analytical chemistry

Open Access

This article is licensed under CC-BY 4.0

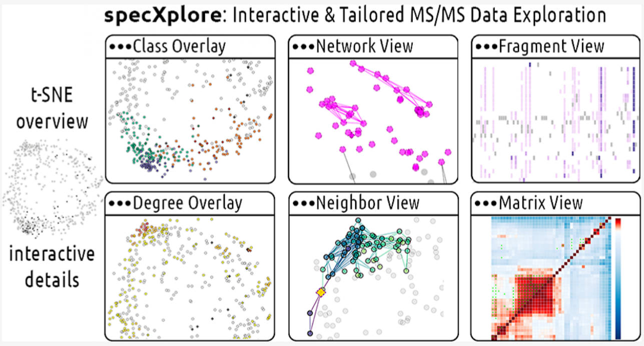
Tailored Mass Spectral Data Exploration Using the SpecXplore Interactive Dashboard

Kevin Mildau,[§] Henry Ehlers,[§] Ian Oesterle, Manuel Pristner, Benedikt Warth, Maria Doppler, Christoph Bueschl, Jürgen Zanghellini,* and Justin J. J. van der Hooft*

Cite This: *Anal. Chem.* 2024, 96, 5798−5806

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information



ABSTRACT: Untargeted metabolomics promises comprehensive characterization of small molecules in biological samples. However, the field is hampered by low annotation rates and abstract spectral data. Despite recent advances in computational metabolomics, manual annotations and manual confirmation of in-silico annotations remain important in the field. Here, exploratory data analysis methods for mass spectral data provide overviews, prioritization, and structural hypothesis starting points to researchers facing large quantities of spectral data. In this research, we propose a fluid means of dealing with mass spectral data using specXplore, an interactive Python dashboard providing interactive and complementary visualizations facilitating mass spectral similarity matrix exploration. Specifically, specXplore provides a two-dimensional t-distributed stochastic neighbor embedding embedding as a jumping board for local connectivity exploration using complementary interactive visualizations in the form of partial network drawings, similarity heatmaps, and fragmentation overview maps. SpecXplore makes use of state-of-the-art ms2deepscore pairwise spectral similarities as a quantitative backbone while allowing fast changes of threshold and connectivity limitation settings, providing flexibility in adjusting settings to suit the localized node environment being explored. We believe that specXplore can become an integral part of mass spectral data exploration efforts and assist users in the generation of structural hypotheses for compounds of interest.

■ INTRODUCTION

Untargeted metabolomics deals with the elucidation and characterization of small molecules in complex biological systems. Small molecules or metabolites cover an enormous chemical diversity involved in a vast range of biological functions. This chemical diversity leads to complex and heterogeneous data that is difficult to provide consistent and automated workflows for.[1] Computational metabolomics tools which assist manual data evaluation and annotations efforts such as experimental networking thus remain critical to the field.[2] Molecular Networking (MN) hosted on the Global Natural Products Social Molecular Networking (GNPS) servers is possibly the most used computational metabolomics tool for exploratory data analysis work using liquid chromatography tandem mass spectrometry (LC−MS/MS) data.[3−6] The core idea behind MN is that, since similar structures tend to fragment similarly, spectral similarity may be used to construct spectral feature groups with implied structural similarity. In MN, the modified cosine score similarity matrix of the measured spectra forms the basis for constructing such groups using a network topology approach.[3] The nodes in the network represent MS/MS spectral features that may be connected via edges as a function of pairwise spectral similarity. Indeed, pairwise similarity thresholds and other network processing parameters are used to filter the complete network of all possible pairwise connections such that only edges for high pairwise spectral similarities remain. Using this approach, interrelated spectra are used to form small, separated (disjoint) groups of nodes of high intraspectral similarity commonly referred to as molecular families.[3] MN can thus be viewed as an exploratory analysis framework merging topological grouping and network visualization. Molecular families serve two separate functions: (a) they provided an ordered data overview and (b) they may be used to assist network annotation propagation, that is, the propagation of structural hypotheses from known structures to unknown ones via proximity in the network.[5,7−9]

While the MN workflow is hugely successful, it comes with its own trade-offs.[5] For instance, the use of the modified cosine

Received: October 3, 2023
Revised: March 21, 2024
Accepted: March 21, 2024
Published: April 2, 2024

