

In this study, we have three deep models to train, word2vec model (training brain regions to dense vectors), hierarchical attention network (supervised cell type classification), and autoencoder model (direct morphology representation for unsupervised clustering analysis). All reconstructed neurons used in these tasks have been published in our previous work (Peng et al., 2021) and neuron cells were registered to the CCF 3D reference space in advance.

For training brain regions, the original dataset with 1074 cells is augmented by adding four types of spatial Gaussian noise (mean= 0μm, standard deviation= 5 μm, 10 μm, 15 μm, 20 μm) respectively. After the augmentation, the total dataset is composed of 5370 neurons.

Classification and clustering tasks are conducted on a dataset of 1282 neuron cells with labels, including twelve classes determined by their projection path in the CCF. Classes consist of CP_SNr (caudate putamen, 100 cells), CP_GPe (caudate putamen, 180 cells), VPM (ventral posteromedial nucleus of the thalamus, 378 cells), ET_SS (159 cells), IT_SS (97 cells), VPL (80 cells), LGd (dorsal part of the lateral geniculate complex, 78 cells), MG (medial geniculate complex, 50 cells), and IT_VIS (48 cells), IT_MO (48 cells), RT (33 cells), ET_MO (31 cells). To train a more robust autoencoder model, we augment the dataset by shifting spatial coordinates to reduce the class-imbalance.

To standardize the neuron reconstructions, we performed two preprocessing steps: small neuron segments which length is less than 10 μm, are removed and the distance between adjacent connected nodes is readjusted to 20 μm (see Code availability for the first step and the second step is conducted by ‘resample swc’, a Vaa3d built-in plugin (Peng et al., 2010)).

Train brain regions to dense vectors

To utilize brain region in neuron cell type analysis effectively, a proper method is needed to vectorize regions. One-hot encoding encodes categorical features by creating a binary column for each category. However, one-hot encoding not only fails to indicate the relationship of encoded vectors but also triggers dimension explosion if there are too many categories in a dataset.

In natural language process tasks, word2vec is commonly used to train word embeddings from one-hot encoding to dense vectors in a corpus (Mikolov et al., 2013). The intuitive idea behind word2vec is to gather similar words and disperse irrelevant words in their latent vector space. Since one-hot encodings of words are similar to brain regions in our case, it is easy to extend this method to train brain region encodings.

There are multiple model architectures to train W2V encodings. As illustrated in Figure 2A, we select continuous bag-of-words (CBOW) as the word2vec architecture. Initially, each word is encoded as 1-of-V vector using one-hot encoding, where V is the size of the vocabulary (all words in the corpus). Instead of feeding a whole sequence as input, we see each word (center word) and its neighboring words (context words, size = N) as an input-output pair, transferring a complete sequence into sub-sequences. The input layer projects context words (N*V data matrix) to the linear hidden layer (D-dimension latent space), using a weight matrix W (V*D, shared by N words). At the hidden layer, the N words embeddings are averaged as the center word representation (D-dimension, dense vector), which is also used to predict 1-of-V vector of center word. Given a one-hot encoding of input word, the W2V model gives its dense vector by multiplying matrix W.

The word2vec model is available in an open-source python library, Gensim (Rehurek and Sojka, 2010) for the training of vector embedding. In this study, we fed neuron sequences to model as input dataset, and the training was automatically finished by Gensim APIs.

TFIDF for morphology classification

In the field of natural language process, the basic form of data is the sequence. By transforming neuron reconstruction data to sequences, we can extend numerous methods from the NLP area to cell type classification task.

The TFIDF value is product of term frequency and inverse document frequency for a word. Term frequency is the number of times that a term occurs in a document, indicating the importance of the term to the document. And inverse document frequency is defined to measure how unique the word is to the document in documents, calculated by dividing the total number of documents by the number of documents containing the word and then taking the logarithm of that quotient.
To perform feature extraction, documents are converted to a matrix of TFIDF features (number of documents * number of words), and then machine learning algorithms such as SVM can be performed on the TFIDF matrix. The above feature extraction and classification are implemented using scikit-learn (Pedregosa et al., 2012) in this study.

