Available online at www.sciencedirect.com

ScienceDirect

Current Opinion in Systems Biology

Graph representation learning for single-cell biology

Leon Hetzel[1,2], David S. Fischer[1,3], Stephan Günnemann[4,5] and Fabian J. Theis[1,2,3]

Abstract

Single-cell RNA sequencing measures gene expression at an unprecedented resolution and scale and allows the analysis of cellular phenotypes which was not possible before. In this context, graphs occur as a natural representation of the system —both as gene-centric and cell-centric. However, many advances in machine learning on graphs are not yet harnessed in models on single-cell data. Taking the inference of cell types or gene interactions as examples, graph representation learning has a wide applicability to both cell and gene graphs. Recent advances in spatial molecular profiling additionally put graph learning in the focus of attention because of the innate resemblance of spatial information to spatial graphs. We argue that graph embedding techniques have great potential for various applications across single-cell biology. Here, we discuss how graph representation learning maps to current models and concepts used in single-cell biology and formalise overlaps to developments in graph-based deep learning.

Addresses

[1]Institute of Computational Biology, Helmholtz Center Munich, Germany
[2]Department of Mathematics, Technical University of Munich, Germany
[3]TUM School of Life Sciences Weihenstephan, Technical University of Munich, Germany
[4]Department of Informatics, Technical University of Munich, Germany
[5]Munich Data Science Institute, Technical University of Munich, Germany

Corresponding author: Theis, Fabian J. (fabian.theis@helmholtz-muenchen.de)

Current Opinion in Systems Biology 2021, **28**:100347

This review comes from a themed issue on **Theoretical approaches to analyze single-cell data (2021)**

Edited by **Olivier Gandrillon** and **Michael Stumpf**

For complete overview of the section, please refer the article collection - Theoretical approaches to analyze single-cell data (2021)

Available online 23 May 2021

https://doi.org/10.1016/j.coisb.2021.05.008

Keywords

Graphs, Single cell, Deep learning, Graph representation learning.

Introduction

Experimental protocols for molecular profiling of single cells from dissociated tissues have drastically advanced in the recent past [1]. Importantly, both the number of observations and the feature resolution are complex enough for machine learning methods to be crucial to understand the underlying cellular phenomena [2]. Here, observations are cells and features correspond to gene expression measured on the level of RNA or protein. In addition, spatial molecular profiling techniques open the way to analyse cellular identities as a product of their spatial context, transitioning away from the dissociated setting. Graphs provide a rich modelling abstraction both for *in situ* and for dissociated single-cell data.

As graph representation learning has become a fast growing and popular research field, several works exist that summarise the field from a computer science perspective [3–5]. Just recently Muzio et al. [6] also presented a review on deep learning, and in particular graph neural networks (GNNs), for network biology. Although the methods are similar, we want to give a perspective on graph representation learning that is specifically motivated from the applications and challenges of single-cell biology as well as recent advances in spatial molecular profiling. For this, we first discuss how to construct graphs from single-cell data and present applications that build on top of these. In a subsequent step, we introduce graph representation learning, both unsupervised and supervised. We discuss current applications of this framework in single-cell biology and give a perspective of what can be expected in future work.

Construction of graphs from single-cell data

Single-cell transcriptomics data are captured in a matrix, where rows refer to cells and columns to genes. Hence, we can identify either the cells or the genes as entities of interest and ask for their interactions. This is equivalent to constructing cell or gene graphs based on adequate distance measures. For cells, the gene vectors can readily be identified as molecular states. For genes, the interpretation of edges depends a lot on the construction, varying from statistical dependence in co-expression networks to causal regulation in gene regulatory networks (GRNs).

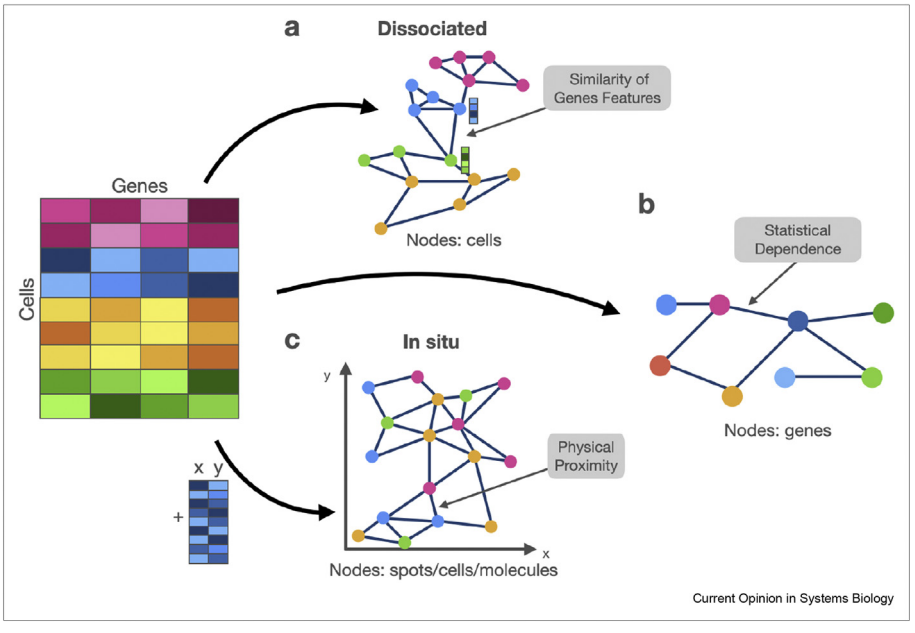
Here, we describe the construction of such molecular similarity graphs, gene graphs and spatial graphs, see Figure 1, as well as applications building on these.