© 2024 The Authors. Published by American Chemical Society

https://doi.org/10.1021/acs.analchem.3c04444

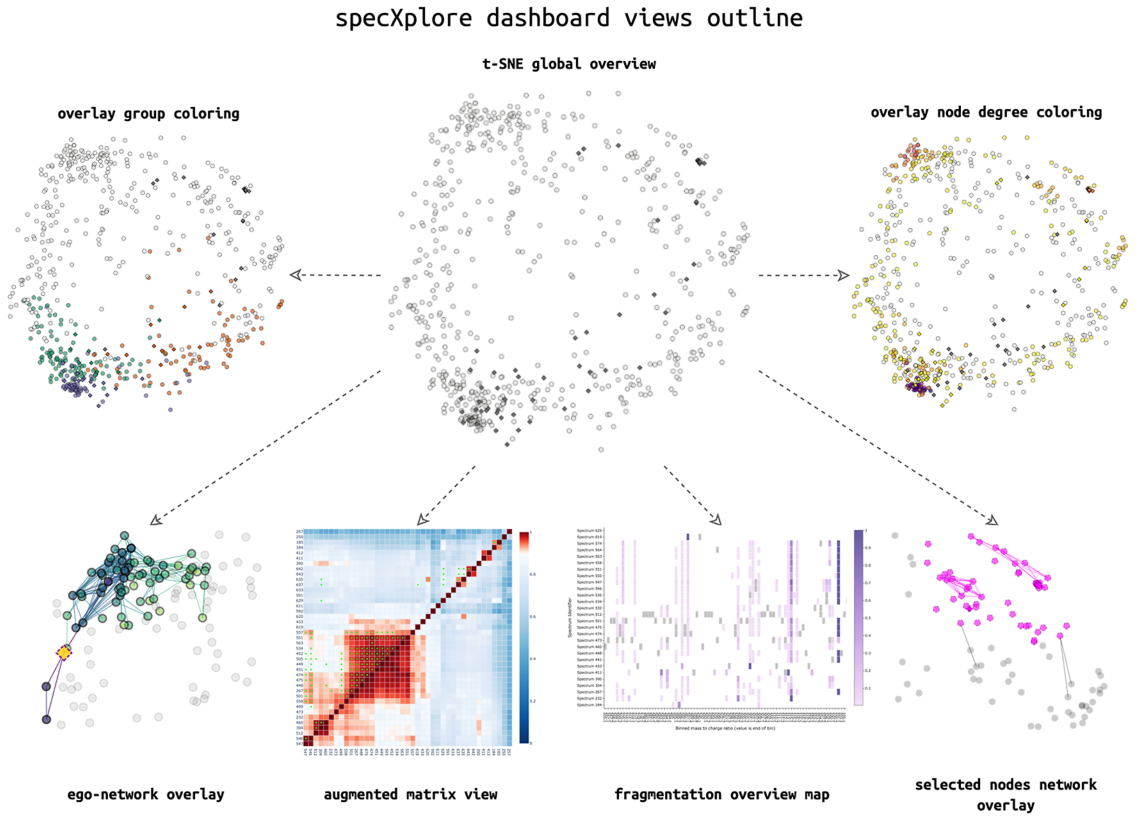

specXplore dashboard views outline
t-SNE global overview
overlay group coloring
overlay node degree coloring
ego-network overlay
augmented matrix view
fragmentation overview map
selected nodes network overlay

Figure 1. SpecXplore dashboard overview of visual components using wheat data (see the Results section). The central t-SNE overview figure provides a jumping board into three categories of visualizations in the form of color overlays such as group highlighting or node degree visualization, networks overlays such as the ego network or selection network views, and add-on panels such as augmap for quantitative insights into pairwise similarity and fragmap for insights into fragmentation overlaps across selections of features. For a video demonstration, please refer to https://youtu.be/9ZqJAr8wdv8.

score and minimum fragment overlap requirements poses a stringent similarity criterion for connectivity, resulting in rather sparse networks suitable for representation as disjoint subnetworks. However, the modified cosine score may miss structurally related analogues which exhibit larger fragmentation differences, while the disjointness of the molecular families may obscure relationships between groups. Importantly, MN operates using a single global threshold setting on the basis of which connectivity may or may not exist. Such a global threshold is unlikely to work well for all chemical families measured, where some may exhibit much richer or much sparser fragmentation or overlap thereof. In addition, the combination of disjointness of spectral groupings, as well as the disconnected runs with new randomly generated layouts for each molecular family make setting comparisons an arduous task.

In this paper, we introduce specXplore, an interactive Python dashboard aimed at facilitating spectral data exploration in a flexible and local network topology tailored fashion. Unlike traditional molecular networking, specXplore was created with adjustable settings for heterogeneous and dense network data in mind. SpecXplore provides complementary and interactive visualizations that allow users to explore connections between the spectral features using interactively adjustable network settings.

SpecXplore consists of an importing module providing data integration capacities and a dashboard-module for interactive exploratory data analysis. The dashboard makes use of a two-dimensional t-distributed stochastic neighbor embedding (t-SNE) overview network of the full pairwise similarity matrix as a jumping board for localized data exploration. Rather than being based on modified cosine scores, specXplore is based on ms2deepscore pairwise similarities, which can more accurately represent the structural similarities between compounds based on their spectra via a deep-learning-based embedding representation.[3,5,10] Being trained to predict pairwise structural similarity from spectral data, ms2deepscore in principle allows grouping of similar compounds even if their spectra are dissimilar.[10] However, ms2deepscore may also introduce a much denser topology. At many reasonable threshold levels, node-link diagram representations of dense matrices tend to become unreadable for the network as a whole.[2] To facilitate the effective exploration of local neighborhoods, specXplore provides various interactive visualizations, providing views of connectivity surrounding a feature or feature group of interest. Here, partial network drawings and matrix representations play an important role. Localized explorations are combined with the ability to quickly change thresholds to regenerate local views under the new constraints, allowing the careful expansion of neighborhood size for some node of interest. In addition to the network-based representations of local connectivity, specXplore provides means for (a) investigating the raw pairwise similarity matrix directly, (b) investigating the fragmentation overlaps across multiple spectra, and (c) inspecting any joined-in metadata or chemical classifications from within the dashboard.

We will first outline the core components of the tool, their intended usage, and their rationale. This will be followed by illustrative examples on real data and a discussion on the tool in the broader contexts of mass spectral exploratory data analyses.

■ MATERIALS AND METHODS

The specXplore workflow is divided into a Python data importing pipeline and a visual analysis dashboard using dash.[11,12] The importing part provides data integration and preprocessing functionalities, while the dashboard provides the interactive user interface for data exploration. The tool is available on github under MIT license as a python package that can be downloaded and installed for local use at https://github.com/kevinmildau/specXplore. We will briefly outline the dashboard's importing pipeline and core visual components.

Data Importing and Preprocessing. Before data can be opened in the specXplore dashboard, it needs to be processed using the specXplore importing pipeline. The processing workflows and intermediate data structures used by specXplore are built upon a cohort of open-source software Python data science packages, namely, matchms,[13] MS2Query,[14] ms2deepscore,[10] spec2vec,[15] kmedoids,[16] Cython,[17] numpy,[18] pandas,[19,20] scikit-learn,[21] and scipy.[22] The input data for specXplore spectral data exploration are MS/MS spectral data. LC−MS/MS data preprocessing (i.e., feature detection and MS/MS spectral exporting) is assumed to have been done elsewhere, e.g., using MZMine3, in order to reduce data set size and feature redundancy.[23] The MS/MS feature data are assumed to be in .mgf (mascot generic format) format, where each entry should have a unique feature identifier, a precursor mass to charge ratio, and spectral data in the form of one or more mass to charge ratio and intensity value tuples. Spectral data are imported into Python using matchms, and basic specXplore data processing is performed (see Supporting Information, Section S2.1).[13] Spectral data can then be used to initialize a template specXplore object that automatically computes pairwise spectral similarities using three similarity scores: ms2deepscore, modified cosine scores, and spec2vec scores.[3,10,13,15] For the machine learning scores, pretrained models provided with MS2Query are used.[14]

The central overview of specXplore is based on a t-SNE embedding of the ms2deepscore pairwise similarity matrix serving as the primary similarity score.[21,22,24] Functionalities are provided for users to test a range of values for the perplexity tuning parameter (usually between 5 and 50) and select an appropriate value. Users will need to balance both high-dimensional distance preservation and network layout qualitative grouping properties.