HAN for morphology classification

Besides the traditional technique, we seek higher and more robust performance in classification.

Hierarchical attention network (HAN; Seo et al., 2016) is a supervised classification algorithm to classify documents in the NLP area. Unlike traditional document classification treating the whole document as one continuous word sequence, HAN focuses on the structure of the document and builds representations of sentences that are then aggregated into a document representation.

The intuition of HAN is not classifying document with isolated words, but rather relate the task with the interaction of words. By transplanting the idea to neuron sequence, we introduce the DSM-HAN model, which decomposes a neuron sequence (a document) into segments (sentences) according to the topology of neuron structure, and treats each segment as a sub-sequence, representing local morphologies.

The model architecture is summarized in Figure 3A. It consists of three parts: word level network (blue boxes), sentence level network (yellow boxes), and classification network (orange boxes). Word level network and sentence level network have basically the same architecture including a word sequence encoder (GRU, gated recurrent unit), a word attention layer, and an average layer. Classification layer consists of several fully-connected layers, outputting the probability estimation of cell types.

GRU, short for a gated recurrent unit, is used to embed sequence data. The attention layer is used to evaluate the importance of each state in a sequence and add weight to them. The GRU (Cho et al., 2014), a gated recurrent unit, belongs to the RNN (recurrent neural network) family, with the ability to encode information from sequence data. Compared with traditional RNN, the GRU has two types of gates: reset gate r_j and update gate z_j, which are used to and help with gradient vanishing in RNN related training. At the j-th state of a sequence, the current hidden state h_j is computed by

$$h_j = (1 - z_j) \odot h_{j-1} + z_j \odot \widetilde{h_j},$$

where update gate z_j decides the proportion of current candidate hidden state $\widetilde{h_j}$ to previous hidden state h_{j-1} in current hidden state. The $\widetilde{h_j}$ is computed as:

$$\widetilde{h_j} = \tanh\left(W_h x_j + r_j \odot (U_h h_{j-1}) + b_h\right),$$

where reset gate r_j balances the contribution between current input information and previous state information to candidate hidden state h_j.

The gates r_j and z_j are computed by

$$r_j = sigmoid(W_r x_j + U_r h_{j-1} + b_r),$$
$$z_j = sigmoid(W_z x_j + U_z h_{j-1} + b_z).$$

The attention layer, origin from the attention mechanism, gives prominence to important states in a sequence by assigning higher weights to them. The mechanism means to filter out fewer relative words and make contributing words dominate in tasks. For the current state h_j, we have

$$u_j = tanh\left(W_w h_j + b_w\right),$$
$$\alpha_j = \frac{\exp\left(u_j^T u_w\right)}{\sum_j \exp\left(u_j^T u_w\right)},$$
$$s = \sum_t \exp\left(\alpha_j h_j\right)$$

where u_j is the embedding of each state h by one-layer MLP, denoting the importance of each state, and α_j is the normalized weight of the state h_j. The s is the weighted sequence embedding as the final output. And the subscript W, U, and b refer to trainable weights.

In this task, the DSM-HAN model is realized under TensorFlow (Abadi et al., 2015) framework (see Code availability for implementation).

Persistent homology and NBLAST

The persistent homology framework (Li et al., 2016; Kanari et al., 2017) integrates both topological features and spatial features, which can extract information of interest using a series of descriptor functions, providing a flexible framework to vectorize the neuron morphologies. NBLAST (Costa et al, 2014) provides a direct cell-cell comparison method by measuring pairwise neuronal similarity. To calculate persistent homology matrix, we accessed open-source code of the persistent homology algorithm in its GitHub repository (https://github.com/Nevermore520/NeuronTools). Taking 1282 cells as input, we preprocess each neuron reconstruction into segments with descriptor function values, using 'GeodesicFileTransfer' (in /Java/src/). Then the result is used to calculate Persistent Diagrams representing neuron morphologies, using codes in CPP/src/NTmain.cpp. Finally, taking these Persistent Diagrams as input, we can compute Persistent Distance Matrix, using codes in Wasserstein/wasserstein/.