An essential task in single-cell analysis is to project gene expression data to a low-dimensional space which arises from the assumption that the true data generative process is that of gene regulation. Put differently, this means to approximate the manifold of cell identities whose underlying structure is that of a non-Euclidean space. As graphs $G(V, E)$ represent complex, non-Euclidean structures, a natural choice for cells (nodes $v_i \in V$ in G) is to build the graph of k nearest neighbours (kNN) (Figure 1a) [7,8]. In this case, nearest is quantified by a similarity metric, usually the Euclidean metric in (possibly downprojected) gene space. As a consequence, the edges $e_{ij} \in E$ in the kNN graph connect pairs of cells, v_i and v_j, that are molecularly similar. It is common to encode any such graph structure $G(V, E)$ by the adjacency matrix A whose entries A_{ij} are 1 if $e_{ij} \in E$ and 0 otherwise. For molecular similarity graphs, this binary encoding readily extends to weighted graphs, where the similarity between nodes is additionally encoded as edge weights, that is $A_{ij} > 0$.

The orthogonal view on single-cell data is to consider the genes as entities of interest, that is as nodes V of the graph. In this view, ultimate interest lies in the regulatory links between genes. Unlike molecular similarity graphs, GRNs are not the entry point of an analysis but the product of such. In GRNs, directed edges E encode the causal relations between pairs of genes, that is their upregulation and downregulation. For directed graphs, the adjacency matrix A is not symmetric anymore as $e_{ij} \in E$ does not imply $e_{ji} \in E$ like previously, where the graph was assumed to be undirected. As this perspective mainly focuses on representation learning on graph-structured data instead of graph inference itself, please refer to Pratapa et al. [9] and Chen and Mar [10] who give comprehensive overviews over the field of gene network inference. Neglecting the causal structure present in regulation, co-expression networks approximate GRNs by constructing an undirected graph based on statistical dependence. More specifically, they are built on the pairwise correlation of genes. In the easiest case, the Pearson correlation is binarised by a threshold to indicate connections between the considered gene pairs (Figure 1b). As a result, the problem of gene regulation is shifted to the question how to prune the resulting network such that only true regulatory relations remain [11].

Complementary to single-cell transcriptomics data and the resulting cell and gene graphs, recent advances in spatial molecular profiling [1,12,13] made it possible to access gene expression data alongside its spatial information. Spatial graphs are the natural way to encode this additional spatial modality, and they are similar to molecular similarity graphs in the sense that nodes represent

Figure 1



Going from (left) single-cell gene expression data presented as a count matrix to (a) a cell graph, (b) a gene graph and (c) a spatial graph. **(a)** The molecular similarity graph which encodes cells as nodes and similarity in gene expression as edges. **(b)** A gene co-expression network that can be deduced from the correlation of two genes. Nodes are genes, and edges are binarised correlation values. **(c)** If additionally spatial information is available (blue matrix), a spatial graph can be constructed which resembles real coordinates in the tissue. In this example, the nodes are identical to (*a*). The resolution does not have to be cell-level but can range from being subcellular to averages over cells (spots), dependent on the available data.

cells; however, the edges relate to physical proximity (Figure 1c). The neighbourhood $N(i) = \{v_j | e_{ij} \in E\}$ of individual cells v_i allows to explicitly account for extrinsic effects on cell states, resulting, for example, from cell-cell communication.

To understand the biological heterogeneity from scRNA-seq data, the cellular manifold is either considered to be discrete or continuous. The discrete setting typically applies to clustering-related tasks such as the identification of distinct cell types or identities. From a graph theoretic point of view, this means to find communities, cell types or modules of genes that are densely connected within but only sparsely connected between communities, for example, by maximising the modularity of the graph [14]. To visualise the cellular manifold and the identified communities, it is standard practice to use nonlinear dimensionality reduction methods, such as t-distributed stochastic neighbour embedding [15] and uniform manifold approximation and projection (UMAP) [16]. These are both graph-based and therefore result in different embeddings compared to linear dimensionality reduction methods such as principle component analysis (PCA) [2].

The continuous setting, in contrast, regards single-cell data as a snapshot of a dynamic process and is interested in the underlying dynamics, such as the identification of cell lineages in diseased and healthy tissue [17]. A central concept for this is random walks on graphs which can be formalised by the transition matrix $T = AD^{-1}$. The degree matrix $D = \text{diag}\left(\sum_j A_{ij}\right)$ normalises the adjacency matrix to get proper transition probabilities T_{ij} between nodes v_i and v_j, and multiple steps of a random walk can be expressed as powers n of the transition matrix T^n. Methods in this context are usually referred to as trajectory inference or pseudotime analysis, and extensions of the standard random walk lead to other related stochastic processes on graphs such as diffusion maps [18]. Note that the described mathematical analysis equally applies to gene and spatial graphs.

Graph representation learning

When biological data are given in a graph-structured way, it is not directly obvious how to relate this to biological questions. The field of graph representation learning, both supervised and unsupervised, is able to bridge this gap and allows formulating biologically motivated learning tasks on graphs. Taking advantage of the flexibility of neural networks, biologically meaningful representations are learnt in a data-driven fashion. Here, we introduce the most important methods for graph representation learning together with their applications in the context of single-cell biology which we expect to be explored more frequently in the future.

Graph representation learning means to find a meaningful, potentially low-dimensional, representation of nodes from the complex relations present in a graph. This requires a map from each node in the graph to a vector space, which is also referred to as latent space.