Similarly, functionalities are provided to construct a range of k-medoid clustering-based data subdivisions to complement the t-SNE embeddings. These clusters provide clear local neighborhood groupings that can be otherwise difficult to evaluate given t-SNE's abstract projection of the similarity matrix.

Finally, to provide the user with chemically informative visual highlighting capacities, MS2Query analog classifications via ClassyFire or NPClassifier, or direct classifications from tools such as Sirius, may be integrated into the specXplore session for visual highlighting capacities (see Supporting Information, Section S2.5).

SpecXplore Interactive Dashboard. SpecXplore provides the user with a variety of views and interactive navigation options (Figure 1).[25] We envision specXplore's functionalities to be used in an interactive local exploratory fashion, where the settings and their impact on local neighborhood can be evaluated seamlessly. This local exploration starts at a two-dimensional t-SNE overview figure rendered using dash and dash-cytoscape.[12,26,27] All other visualizations in specXplore make use of plotly.[28] In its most basic form, the t-SNE map only contains a node for each feature, and a number of visually highlighted spike-in standard features. Following Shneiderman's mantra of "Overview first, details on demand", specXplore allows for the selection of any node, or collection of nodes, for more in-depth analysis in complementary visualization approaches falling into two broad categories: (a) topology overlay views and (b) data details views.[29]

Three different topology-based views are provided. First, to provide insights into the impact of thresholds on the topology, a node degree overlay visualization using a color gradient to represent individual node connectivity levels can be prompted via a button click for the whole t-SNE map (Figure 1 and Supporting Information Figure S2 in Supporting Information Section S3.1.3). This node degree visualization as well as a supporting edge weight distribution plot included in the settings panel follow the principle idea of Willett et al.'s Scented Widgets, providing intuitions about the impact of settings on the topology (see Supporting Information, Section S4.5).[30] This renders adjusting settings a more informed process.

Second, to provide quantitative insights into the underlying pairwise similarity data of selections of nodes, a heatmap portraying the ms2deepscore pairwise similarity matrix for a selection is provided. Here, the ms2deepscore scores are quantitatively portrayed using a divergent color scale around the current threshold setting, while the exact numeric values are available via mouse-hover panels. This view is called augmap (AUGmented HeatMAP) in specXplore since it also incorporates implied adjacency matrices for modified cosine score and spec2vec score matrices at current threshold settings via additional markers and mouse-hover information, providing a means of comparing adjacency between the scores at current threshold levels.

As a third visualization component, we provide network visualization overlays. For single nodes, specXplore provides so-called ego-network overlays, visualizations suitable for studying a network's topology relative to a single node.[31,32] Here, all edges connecting to the selected node, but also all edges connecting to those connections and so forth, are superimposed onto the t-SNE map in line with hop distance settings. This allows the user to explore a branching view of network connectivity emanating from the ego node. For multinode selections, specXplore provides a network view for intragroup connectivity assessments highlighting all edges within a selection of nodes, as well as those connecting outward of the selection.

Finally, specXplore provides a number of data details add-on panels that can be prompted for selections of the nodes. For any group of spectra, the metadata information can be presented as a table, and MS/MS spectrum plots can be generated. Moreover, for pairwise spectral comparison, so-called mirror plots depicting one spectrum on the positive y axis and the other on the negative y axis are available. However, the latter are not suitable for multispectrum comparison and evaluations of fragmentation overlaps across more than two spectra. To support users with multispectrum comparisons, specXplore provides fragmentation overview heatmaps we refer