To calculate NBLAST similarity matrix, we utilized the NBLAST R package (version: 1.6.5). Using 'nblast' function (version: 2; other parameters: default), we calculate the similarity between cells within the 1282 cells, generating a similarity matrix (1282 * 1282).

Through above two methods, we can calculate similarity scores between neurons, making the neuron dataset a 'number of cells' by 'number of cells' matrix filled by these scores. Finally, we train a machine learning classifier, SVM, for neuronal morphology classification, using the two previous similarity matrixes as input. The SVM algorithm is implemented by a python library, scikit-learn (version: 1.0.2, parameters: default).

Clustering and cell retrieval

Traditionally, neuron morphologies are encoded by selecting features that are highly expert-dependent and suffer from information loss inevitably. Here, we introduce the autoencoder model, which is an unsupervised neural network built for representation learning. The model aims to learn a latent representation for input neuron, and reconstruct original morphology reconstruction from the representation.

The autoencoder architecture is summarized in Figure 4A. The model has two parts: encoder and decoder. The encoder part takes a neuron sequence as input, and encode it into a reduced dimension vector. And decoder is responsible for reconstructing original input sequence from the reduced vector. Both encoder and decoder are composed of GRU layers, and they are connected by RepeatVector (RV) layer. The RV layer simply repeats its input data, and we use it here to recover the sequence length. Then we cluster these cells with their 32-dimension feature by the DBSCAN clustering algorithm. In detail, the noise recognized by DBSCAN is removed firstly, and the rest are seen as effective data. In the cell recommendation part, we convert the input cell sequence into a 32-dimension vector by the autoencoder model and use the Euclidean distance to recommend the top-4 cells similar to the input cell.

The DSM-AE is realized under TensorFlow (Abadi et al., 2015) framework (see Code availability for implementation).

Outlier detection and automated dataset annotation

For unknown cell types identification, we introduce the outlier detection module, combining both DSM-AE model and DSM-HAN model. Our previous result shows DSM-HAN can provide a reliable cell-type identification for input cells within the twelve projections, but it is still hard to classify cells beyond the twelve classes. So, we train 12 extra One-Class SVMs as outlier detectors to correct the DSM-HAN predictions, changing it to 'unknown types' accordingly. Taking the DSM-AE embeddings as training dataset, we tarin these One-Class SVMs, predicting whether it is inside-class cell or outside-class cell for each cell type. The One-Class SVMs are also implemented by scikit-learn (version: 1.0.2, parameters: default).

Acknowledgement

H.P. and P.X. conceptualized and supervised the project. F.X. implemented word2vec and deep neural networks. F.X., P.X. and Z.Z. performed data analysis. P.X., and F.X. wrote the manuscript along with help and feedback from H.P. and coauthors. Y.L., F.X. and S.Z. double-checked and curated the projection types of 1,002 cells in the Janelia MouseLight dataset. L.L. led a team to generate the SEU morphology dataset. This study was mainly supported by several grants awarded to H.P. at Institute for Brain and Intelligence, Southeast University. This work was also partially supported by a National Science Foundation of China (NSFC) grant (U20A6005). P.X. was partially supported by NSFC grant 32100529. L.L. was partially supported by a grant from Tencent Inc (8550270010). We also thank Giorgio Ascoli for discussion and comments on the manuscript, and various members in the BICCN Morphology Workgroup for feedback.

Conflict of Interest: none declared.

Code availability

The code and related data can be accessible from the following GitHub respiratory https://github.com/xiongfengNJ/neuron2seq. A website for neuronal data retrieval based on morphological similarity is available via http://139.155.28.154:8501/ (backup web service: http://101.43.104.173:8501/).