In the unsupervised setting, a crucial step as a user is to identify which property of the graph is important for the biological question at hand and should be respected in the latent space. A helpful separation is that into positional and structural encodings. Positional encodings matter if the relations and positions of multiple nodes to one another, far apart or close together, are essential. This is usually the case for trajectory inference from molecular similarity graphs. In contrast, structural encodings are important if the direct neighbourhood defines the nodes' representation rather than how nodes are relatively positioned. This, for example, applies to the modelling of intercellular interactions such as ligand-receptor pairs.

If, however, labels are available, we encounter the scenario of (semi-) supervised learning in which we aim to find an embedding targeted at a specific purpose. This purpose usually refers to nodes (e.g. the prediction of cell types) or the whole graph (e.g. the classification of cancer types). Link prediction tasks, however, depend more on the positional encoding of the graph and can also be trained in an unsupervised manner by leveraging the graph topology.

Node embeddings

Node embeddings aim to find low-dimensional representations of the nodes which summarise the geometric properties of the graph. These methods are trained in an unsupervised fashion, and a particularly instructive framework is that of an encoder–decoder perspective [19]. Although the encoder f maps all nodes i to a low-dimensional Z_i, the decoder g tries to reconstruct information about the original graph structure from all the Z_i embeddings. In the simplest case, the encoder simply becomes a lookup table:

$$Z_i = f(i) \ .$$

Neglecting the complex, nonlinear relationships in the graph, this encoding is similar to dimensionality reduction methods like PCA. To map the embedding back to the original graph structure, the decoder g takes two arguments Z_i and Z_j. Cast in matrix form, this can be understood as a similarity matrix $\hat{W}$ which should be similar to another matrix W that is directly inferred form the graph structure. Formally, this becomes

$$\hat{W} = ZZ^{\mathrm{T}} \quad \text{with} \quad \hat{W}_{ij} = Z_i^{\mathrm{T}} Z_j = g(Z_i, Z_j) \ .$$

Differences between algorithms arise through differences in the comparison with the original graph

structure (Figure 2a). In the simplest case [20], which is referred to as graph factorisation, the similarity matrix is compared with the (weighted) adjacency matrix W; hence, it uses the edge structure of the graph:

$$W_{ij} - \hat{W}_{ij} \ , \quad \forall e_{ij} \in E \ .$$

Other unsupervised methods like DeepWalk or node2vec [21,22] extend the notion of similarity to paths on the graph, thus increasing the order of proximity, that is neighbours, taken into account. To optimise over paths in the graph, both methods first sample random walks $\mathcal{D}$ starting from each node v_u and then update the latent representation such that the probability of co-occurring nodes $P(v_k|v_u)$ in the random walk is maximised. In practice, this is carried out by minimising the loss

$$L = - \sum_{(u,k) \in \mathcal{D}} \log(g(Z_u, Z_k)) \ ,$$

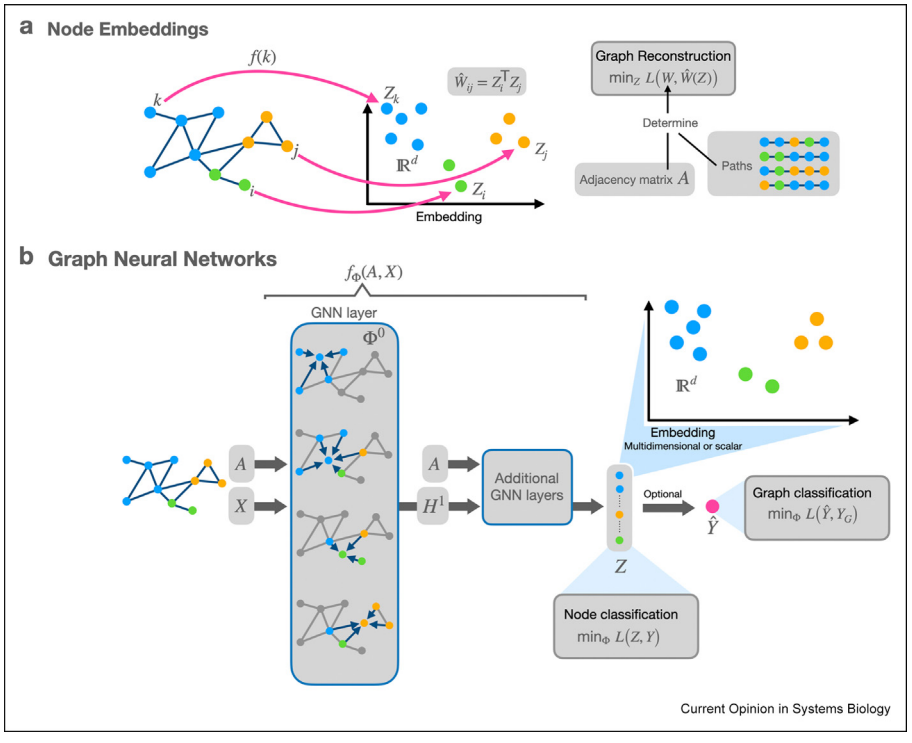
where the decoder approximates the conditional probability $P(v_k|v_u)$ through:

$$g(Z_u, Z_k) = \frac{\exp(Z_u^{\mathrm{T}} Z_k)}{\sum_{v_i \in V} \exp(Z_u^{\mathrm{T}} Z_i)} \ .$$

This method was originally introduced as the skip-gram language model [23] for the identification of contexts from words. Both models use paths of fixed length; node2vec, however, can be understood as an extension to DeepWalk as it includes different types of random walks with local (microscopic) and global (macroscopic) exploration behaviour.