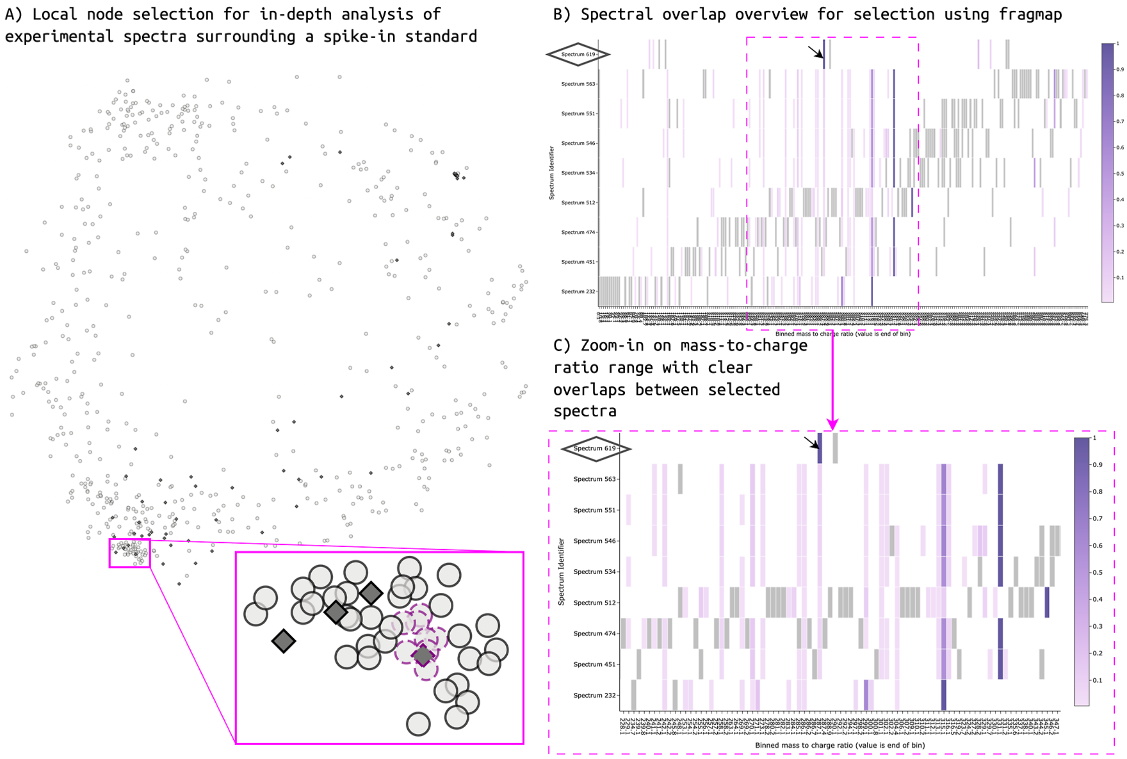


Figure 2. Process of exploring the local environment of a known feature node is illustrated. (A) Procyanidin-A2 reference standard surrounding feature selection in the lower left corner of the t-SNE embedding. The selected nodes are highlighted using magenta outlines. The selected reference standard is highlighted as a dark gray diamond with a dashed selection outline in magenta. (B) Fragmap view of the fragmentation overlap between the spectra selected in A. The reference standard spectrum is of significantly lower complexity than that of the experimental spectra. Only few fragments appear overlapping. However, many fragments are shared across experimental standards in a clear pattern indicating a structural relationship. The reference standard is highlighted on the *y*-axis using an added diamond outline. The highest intensity fragment for the reference standard is further is highlighted using an arrow. (C) Zoom-in of the overlapping area in fragmap view. All except one experimental feature in the local selection share a low relative intensity fragment ion at mass 287.055 that appears to correspond to the most intense fragment in the reference standard.

to as fragmap. To generate a fragmap, binned mass-to-charge ratios are sorted in ascending order and factorized. This allows to (a) reduce the amount of unused white space in each individual spectrum plot by putting the ascending fragments immediately next to each other regardless of mass differences and (b) to separate crowded areas of the mass-to-charge ratio axis into more easily separable pieces. The *y*-axis in the fragmap is used for aligning the different spectra, while a color gradient is used to highlight the fragment intensity. In addition, neutral losses are indicated as constant colored blocks inside the same visualization, providing a rich view of the overlaps in the fragmentation patterns. The fragmap thus provides immediate insights into the spectral overlaps or lack thereof in a single, concise overview. This in turn facilitates assessment of the meaningfulness of local connectivity.

■ RESULTS

We illustrate specXplore by applying it on real LC−MS/MS untargeted metabolomics data from two experiments: (a) wheat plants LC−MS/MS data[33,34] and (b) urine metabolome LC−MS/MS from a polyphenol exposome study.[35] In both cases, the approach and illustrations are similar. We hence focus on the wheat data here and refer to the supplement for the briefer urine data example (see Supporting Information, Section S3.2). In addition, a video overview of the different views of the tool can be found online at https://youtu.be/9ZqJAr8wdv8. Spectral data from .mgf files was loaded into the preprocessing jupyter notebooks and processed into a specXplore session object (see illustrative example notebooks https://github.com/kevinmildau/specxplore-illustrative-examples). For the wheat data set, the pipeline preprocessing time was less than 15 min (on a MacBook Pro laptop with Apple M1 Pro processor 2021), most of which was spent on local library search via MS2Query. Using the same system, the processing time for the larger urine data set was less than 75 min, whereof approximately 62 min was taken up by MS2Query and 9 min for all pairwise spectral similarity computations.

Upon opening the app and loading the data, the user faces an interactive two-dimensional projection of all spectra created by the t-SNE. Each spectral feature in the data set is represented as either as a circular node or highlighted as darkened diamonds for reference standards in this example. On its own, the t-SNE overview figure is difficult to read. Only a limited view of clustering trends can be observed through denser node regions and positioning of reference standards (Figure 1). This being the case, the t-SNE overview figure still provides an excellent basis for in-depth exploration of the data via interactivity. The most useful top-level exploration features are *k*-medoid clustering at low values of *k* (see Supporting Information, Figure S3A in Supporting Information, Section S3.1.4), chemical classification (see Supporting Information, Figure S3B in Supporting Information, Section S3.1.4), and node degree visualizations (see Supporting Information, Figure S2A in Supporting Information, Section S3.1.3).

Node degree visualization provides an immediate and uncluttered view of the network topology at current threshold levels, providing insights into topological groupings in the data (see Supporting Information, Figure S2A in Supporting Information, Section S3.1.3). In addition, node degree visualizations provide the most straightforward assessment of the impact of threshold changes on local topology while avoiding the computational cost and visual clutter of potentially large numbers of edges (see Supporting Information, Figure S2A,B in Supporting Information Section, S3.1.3).

Color-based highlighting of groupings based on *k*-medoid clustering or chemical ontology predictions can provide additional means of determining the local areas of interest in the t-SNE overview. Here, *k*-medoid clustering provides an edge-threshold-independent means of dividing the feature space into smaller groups while still making use of the pairwise similarity matrix. *K*-medoid clustering tends to show good agreement with both t-SNE projections and topological insights, while it provides a means of determining neighbor sets of nodes of interest (see Supporting Information, Figure S3A). At low *k* values, *k*-medoid clustering tends to produce larger data groupings (see Supporting Information, Figure S3A), higher values of k tend to subdivide the data into smaller, localized clusters often corresponding well to topological connectivity at higher thresholds (see Supporting Information, Figure S4 in Supporting Information, Section S3.1.4). Putative chemical classifications may serve a similar means of prioritizing areas of interest in the t-SNE embedding.