Supplementary Figures

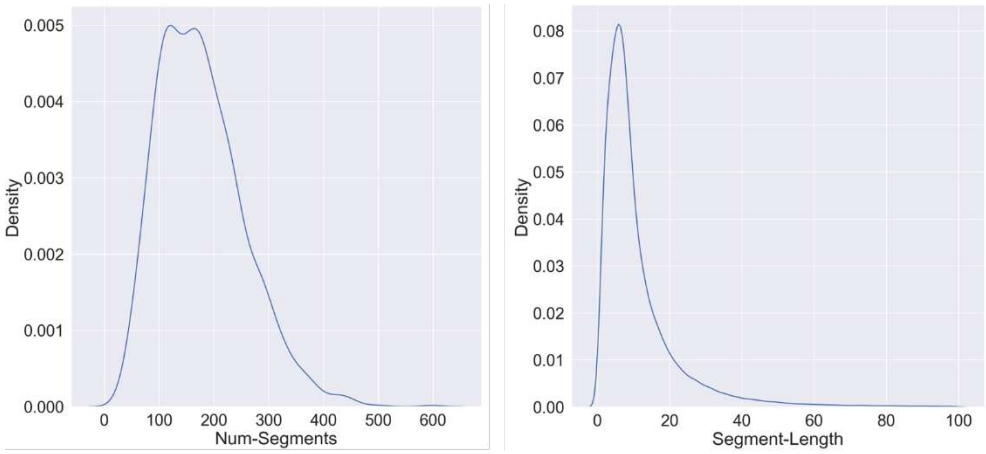

Figure S1

To train a hierarchical attention network, the neuron sequence is split into segments by termination nodes. We count the nodes number in each segment across the whole dataset (1282 neuron cells) and make a distribution of it. As we can see from figure S1, most segments have nodes numbers within 20, and most neuron sequences have segment numbers within 300. To reduce the training cost and cover most nodes in SWC files, we regulate all sequences to 300 segments and 20 nodes (each segment) by truncation and padding.

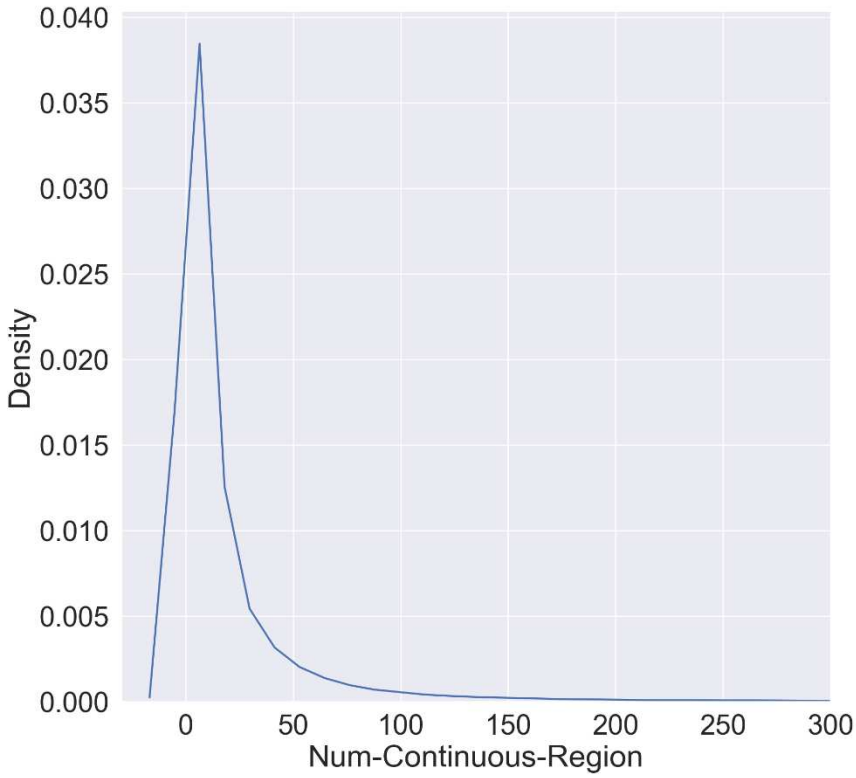

Figure S2

To train a word2vec model, we need to set the window size for each center word. We count the number of nodes in continuous a sub-sequence of an identical region. Illustrated by figure S2, most sub-sequences of an identical region have the number of nodes within 50. As the result, the window size is set to 50.