This focus on the topology of the graph is similar to the described approaches for trajectory inference. Common methods such as diffusion pseudotime [24] or partition-based graph abstraction (PAGA) [25] follow this approach. Diffusion pseudotime averages over paths on a molecular similarity graph to identify cell differentiation and branching points, whereas PAGA generates coarse-grained graphs that are faithful to the global topology of the cell graph and therefore able to present cluster-like structure while also inferring continuous trajectories.

Figure 2



The two presented methods for graph representation learning: (*a*) Node embeddings and (*b*) Graph Neural Networks. (***a***) Nodes are mapped to a low dimensional space in which their representation should resemble a graph property W which can be computed from the adjacency matrix or paths on the graphs. The prediction of this similarity matrix $\hat{W}_{ij}$ is the inner product between nodes v_i and v_j. (***b***) Illustration of a GNN which takes the adjacency matrix A as well as node features X as inputs and then aggregates information over neighbourhoods to update the representation of nodes H^i. The final representation Z of nodes (coloured dots in grey box) can be multidimensional or scalar for node classification. For graph classification tasks with labels Y_G, this embedding can be projected further to just one prediction for the whole graph ($\hat{Y}$).

A drawback of these—often referred to as shallow—unsupervised approaches is the linear growth of parameters with respect to the number of nodes in the graph and the failure of generalisation to unseen nodes. They are inherently transductive. One way to alleviate the latter is to extend the encoding function to also include node features X, that is to learn a parameterised function f_Φ as an encoder. This has been carried out in Graph2Gauss [26], which creates low-dimensional embeddings comparable with PAGA, t-distributed stochastic neighbour embedding or UMAP. In addition, the produced Gaussian embeddings are interpretable as the uncertainty of the embedding is directly accessible. The simultaneous treatment of features and graph structure has also been carried out in GNE which stands for gene network embedding [27]. The authors propose an architecture which first maps the topology of a gene co-expression network to a latent space using a path-based approach. This is subsequently combined with the representation of the genes' expression to find the association $P(g_i|g_j)$ between pairs of genes g_i and g_j.

Importantly, the above described methods are not necessarily task specific and all optimised through the graph structure. In many application scenarios, however, the task is not to capture some graph properties but to generalise from graph data to biological quantities using labels, for example, the classification of cancer subtypes or the identification of genic interactions. To accomplish this, we have to switch from unsupervised embedding techniques to GNNs, which have proven to be a very rich model class for graph representation learning.

Graph neural networks

GNNs have attracted a lot of attention over past years and became the standard method for graph learning. Complementary to the abovementioned information, GNNs can be understood as generalised encoder architectures which additionally use the nodes' features X and the graph structure A as input into the encoding function:

$$Z_i = f_\Phi(A, X) \ ,$$

where Φ are learnable parameters.

Motivated by the success of convolutional neural networks in computer vision, a large effort was put into the formalisation of convolutions on graphs which conceptualise the aggregation of information from a node's neighbourhood. This local propagation from nodes to nodes via edges makes GNNs structure aware and therefore, by identifying common patterns across the graph structure, suitable for many prediction tasks. Because of the generality of graphs, for which the variety of different graph types is a good example (Figure 4), the rigorous formulation of graph convolutions is not straight forward. As a result, there exist two perspectives on convolutions, spatial and spectral, where spatial convolutions are more applicable in practice as they are often computationally less expensive.

Spectral approaches build on the eigendecomposition of the graph Laplacian and are therefore not strictly localised in the original graph domain. Spatial approaches, in contrast, formulate convolutions directly on the graph and through this have to handle the variation of node degrees. Building on the theoretical work of ChebNet [28], which provided the connection between spatial and spectral convolutions, Kipf and Welling [29] proposed the model which is today known as the standard graph convolutional network, GCN (Figure 2b):

$$\begin{array}{ll} g(A,D) & = (D+I)^{-1/2}(A+I)(D+I)^{-1/2} \\ H^{l+1} & = \sigma\big(g(A,D)H^l\Phi^l\big) \ , \text{ with } H^0 = X, \end{array}$$

where Φ^l are the parameters of the l-th GCN layer, σ is a nonlinear activation function such as RELU and H is the (hidden) representation of all nodes.

In principle, a GCN can incorporate any number of layers before producing the final embedding Z. However, experiments have shown that fewer layers usually perform better, whereas too many lead to oversmoothing [30]. To overcome this issue, PPNP, personalized propagation of neural predictions, took inspiration from personal PageRank and classical graph theory to perform infinitely many propagation steps in an implicit manner, thus considering a large neighbourhood without encountering oversmoothing [31].

Cell graphs especially benefit if the spatial information is taken into account and the molecular profile can be used as node features. To date, spatial graphs are often kNN graphs inferred from the coordinates of spots or cells in the tissue. This, however, leads to the loss of directional and angular relations between the nodes which have been shown to be crucial for the predictive power of GNNs for molecular graphs [32].

On top of that, methods used in digital pathology will likely guide how graphs will be used for applications that arise from spatial molecular profiling techniques [33,34]. Pati et al. [34], for example, discuss the relevance of different length scales for tissue function and construct both a tissue graph and a cell graph before combining these with a GCN for cancer type prediction.