In addition to the use of various color highlighting approaches to detect groupings of interest or achieve a bird's-eye view of topology, specXplore makes use of edge overlay visualizations granting insights into local node connectivity patterns at adjustable similarity threshold settings (see Supporting Information, Figure S2B in Supporting Information, Section S3.1.3). These views provide insights on what nodes are considered adjacent to one another given the current similarity thresholds. While network views provide simplified views of the similarity relationship between nodes, augmap views provide deeper insights into the similarity matrix and quantitative backbone of all of specXplore's visualizations (see Supporting Information, Figure S1 in Supporting Information, Section S3.1.2). These quantitative insights can be used to adjust the local threshold settings accordingly or be used as an alternative to network views for small node selections altogether.

While specXplore's global overviews provide insight into the rough patterns of the data, its localized views provide more detailed insights into the possible relationships between the features in the wheat data. There are numerous ways to delve further into the data, given the areas of interest have been found. One sensible approach in specXplore is to explore connectivity around the known reference standards. For instance, exploring the densely connected feature area around the Procyanidin-A2 corresponding feature (Figure 2 A), we can see the characteristic fragment ion overlap as well as the much higher complexity of the experimental spectra (Figure 2 B,C). Such overlaps in fragmentation patterns and implied substructure overlaps alongside local topology and other metadata information may serve as vital starting points for MS/MS structural hypothesis generation and manual annotation efforts. Which features are considered of interest, and how stringent overlaps will need to be to be useful naturally depend on the analysis goals.

■ DISCUSSION

We have developed specXplore with two aims in mind: for it to provide a flexible data exploration platform and for it to provide a means of understanding and tailoring network processing settings to the data at hand. In keeping with these aims, specXplore allows the data to be explored interactively with adjustable settings in the well anchored context provided by the t-SNE embedding. In specXplore, there are no rigid subdivisions of the data nor topological parameters to "fix". We opted for this approach since we think of specXplore as providing an interface to a heterogeneous and complex network of spectral similarities. This network aims to present pairwise structural similarities between spectra via the ms2deepscore model but is impacted by data and model heterogeneity. Here, many different compound classes with different fragmentation behaviors and different model coverages are lumped together into a single, highly heterogeneous network. With data this heterogeneous, local exploration and local setting tailoring seem the most sensible. In practice, specXplore thus requires the user to delve into the network and find localized target groups based on their own criteria of interest and tolerances in spectral and implied structural similarity. While requiring more effort from the user, this also provides them with unprecedented flexibility.

During the development of specXplore, we drew inspiration from two extensions of MN, MolNetEnhancer, and MetGem (see Supporting Information, Figure S6A−C in Supporting Information, Section S4.1).[36−39] MolNetEnhancer extends MN by providing molecular families with a dominant ms2lda motif classification as a visually highlighteable component of molecular networks in Cytoscape.[40,41] MetGem extends MN by providing an interlinked network and a t-SNE visualization, providing a means of inspecting the data from two complementary angles at once.[24,37] Both MolNetEnhancer of MetGem have found use in the field (e.g., refs 42−44 and refs 45−47) and highlight the potential in extending and tailoring the MN workflows.

We also drew inspiration from the network visualization tool EdgeMaps (see Supporting Information, Figure S6D in Supporting Information, Section S4.1).[39,48] In EdgeMaps, a complex and dense network is embedded in a two-dimensional projection of node similarities, and directed edges between any interconnected nodes are visualized only upon interactive demand.

SpecXplore makes use of the embedding approach of MetGem and EdgeMaps to create a similarity preserving layout and makes use of interactive prompting as in EdgeMaps to highlight the local topological relationships. In addition, chemical space prioritization is facilitated through chemical-classification-based node coloring as done in MolNetEnhancer.

Naturally, as a tool for spectral data exploration, specXplore draws inspiration from Network Annotation Propagation, be it automatic or manual, where we designate the primary aim of specXplore to be the exploration of the local topology with the goal of structural hypothesis propagation and spectral cross comparison.[8,49,50] Combining elements of all these approaches, specXplore is a uniquely flexible data exploration approach for mass spectral data that covers a broad range of complex network visualization tasks.[51]

Impact of Settings in Traditional Molecular Networking and SpecXplore. MN, as hosted on GNPS, provides an interesting and successful framework for topology-based mass spectral exploratory data analysis (EDA).[4] EDA is characteristically dynamic and flexible, yet the analysis approach and settings may have a strong impact on how the data is viewed and used. The primary topological settings in MN are (a) the pairwise similarity thresholds used as well as corresponding minimum fragmentation overlaps, (b) top-K neighbor limits on individual nodes, and (c) maximal molecular family sizes. Additionally, one can consider the choice of the modified cosine score (or spec2vec scores) as the underlying metric a setting. In the GNPS workflow, these settings are used to create a subdivision of the spectral data into disjoint groupings that are visualized separately as subnetworks. This subdivision is possible because of the general connection sparsity encouraged by the settings: modified cosine similarity matrices will be comparatibely sparse, while typical default thresholds of 0.7 will lead to even sparser adjacency matrices. In addition, top-K limitations on the number of neighbors for each node can limit the scope of hub nodes and reduce cross-network connectivity. Thus, the combination of settings and visualization approaches are tailored to accommodate sparse representations and not dense networks. A feature of MN is thus that that molecular families represent groups of high spectral similarity with no visible links to other families. Missing connections between nodes and clusters through restrictive settings, but in rarer instances also hard to decipher hairball molecular networks through too liberal settings, may be encountered.

With settings being as impactful, it is hence a key issue that GNPS reruns with different settings are slow, while comparisons of runs from one to another are nontrivial. Indeed, MN produces disjoint molecular families to be analyzed separately, while reruns with different settings produce different molecular families in size, composition, and natural ordering. In addition, individual molecular families make use of randomized force-directed layouts, leading to possibly different node positioning in each run and family. This means that the preservation of any kind of mental map of the different runs and how they compare against one another are difficult. While the speed bottleneck of traditional MN on GNPS can be partly overcome with tools such as MetGem or MZMine, which allow fast local reruns separating data processing from networking settings, the comparison of different runs to one another still remains difficult.[23,37]

The comparability issue is addressed in specXplore via its local subnetwork and information on demand approach. The use of fixed t-SNE coordinates as layout allows the user to create a mental map of the data, as well as allows them to study the impact of on-the-fly changeable network settings on their requested local views easily against the thus provided visual anchor-point.