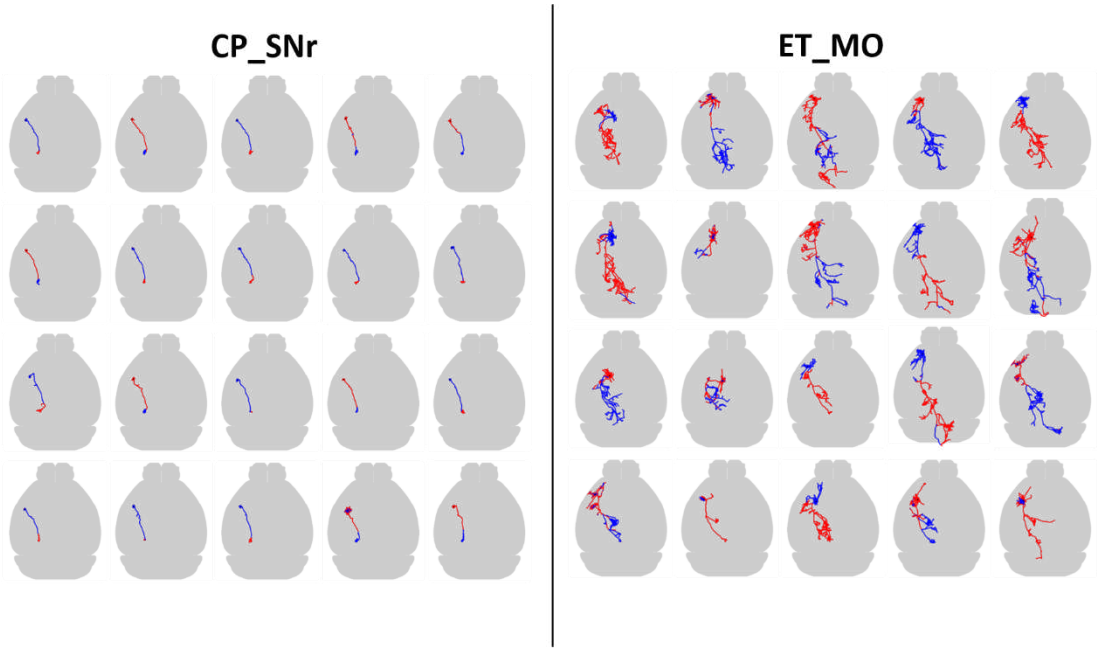

Figure S3

Hierarchical clustering was performed for the HAN segment-level encodings for each cell, resulting in two clusters of segments corresponding to the first division of the hierarchical tree. Here we display the top-views of 40 neurons from two cell types, CP_SNr and ET_MO, and segments from same cluster are decorated by same color.