A first attempt to model extracellular interactions based on spatial molecular profiling data from the seqFISH protocol [12] has been carried out by the GCNG model which stands for graph convolutional neural networks for genes [35]. On top of the spatial graph, a 2-layer GCN together with a feed forward network was trained to

predict the interaction probability of gene pairs. The expression of these pairs was additionally fed in as node features, and the training was performed in a supervised manner taking known ligand-receptor pairs as labels.

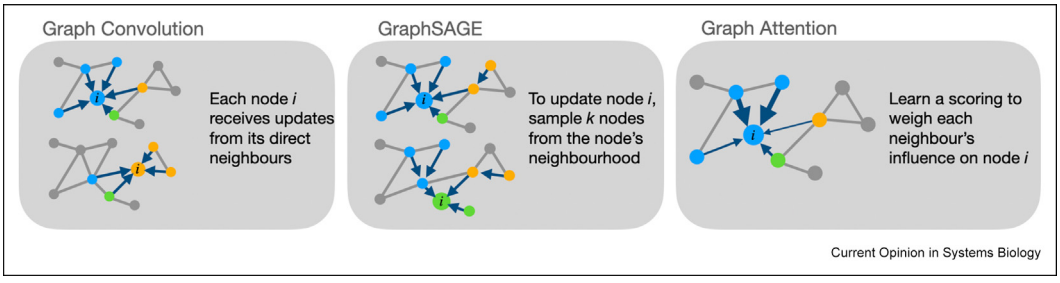
Other applications that build on spatial graphs aim at the identification of spatial domains and spatially variable genes, for which GCNs [36] yield comparable results with count-based modelling methods [37,38].

To give an example of another, purely spatial architecture, consider GraphSAGE [39] (Figure 3a) which handles the heterogeneity and dependency on the specific graph structure by a sampling method. For the neighbourhood sampling, a fixed number of k nodes is drawn proportional to the transition matrix $T = AD^{-1}$. Being dependent only on a fixed number of input features X_i, GraphSAGE becomes inductive, that is generalisable to unseen nodes in the graph, and similar to Graph2Gauss with respect to its mathematical formulation. Consequently, GraphSAGE can also be trained in an unsupervised manner as discussed in the previous section on node embeddings.

In Spage2vec, GraphSAGE has been used for the analysis of gene expression in tissues [40]. On a spatially resolved gene graph which is built from MERFISH data [13], the authors applied GraphSAGE followed by a binary classifier to predict whether pairs of genes occur co-localised. To learn the embedding, they formalise the problem as a link prediction task and train on the co-occurrence of gene pairs which is computed from 2-step random walks on the spatial gene graph. Once trained, the model generalises to inputs which have not been seen during the training phase and therefore provides putative biological hypotheses, in the above case the co-occurrence probability for any pair of genes in the data.

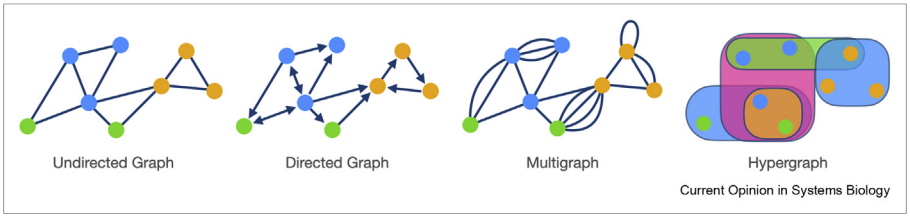
Going beyond standard graphs, see Figure 4, GNNs can also be generalised to other graph types such as multigraphs or hypergraphs and therefore provide methods to account for multidimensional or higher-order interactions in biological networks [41]. Although not based on single-cell data, DANGO [42] is an illustrative example for the inclusion of prior knowledge within the framework of GNNs. Using several, on protein-protein interaction networks pretrained GNNs, DANGO extends the view of gene interactions to trigenic processes by learning a hypergraph embedding [43]. This illustrates the potential of graph learning to tackle some of the challenges present in GRN inference in future work

Figure 3



Comparison of popular GNN architectures. From *left* to *right*: Graph Convolution, GraphSAGE, and Graph Attention. In a GCN, all nodes receive messages from their direct neighbours which are aggregated in an equal fashion. In contrast, GraphSAGE always samples k (here 6) nodes from each node's neighbourhood to perform the information aggregation. This fixed number of incoming messages makes GraphSAGE inductive. Graph attention is similar to GCN but additionally learns to weigh the incoming information of neighbours before they are aggregated. The computation of scores is identical for all nodes but through the comparison of neighbours (softmax) still sensitive to a node's individual neighbourhood.

Figure 4



From left to right: An undirected graph whose adjacency matrix A is symmetric, a directed graph where each edge is directed from one node to another, a undirected multigraph which allows for multiple edges between a pair of nodes and a hypergraph which extends the concept of edges to sets of nodes—edges are the blue, pink and green rectangles surrounding their nodes.

[11]. We expect to see similar efforts to translate this to mammalian cells and single-cell genomics, potentially overcoming the sparseness of data by imputation.

GCNs have already been used for the imputation of gene counts and denoising. For the scGNN model, Huang [44] took the cell similarity graph as input of a 2-layer GCN, where the feature dimension has been additionally downsampled after the first layer to incorporate the advantages of a bottleneck similar to the study by Eraslan et al. [45]. Looking specifically at genes and single-cell data, we see great opportunities in models that take perturbations [46] on single cells as labels and are trained to predict regulatory effects of genes [47].