In addition to fundamental visual design differences, specXplore makes use of the ms2deepscore model for generating the pairwise similarity matrix underlying its visualizations.[10] This model is used as it provides better capabilities for linking spectra with stronger fragmentation differences, albeit at the cost of denser networks that are more challenging to visualize.[10] However, since ms2deepscore is a machine learning model it can only be expected to work well for spectra and metabolites well covered by or related to its training data. To accommodate this deficit at least in part, specXplore provides the augmap views granting insights into the differences between ms2deepscore, modified cosine scores, and spec2vec scores on the same data.

The specXplore dashboard has been shown in our illustrative examples to work well for the wheat data and the urine data sets containing <1000 and <4000 features each after processing. For these smaller, processed data sets, specXplore works provides broad flexibility and interconnected visual analysis features. We expect that the dashboard will scale well to 5000 features but face difficulties for larger data sets and liberal settings. Network representations may quickly become visually overwhelming or computationally demanding to process and render. The feature-rich and liberal settings approach of specXplore does not lend itself to repository scale analyses.

Dense Networks and Network Layout Choice. Since specXplore contains a large network analysis component, the choice of feature positioning in its general overview is in part a question of network layout choice. Laying out dense networks is a difficult task often addressed using force-directed layout algorithms owing to their computational tractabiity.[52−55] However, the latter do not scale well to large and dense networks and tend to produce hard-to-read or unintelligable networks rendition.[55,56] This is particularly because of often created dense "hairballs" of nodes and edges, as well as edge-crossings.[57−59] In specXplore, this is addressed by a combination of latent variable space embedding of nodes with interactively triggered partial network drawings.[37,39,60,61] Here, the latent variable embedding serves as a jumping board for localized network explorations. This approach is in line with Shneiderman's mantra of "Overview first; details on demand".[29] Dense hairball visualizations or edges traversing the whole t-SNE embedding are avoided by only visualizing edges on demand, using stringent user-modifiable edge filter settings.

Alternative approaches available to dealing with poor readibilty make use of summarization.[62] The nodes can be hierarchically aggregated, or the edges can be bundled.[63,64] Such approaches however tend to alter the preceived relationships within the graph and ultimately require interactivity such as hypernode expansion or semantic zooming to allow insights into the various levels of granularity.[65−67] Hence, no matter the approach taken, some form of interactive data visualization is needed to handle dense networks. The approach used in specXplore aims to only minimally alter the perceived topology of the network and provide intuitive and easily understandable forms of interaction via interactive overlays.

It should be noted, however, that the use of t-SNE as the layout approach in specXplore does not come without disadvantages. Embeddings produced by t-SNE are built to preserve local neighborhoods in the high-dimensional space in their projection to a two-dimensional space.[24] This focus may lead to difficulties in interpretation of t-SNE results as neglect of global similarity preservation may render cluster proximity a poor indicator of cluster similarity.[68] Alternative approaches such as UMAP or PaCMAP may be more capable at preserving global or even global and local trends.[69−71] However, recent research shows that t-SNE artifacts may be avoided with very careful method tuning.[68,72] We note that t-SNE's local focus works well within specXplore as it tends to group together high-similarity nodes, while global-similarity distortions are partially offset by the use of complementary visualization overlays such as network representations which afford a view of effective connectivity between groupings at given threshold levels.

***K*-Medoid Clustering in SpecXplore.** The t-SNE embedding overview panel of specXplore provides the user with an abstract and condensed representation of the pairwise similarity matrix lacking a clear node grouping structure. To alleviate this problem, specXplore provides complementary *k*-medoid clustering color overlays, where clustering at various values of k provides quick glances at local neighborhoods in the t-SNE overview. The *k*-medoid clustering algorithm is closely related to *k*-means clustering, where *k*-medoid omits the necessity of computing some form of centroid against which to measure distances, instead making use of the median distanced observation within a cluster, i.e., the medoid, as the reference against which to measure distances.[73] *K*-medoid clustering thus has a number of advantages within specXplore: (a) any arbitrary distance measure may be used with *k*-medoid clustering, including ms2deepscore itself, (b) making use of medoids circumvents the need of defining centroids, as well as any associated needs for recomputing distances from the latter making it computationally cheap, and (c) since *k*-medoid operates directly on the similarity matrix its cluster assignments are unaffected by t-SNE projection artifacts.

We considered making use of hierarchical clustering with medoid linkage for maintaining cluster consistency across levels of granularity.[74] Clusters being hierarchically subdivided into subclusters, which in turn are further subdivided, and so on, would provide a mental map advantage when exploring different granularity levels. For *k*-medoid clustering, no such agreement across different values of *K* is enforced, and hence, cluster assignments may vary across settings of *k* sometimes grouping features together and sometimes not. However, due to both lacking implementation availability in Python and suboptimality of the hierarchically constrained clusters at any level of *k*, we opted to use *k*-medoid clustering only.[74] In specXplore, we make use of *k*-medoid clustering as an assistive grouping approach rather than an end-point. Hence, different values of k are used to provide variable granularity groupings to be further explored in the general t-SNE embedding and using other complementary views such as partial network drawings. More work on measuring and comparing optimality would be needed to provide users with stronger guidelines for automatic cluster tuning.

■ CONCLUSIONS

SpecXplore provides a means of interactively slicing into the complete matrix of pairwise spectral similarities via adjustable settings and complementary views. Exploration of the data in this way provides a means of determining spectral neighborhoods of interest and to assist direct and indirect network annotation propagation. In addition, specXplore exposes the impact of topological filtering settings on effective topology and local neighborhood contexts. Being based on ms2deepscore, it further allows for state-of-the-art similarity scoring that reflects structural similarities more closely. This in turn opens up opportunities for finding similarities missed by traditional scoring approaches. SpecXplore thus provides users with flexible state-of-the-art data exploration platform. Future works for specXplore we are considering are (a) providing an online hosted version for more straightforward accessibility and (b) an integration with statistical testing results for more effective prioritization leveraging experimental designs (such as in FERMO[75]).