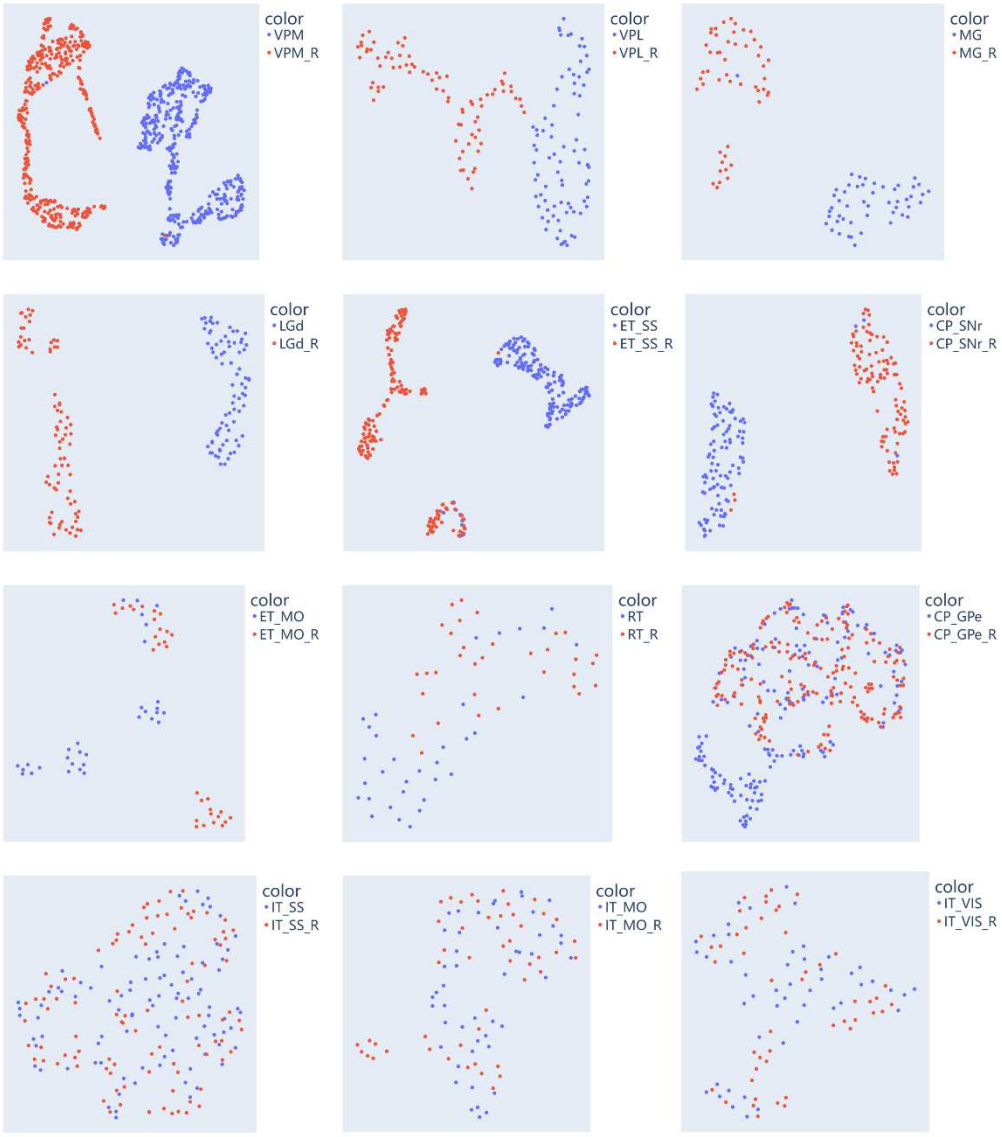

Figure S4

Clustering of HAN neuronal encodings for cells with or without sequence shuffling. Each dot represents a neuron and each plot shows the results of UMAP dimension reduction. Here, we tested all the 1,282 cells from twelve cell types (from the SEU dataset) and colored original (blue) and shuffled (red) cells.

Supplementary Tables

Supplementary Table 1. Parameters used in hierarchical attention network, word2vec model, autoencoder model, and NBLAST. We also provided source codes which contains these parameters in the GitHub.

Supplementary Table 2. Brain region vectors encoded by word2vec model. Each brain region is assigned with a six-dimension vector. And these brain region features can be reused in other studies.

Supplementary Table 3. Dataset details. Our training datasets come from SEU neuron reconstruction. According to the dataset composition used in hierarchical attention network, word2vec model, and

autoencoder model, each cell is marked as Y or N.

Supplementary Table 4. Summary of the datasets, and model predictions on SEU swc dataset (n=1282) and Janelia swc dataset (n=1002).

Supplementary Table 5. Outlier detector performance. An outlier detector is trained to filter unknown cells (not belong to the 12 classes). The training is performed on SEU dataset, and the testing is on Janelia dataset.

Reference

Abadi, M. et al. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems." (2015).

Armaanzas, R. & Ascoli, G. A. "Towards the Automatic Classification of Neurons." other, 38.5(2015).

Cho, K. et al. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation." Computer Science (2014).

Costa, M. et al. "NBLAST: Rapid, sensitive comparison of neuronal structure and construction of neuron family databases." (2014).

Glaser, E. M. & Mcmullen, N. T. "The fan-in projection method for analyzing dendrite and axon systems." Journal of Neuroscience Methods 12.1(1984):37-42.

Gillette, T. A. & Ascoli, G. A. "Topological characterization of neuronal arbor morphology via sequence representation: I - motif analysis." BMC Bioinformatics 16.1(2015):216.

Gillette, T. A., Hosseini, P. & Ascoli, G. A. "Topological characterization of neuronal arbor morphology via sequence representation: II - global alignment." BMC Bioinformatics,16,1(2015-07-04) 16.1(2015):209.