Already carried out by DANGO, graph convolutions can readily be combined with attention mechanisms [48] which allow to additionally learn the neighbours' relevance in a weighted fashion (Figure 3b). Compared with GraphSAGE, the graph attention model [49] does not sample a fixed size neighbourhood but assigns a relevance score to each neighbour of a node. This is carried out by first linearly projecting each node's representation H with the weight matrix Φ and then applying a weight vector a and a nonlinearity σ to every concatenated node−neighbour pair:

$$e_{ij} = f(\Phi H_i, \Phi H_j) = \sigma(a^{\mathrm{T}}[H_i \| H_j])$$

$$\alpha_{ij} = \mathrm{softmax}(e_{ij}) = \frac{\exp(e_{ij})}{\sum_k \exp(e_{ik})}$$

A softmax function is applied in the end to normalise the result and get the final scoring α_{ij}. Note that the evaluation and comparison of these GNN methods is crucial for model design and applications and that simpler models can even outperform sophisticated ones when hyperparameters are correctly tuned [50].

From a biological perspective, it is crucial to have interpretable models [51]. A possible way to explain model predictions on graphs is to optimise over masks that effectively prune the graph such that only relevant subgraph structures remain [52]. Jaume et al. [53] use this approach in the context of digital pathology to identify subgraph regions of the cell graph which are relevant for cancer subtyping and compare this to prior pathological knowledge.

In addition, attention mechanisms also allow the interpretation of GNNs by analysing the learnt scoring weights. This was applied for the disease state prediction task on COVID data, for which the authors also considered edge features which were precomputed from the single-cell data and the graph structure [54].

Finally, variational graph autoencoders allow training a GNN in an unsupervised manner [55]. Similar to the presented methods for node embeddings in the previous section, variational graph autoencoders use the graph structure for the decoder optimisation while using a GNN for the encoder. This setting has been applied to explore an unsupervised workflow for different downstream tasks on scRNA-seq data [56], for which the authors also discuss the interpretability of the graph attention scorings.

Conclusion

In this review, we illustrated how the framework of graph representation learning applies to single-cell biology. To this end, the construction and applications of both cell and gene graphs from single-cell data have been discussed before introducing the framework of graph embeddings. We discussed the modelling aspects of both unsupervised and supervised graph representation learning and their current applications in single-cell biology. Focussing on future applications, we paid special attention to GNNs.

From a computer science perspective, there exist several open challenges which concern new models for unstudied graph structures, the expressiveness, interpretability and robustness of graph embeddings [57,58], and also the compositionality of existing model architectures [4]. Likewise, there exist a multitude of conceivable applications of graph learning in single-cell bio which have not yet been explored. Similar to the study by Buterez et al. [56], we expect more work towards end-to-end graph models that are learnt directly from the single-cell data, replacing PCA dimensionality reduction and the initial construction of kNN graphs. So far, the abstraction of spatial information to graphs has not yet been fully exploited. Through the addition of directional and angular information to spatial graphs [31], we anticipate new insights towards the interplay of tissue structure and function from graph learning techniques. Looking at genes, the inclusion of prior knowledge from laboratory experiments, as possible for transcription factor target genes or protein−protein interaction networks, will additionally guide future research on gene regulation in the framework of graph embedding.

To conclude, single-cell data pose several challenges which can be tackled by graph representation learning; however, more effort is required to take advantage of its full potential.

Conflict of interest statement

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: F.J.T. reports receiving consulting fees from Roche Diagnostics GmbH and Cellarity Inc. and ownership interest in Cellarity Inc. and Dermagnostix.

Acknowledgements

This work has been funded by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036B F.J.T. acknowledges support by the BMBF (grant# 01IS18053A) and by the Helmholtz Association's Initiative and Networking Fund through Helmholtz AI [grant number: ZT-I-PF-5-01] and sparse2big [grant number ZT-I-007].

References

Papers of particular interest, published within the period of review, have been highlighted as:

- • of special interest

1. Marx V: **Method of the year: spatially resolved transcriptomics**. *Nat Methods* 2021, **18**:9–14.

2. Luecken MD, Theis FJ: **Current best practices in single-cell RNA-seq analysis: a tutorial**. *Mol Syst Biol* 2019, **15**, e8746.

3. Chami I, Abu-El-Haija S, Perozzi B, Ré C, Murphy K: **Machine learning on graphs: a model and comprehensive taxonomy**. *arXiv* 2005, 03675.

4. Zhang Z, Cui P, Zhu W: **Deep learning on graphs: a survey**. In *IEEE transactions on knowledge and data engineering*; 2020.

5. Zhou J, Cui G, Zhang Z, Yang C, Liu Z, Wang L, Li C, Sun M: **Graph neural networks: a review of methods and applications**. *arXiv* 1812, 08434.

6. Muzio G, O'Bray L, Borgwardt K: **Biological network analysis withdeep learning**. *Briefings Bioinf* 2021, **22**(2):1515–1530.

7. Grün D: **Revealing dynamics of gene expression variability in cell state space**. *Nat Methods* 2020, **17**:45–49.

8. Islam S, Kjällquist U, Moliner A, Zajac P, Fan J-B, Lönnerberg P, Linnarsson S: **Characterization of the single-cell transcriptional landscape by highly multiplex RNA-seq**. *Genome Res* 2011, **21**:1160–1167.

9. Pratapa A, Jalihal AP, Law JN, Bharadwaj A, Murali T: **Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data**. *Nat Methods* 2020, **17**: 147–154.

10. Chen S, Mar JC: **Evaluating methods of inferring gene regulatory networks highlights their lack of performance for single cell gene expression data**. *BMC Bioinf* 2018, **19**: 1–21.