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.analchem.3c04444.

Supplementary notes and details regarding the materials discussed in the manuscript (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Jürgen Zanghellini − *Department of Analytical Chemistry, University of Vienna, 1090 Vienna, Austria;* orcid.org/0000-0002-1964-2455; Email: juergen.zanghellini@univie.ac.at

Justin J. J. van der Hooft − *Bioinformatics Group, Wageningen University, 6708PB Wageningen, The Netherlands; Department of Biochemistry, University of Johannesburg, 2006 Johannesburg, South Africa;* orcid.org/0000-0002-9340-5511; Email: justin.vanderhooft@wur.nl

Authors

Kevin Mildau − *Department of Analytical Chemistry, University of Vienna, 1090 Vienna, Austria; Austrian Centre of Industrial Biotechnology (ACIB GmbH), 8010 Graz, Austria; Doctoral School in Chemistry, University of Vienna, 1090 Vienna, Austria*

Henry Ehlers − *Institute of Visual Computing and Human-Centered Technology, 1040 Vienna, Austria*

Ian Oesterle − *Department of Food Chemistry and Toxicology, Department of Biophysical Chemistry, and Doctoral School in Chemistry, University of Vienna, 1090 Vienna, Austria;* orcid.org/0000-0003-4106-5452

Manuel Pristner − *Department of Food Chemistry and Toxicology and Doctoral School in Chemistry, University of Vienna, 1090 Vienna, Austria*

Benedikt Warth − *Department of Food Chemistry and Toxicology, University of Vienna, 1090 Vienna, Austria;* orcid.org/0000-0002-6104-0706

Maria Doppler − *University of Natural Resources and Life Sciences (BOKU), 3430 Tulln, Austria*

Christoph Bueschl − *University of Natural Resources and Life Sciences (BOKU), 3430 Tulln, Austria*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.analchem.3c04444

Author Contributions

[§]K.M. and H.E. are shared first authors.

Notes

The authors declare the following competing financial interest(s): JJJvdH is member of the Scientific Advisory Board of NAICONS Srl., Milano, Italy and consults for Corteva Agriscience, Indianapolis, IN, USA. All other authors declare to have no competing interests.

■ ACKNOWLEDGMENTS

The wheat dataset was kindly provided by Rainer Schumacher and the BOKU Core Facility Bioactive Molecules: Screening and Analysis. K.M. received support from the COMET center acib: Next Generation Bioproduction, which is funded by

Ministry of Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMVIT), Federal Ministry for Digital and Economic Affairs (BMDW), Steirische Wirtschaftsförderungsgesellschaft m.b.H. (SFG), Standortagentur Tirol, Government of Lower Austria und Vienna Business Agency in the framework of COMET—Competence Centers for Excellent Technologies (project 94011). The COMET-Funding Program is managed by the Austrian Research Promotion Agency FFG.

■ REFERENCES

(1) de Jonge, N. F.; Mildau, K.; Meijer, D.; Louwen, J. J. R.; Bueschl, C.; Huber, F.; van der Hooft, J. J. J. *Metabolomics* **2022**, *18*, 103.

(2) Amara, A.; Frainay, C.; Jourdan, F.; Naake, T.; Neumann, S.; Novoa-del Toro, E. M.; Salek, R. M.; Salzer, L.; Scharfenberg, S.; Witting, M. *Front. Mol. Biosci.* **2022**, *9*, 841373.

(3) Watrous, J.; Roach, P.; Alexandrov, T.; Heath, B. S.; Yang, J. Y.; Kersten, R. D.; van der Voort, M.; Pogliano, K.; Gross, H.; Raaijmakers, J. M.; Moore, B. S.; Laskin, J.; Bandeira, N.; Dorrestein, P. C. *Proc. Natl. Acad. Sci. U.S.A.* **2012**, *109*, E1743−E1752.

(4) Wang, M.; Carver, J. J.; Phelan, V. V.; Sanchez, L. M.; Garg, N.; Peng, Y.; Nguyen, D. D.; Watrous, J.; Kapono, C. A.; Luzzatto-Knaan, T.; et al. *Nat. Biotechnol.* **2016**, *34*, 828−837.

(5) Nothias, L.-F.; Petras, D.; Schmid, R.; Dührkop, K.; Rainer, J.; Sarvepalli, A.; Protsyuk, I.; Ernst, M.; Tsugawa, H.; Fleischauer, M.; et al. *Nat. Methods* **2020**, *17*, 905−908.

(6) Nothias, L.-F.; Nothias-Esposito, M.; da Silva, R.; Wang, M.; Protsyuk, I.; Zhang, Z.; Sarvepalli, A.; Leyssen, P.; Touboul, D.; Costa, J.; Paolini, J.; Alexandrov, T.; Litaudon, M.; Dorrestein, P. C. *J. Nat. Prod.* **2018**, *81*, 758−767.

(7) Fox Ramos, A. E.; Evanno, L.; Poupon, E.; Champy, P.; Beniddir, M. A. *Nat. Prod. Rep.* **2019**, *36*, 960−980.

(8) Qin, G.-F.; Zhang, X.; Zhu, F.; Huo, Z.-Q.; Yao, Q.-Q.; Feng, Q.; Liu, Z.; Zhang, G.-M.; Yao, J.-C.; Liang, H.-B. *Molecules* **2022**, *28*, 157.

(9) Tian, Z.; Liu, F.; Li, D.; Fernie, A. R.; Chen, W. *Comput. Struct. Biotechnol. J.* **2022**, *20*, 5085−5097.

(10) Huber, F.; van der Burg, S.; van der Hooft, J. J. J.; Ridder, L. *J. Cheminf.* **2021**, *13*, 84.

(11) Foundation, P. S. Python Programming language. https://www.python.org/, Last Accessed 01 26, 2024.

(12) Hossain, S. Visualization of Bioinformatics Data with Dash Bio. *Proceedings of the 18th Python in Science Conference*. 2019; pp 126−133.