Gouwens, N. W. et al. "Classification of electrophysiological and morphological neuron types in the mouse visual cortex." Nature Neuroscience 22, 1182–1195 (2019).

Guerra, L. et al. "Comparison Between Supervised and Unsupervised Classifications of Neuronal Cell Types: A Case Study." Developmental Neurobiology 71.1(2011):71-82.

Harris, J. A. et al. "Hierarchical organization of cortical and thalamic connectivity." Nature 575.7781(2019).

Hosp, J. A. et al. "Morpho-physiological criteria divide dentate gyrus interneurons into classes." Hippocampus 24.1(2014):189-203.

Kanari, L. et al. "A Topological Representation of Branching Neuronal Morphologies." Neuroinformatics 16.4(2017):3-13.

Li, Y. et al. "Metrics for comparing neuronal tree shapes based on persistent homology." Plos One 12.8(2016): e0182184.

Lu, Y. et al. "Quantitative Arbor Analytics: Unsupervised Harmonic Co-Clustering of Populations of Brain Cell Arbors Based on L-Measure." Neuroinformatics 13.1(2015):47-63.

Mihaljevi, B., Benavides-Piccione, R., Bielza, C., Defelipe, J. & Larraaga, P. "Bayesian Network Classifiers for Categorizing Cortical GABAergic Interneurons." Neuroinformatics 13.2(2014):193-208.

Mikolov, T., Chen, K., Corrado, G. & Dean, J. "Efficient Estimation of Word Representations in Vector Space." Proceedings of the International Conference on Learning Representations (ICLR 2013).

Panico, J. & Sterling, P. "Retinal neurons and vessels are not fractal but space‐filling." Journal of Comparative Neurology 361.3 (1995).

Peng, H. et al. "V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets." Nature Biotechnology 28.4(2010):348-353.

Peng, H. et al. Morphological diversity of single neurons in molecularly defined cell types. Nature 598, 174–181 (2021).

Pedregosa, Fabian et al. “Scikit-learn: Machine Learning in Python.” J. Mach. Learn. Res. 12 (2011): 2825-2830.

Ehek, Radim , and P. Sojka . "Software Framework for Topic Modelling with Large Corpora." Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks 2004.

Roberto, S. et al. "Classification of neocortical interneurons using affinity propagation." Front Neural Circuits 7(2014):185.

Sadler, M. & Berry, M. “Morphometric study of the development of Purkinje cell dendritic trees in the mouse using vertex analysis.” Journal of microscopy vol. 131, Pt 3 (1983): 341-54. https://doi.org/10.1111/j.1365-2818.1983.tb04259.x

Salton, G. & Yu, C. T. "On the construction of effective vocabularies for information retrieval." Acm Sigplan Notices 10.1(1975):48-60.

Scorcioni, R., Polavaram, S. & Ascoli, G. A. "L-Measure: a web-accessible tool for the analysis, comparison and search of digital reconstructions of neuronal morphologies." Nature Protocols

3.5(2008):866-876.

Yang, Z. , et al. "Hierarchical Attention Networks for Document Classification." Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 2016.

Sümbül, U., Song, S., McCulloch, K. et al. A genetic and computational approach to structurally classify neuronal types. Nat Commun 5, 3512 (2014). https://doi.org/10.1038/ncomms4512

Wang, Q. et al. "The Allen Mouse Brain Common Coordinate Framework: A 3D Reference Atlas - ScienceDirect." Cell (2020).

Wan, Y. et al. "BlastNeuron for Automated Comparison, Retrieval and Clustering of 3D Neuron Morphologies. " Neuroinformatics 13.4(2015):487-499.

Winnubst, J. et al. "Reconstruction of 1,000 Projection Neurons Reveals New Cell Types and Organization of Long-Range Connectivity in the Mouse Brain." Cell 179.1(2019).

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SupplementaryTable3.xlsx
- SupplementaryTable1.xlsx
- SupplementaryTable2.xlsx
- SupplementaryTable5.xlsx
- SupplementaryTable4.xlsx