11. Saint-Antoine MM, Singh A: **Network inference in systems biology: recent developments, challenges, and applications**. *Curr Opin Biotechnol* 2020, **63**:89–98.

12. Eng C-HL, Lawson M, Zhu Q, Dries R, Koulena N, Takei Y, Yun J, Cronin C, Karp C, Yuan G-C, Cai L: **Transcriptome-scale super-resolved imaging in tissues by RNA seqFISH**. *Nature* 2019, **568**:235–239.

13. Xia C, Fan J, Emanuel G, Hao J, Zhuang X: **Spatial transcriptome profiling by MERFISH reveals subcellular RNA compartmentalization and cell cycle-dependent gene expression**. *Proc Natl Acad Sci USA* 2019, **116**: 19490–19499.

14. Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E: **Fast unfolding of communities in large networks**. *J Stat Mech* 2008, **2008**:P10008.

15. van der Maaten L: *Visualizing data using t-SNE*. 2008. https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf.

16. McInnes L, Healy J, Melville J: **UMAP: uniform manifold approximation and projection for dimension reduction**. *arXiv* 1802, 03426.

17. Saelens W, Cannoodt R, Todorov H, Saeys Y: *A comparison of single-cell trajectory inference methods*. 2019.

18. Chung NC, Storey JD: **Statistical significance of variables driving systematic variation in high-dimensional data**. *Bioinformatics* 2015, **31**:545–554.

19. Hamilton WL: **Graph representation learning**. *Synth Lect Artif Intell Mach Learn* 2020, **14**:1–159.

20. Ahmed A, Shervashidze N, Narayanamurthy S, Josifovski V, Smola AJ: **Distributed large-scale natural graph factorization**. In *Proceedings of the 22nd international conference on World Wide Web, WWW '13*. New York, NY, USA: Association for Computing Machinery; 2013:37–48.

21. Perozzi B, Al-Rfou R, Skiena S: **DeepWalk: online learning of social representations**. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '14*. New York, NY, USA: Association for Computing Machinery; 2014:701–710.

22. Grover A, Leskovec J: **node2vec: scalable feature learning for networks**. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, KDD '16*. New York, NY, USA: Association for Computing Machinery; 2016: 855–864.

23. Mikolov T, Chen K, Corrado G, Dean J: **Efficient estimation of wordrepresentations in vector space**. *arXiv* 2013, **arXiv**: 1301.3781.

24. Haghverdi L, Büttner M, Wolf FA, Buettner F, Theis FJ: **Diffusion pseudotime robustly reconstructs lineage branching**. *Nat Methods* 2016, **13**:845–848.

25. Wolf FA, Alexander Wolf F, Hamey FK, Plass M, Solana J, Dahlin JS, Göttgens B, Rajewsky N, Simon L, Theis FJ: **PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells**. *Genome Biology* 2019, **20**(1).

26. Bojchevski A, Günnemann S: **Deep Gaussian embedding of graphs: unsupervised inductive learning via ranking**. *arXiv* 1707, 03815.

27. • Kc K, Li R, Cui F, Yu Q, Haake AR: **GNE: a deep learning framework for gene network inference by aggregating biological information**. *BMC Syst Biol* 2019, **13**(Suppl 2):38.
The authors present a model for the prediction of gene interactions from the combined treatment of topological and expression information. They compare against correlation based approaches and also node2vec and show the additional impact that expression data has when the model is trained on only a fraction of gene interactions.

28. Defferrard M, Bresson X, Vandergheynst P: **Convolutional neural networks on graphs with fast localized spectral filtering**. *arXiv* 2016, **arXiv**:1606.09375.

29. Kipf TN, Welling M: **Semi-Supervised classification with graph convolutional networks**. *arXiv* 1609, 02907.

30. Li Q, Han Z, Wu X-M: **Deeper insights into graph convolutional networks for Semi-Supervised learning**. *Proceedings of the AAAI Conference on Artificial Intelligence* 2018, **32**(1).

31. Klicpera J, Bojchevski A, Günnemann S: **Predict then propagate: graph neural networks meet personalized PageRank**. *arXiv* 1810, 05997.

32. Klicpera J, Groß J, Günnemann S: **Directional message passing for molecular graphs**. *arXiv* 2003, 03123.

33. Zhou Y, Graham S, Alemi Koohbanani N, Shaban M, Heng P-A, Rajpoot N: **Cgc-net: cell graph convolutional network for grading of colorectal cancer histology images**. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*; 2019. 0–0.

34. Pati P, Jaume G, Fernandes LA, Foncubierta-Rodríguez A, Feroce F, Anniciello AM, Scognamiglio G, Brancati N, Riccio D, Di Bonito M, De Pietro G, Botti G, Goksel O, Thiran J-P, Frucci M, Gabrani M: **HACT-Net: a hierarchical Cell-to-Tissue graph neural network for histopathological image classification**. In *Uncertainty for safe utilization of machine learning in medical imaging, and graphs in biomedical image analysis*. Springer International Publishing; 2020:208–219.

35. • Yuan Y, Bar-Joseph Z: **GCNG: graph convolutional networks for inferring gene interaction from spatial transcriptomics data**. *Genome Biol* 2020, **21**:300.
A model for the inference of cell-cell interactions from spatial transcriptomics data using GCNs. Building on a spatial cell graph, GCNG learns to predict whether a pair of genes forms a ligand-receptor pair.

The authors use a curated list of interacting ligands and receptors for training and additionally examine the ability to infer causal interactions.

36. Hu J, Li X, Coleman K, Schroeder A, Irwin DJ, Lee EB, Shinohara RT, Li M: *Integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network*. Dec. 2020.

37. Svensson V, Teichmann SA, Stegle O: **SpatialDE: identification of spatially variable genes**. *Nat Methods* 2018, **15**:343–346.

38. Sun S, Zhu J, Zhou X: **Statistical analysis of spatial expression patterns for spatially resolved transcriptomic studies**. *Nat Methods* 2020, **17**:193–200.

39. Hamilton WL, Ying R, Leskovec J: **Inductive representation learning on large graphs**. *arXiv* 1706, 02216.

40. • G. Partel, C. Wählby, Spage2vec: unsupervised representation of localized spatial gene expression signatures, FEBS J.Building a spatial gene graph from spatial transcriptomics data, the authors propose a GrahSage based model that learns to find a low dimensional vector representation for each mRNA spot in the data. They train their model on the co-occurrence of genes and show how the found gene representations map to cellular identities.

41. Klamt S, Haus U-U, Theis F: **Hypergraphs and cellular networks**. *PLoS Comput Biol* 2009, **5**, e1000385.

42. • Zhang R, Ma J, Ma J: *DANGO: predicting higher-order genetic interactions*. Nov. 2020.
The first GNN based method to tackle higher order genic interactions. Based on GraphSage, the model first learns multiple node embeddings from six pairwise molecular interactions networks which are then combined for each node type (gene). Subsequently, the model computes static and dynamic embeddings of gene triplets using a self-attention mechanism which are then used to infer trigenic interactions.

43. Zhang R, Zou Y, Ma J: **Hyper-SAGNN: a self-attention based graph neural network for hypergraphs**. *arXiv* 1911, 02613.

44. Huang K: **scGNN: scRNA-seq dropout imputation via induced hierarchical cell similarity graph**. *arXiv* 2008, 03322.

45. Eraslan G, Simon LM, Mircea M, Mueller NS, Theis FJ: **Single-cell RNA-seq denoising using a deep count autoencoder**. *Nat Commun* 2019, **10**:390.

46. Dixit A, Parnas O, Li B, Chen J, Fulco CP, Jerby-Arnon L, Marjanovic ND, Dionne D, Burks T, Raychowdhury R, Adamson B, Norman TM, Lander ES, Weissman JS, Friedman N, Regev A: **Perturb-Seq: dissecting molecular circuits with scalable Single-Cell RNA profiling of pooled genetic screens**. *Cell* 2016, **167**:1853–1866. e17.

47. Lotfollahi M, Wolf FA, Theis FJ: **Scgen predicts single-cell perturbation responses**. *Nat Methods* 2019, **16**:715–721.

48. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I: **Attention is all you need**. *arXiv* 1706, 03762.

49. Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y: **Graph attention networks**. *arXiv* 2017, **arXiv**:1710.10903.

50. Shchur O, Mumme M, Bojchevski A, Günnemann S: *Pitfalls of graph neural network evaluation, relational representation learning workshop*. NeurIPS; 2018.

51. Yuan H, Yu H, Gui S, Ji S: **Explainability in graph neural networks: a taxonomic survey**. CoRR abs/2012.15445 *arXiv* 2012, 15445. https://arxiv.org/abs/2012.15445; 2012.

52. Ying R, Bourgeois D, You J, Zitnik M, Leskovec J, Gnnexplainer: **Generating explanations for graph neural networks**. *Adv Neural Inf Process Syst* 2019, **32**:9240.

53. • Jaume G, Pati P, Foncubierta-Rodriguez A, Feroce F, Scognamiglio G, Anniciello AM, Thiran J-P, Goksel O, Gabrani M: **Towards explainable graph representations in digital pathology**. arXiv *arXiv* 2007, 00311.
The first method to address explainability of graph-based approaches for Digital Pathology. Using a nuclei segmentation method and the kNN algorithm to create a cell graph, the authors train a GNN model to predict breast cancer subtypes. In a subsequent step, they prune the cell graphs to identify relevant parts in the graph for the prediction and discuss the in this way gained interpretability of the model.

54. Sehanobish A, Ravindra NG, van Dijk D: **Gaining insight into SARS-CoV-2 infection and COVID-19 severity using self-supervised edge features and graph neural networks**. *arXiv* 2006:12971.

55. Kipf TN, Welling M: **Variational graph auto-encoders**. *NeurIPS Workshop on Bayesian Deep Learning* 2016.

56. • D. Buterez, I. Bica, I. Tariq, H. Andrés-Terré, P. Liò, CellVGAE: an unsupervised scRNA-seq analysis workflow with graph attention networks, bioRxiv 2020.12.20.423645; doi: https://doi.org/10.1101/2020.12.20.423645.Combines a Graph-VAE approach with graph attention layers to find low dimensional representations of cells. The authors show how Cell-VGAE can be used for typical downstream tasks such as clustering and marker gene identification. Additionally, they discuss how to take advantage of the attention weights for visualisation of expression values and interactions across cell types.

57. Zügner D, Akbarnejad A, Günnemann S: **Adversarial attacks on neural networks for graph data**. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. New York, NY, USA: Association for Computing Machinery; 2018:2847–2856.

58. Bojchevski A, Günnemann S: **Adversarial attacks on node embeddings via graph poisoning**. In *Proceedings of the 36th international conference on machine learning, vol. 97 of proceedings of machine learning research*. Edited by Chaudhuri K, Salakhutdinov R, PMLR; 2019:695–704.