(13) Huber, F.; Verhoeven, S.; Meijer, C.; Spreeuw, H.; Castilla, E. M. V.; Geng, C.; van der Hooft, J.; Rogers, S.; Belloum, A.; Diblen, F.; Spaaks, J. H. *J. Open Source Softw.* **2020**, *5*, 2411.

(14) de Jonge, N. F.; Louwen, J. R.; Chekmeneva, E.; Camuzeaux, S.; Vermeir, F. J.; Jansen, R. S.; Huber, F.; van der Hooft, J. J. *bioRxiv* **2022**.

(15) Huber, F.; Ridder, L.; Verhoeven, S.; Spaaks, J. H.; Diblen, F.; Rogers, S.; van der Hooft, J. J. J. *PLoS Comput. Biol.* **2021**, *17*, No. e1008724.

(16) Schubert, E.; Lenssen, L. *J. Open Source Softw.* **2022**, *7*, 4183.

(17) Behnel, S.; Bradshaw, R.; Citro, C.; Dalcin, L.; Seljebotn, D.; Smith, K. *Comput. Sci. Eng.* **2011**, *13*, 31−39.

(18) Harris, C. R.; Millman, K. J.; van der Walt, S. J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N. J.; et al. *Nature* **2020**, *585*, 357−362.

(19) McKinney, W. Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*. 2010; pp 56−61.

(20) The pandas development team. *Pandas-Dev/pandas: Pandas*; Zenodo, 2020..

(21) Pedregosa, F.; et al. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.

(22) Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. *Nat. Methods* **2020**, *17*, 261−272.

(23) Schmid, R.; Heuckeroth, S.; Korf, A.; Smirnov, A.; Myers, O.; Dyrlund, T. S.; Bushuiev, R.; Murray, K. J.; Hoffmann, N.; Lu, M.; et al. *Nat. Biotechnol.* **2023**, *41*, 447−449.

(24) van der Maaten, L.; Hinton, G. *J. Mach. Learn. Res.* **2008**, *9*, 2579−2605.

(25) Yi, J. S.; Kang, Y. a.; Stasko, J.; Jacko, J. *IEEE Trans. Visualization Comput. Graphics* **2007**, *13*, 1224−1231.

(26) Franz, M.; Lopes, C. T.; Huck, G.; Dong, Y.; Sumer, O.; Bader, G. D. *Bioinformatics* **2016**, *32*, 309−311.

(27) Franz, M.; Lopes, C. T.; Fong, D.; Kucera, M.; Cheung, M.; Siper, M. C.; Huck, G.; Dong, Y.; Sumer, O.; Bader, G. D. *Bioinformatics* **2023**, *39*, btad031.

(28) Plotly Technologies Inc.. *Collaborative Data Science*, 2015. https://plot.ly.

(29) Shneiderman, B., Interactive Technologies. In *The Craft of Information Visualization*; Bederson, B. B., Shneiderman, B., Eds.; Morgan Kaufmann: San Francisco, 2003; pp 364−371.

(30) Willett, W.; Heer, J.; Agrawala, M. *IEEE Trans. Visualization Comput. Graphics* **2007**, *13*, 1129−1136.

(31) Wu, Y.; Pitipornvivat, N.; Zhao, J.; Yang, S.; Huang, G.; Qu, H. *IEEE Trans. Visualization Comput. Graphics* **2016**, *22*, 260−269.

(32) Shi, L.; Wang, C.; Wen, Z. Dynamic network visualization in 1.5D. In *2011 IEEE Pacific Visualization Symposium*, 2011; pp 179−186. ISSN: 2165−8773.

(33) Doppler, M.; Bueschl, C.; Kluger, B.; Koutnik, A.; Lemmens, M.; Buerstmayr, H.; Rechthaler, J.; Krska, R.; Adam, G.; Schuhmacher, R. *Front. Plant Sci.* **2019**, *10*, 1366.

(34) Bueschl, C.; Doppler, M.; Varga, E.; Seidl, B.; Flasch, M.; Warth, B.; Zanghellini, J. *Bioinformatics* **2022**, *38*, 3422−3428.

(35) Oesterle, I.; Pristner, M.; Berger, S.; Wang, M.; Verri Hernandes, V.; Rompel, A.; Warth, B. *Anal. Chem.* **2023**, *95*, 10686−10694.

(36) Ernst, M.; Kang, K. B.; Caraballo-Rodríguez, A. M.; Nothias, L.-F.; Wandy, J.; Chen, C.; Wang, M.; Rogers, S.; Medema, M. H.; Dorrestein, P. C.; van der Hooft, J. J. *Metabolites* **2019**, *9*, 144.

(37) Olivon, F.; Elie, N.; Grelier, G.; Roussi, F.; Litaudon, M.; Touboul, D. *Anal. Chem.* **2018**, *90*, 13900−13908.

(38) Elie, N.; Santerre, C.; Touboul, D. *Anal. Chem.* **2019**, *91*, 11489−11492.

(39) Dörk, M.; Carpendale, S.; Williamson, C. EdgeMaps: visualizing explicit and implicit relations. Visualization and Data Analysis 2011. In *Proceedings of the SPIE*, 2011.

(40) van der Hooft, J. J. J.; Wandy, J.; Barrett, M. P.; Burgess, K. E. V.; Rogers, S. *Proc. Natl. Acad. Sci. U.S.A.* **2016**, *113*, 13738−13743.

(41) Shannon, P.; Markiel, A.; Ozier, O.; Baliga, N. S.; Wang, J. T.; Ramage, D.; Amin, N.; Schwikowski, B.; Ideker, T. *Genome Res.* **2003**, *13*, 2498−2504.

(42) Li, Y.; Cui, Z.; Li, Y.; Gao, J.; Tao, R.; Li, J.; Li, Y.; Luo, J. *J. Pharm. Biomed. Anal.* **2022**, *209*, 114523.

(43) Nephali, L.; Steenkamp, P.; Burgess, K.; Huyser, J.; Brand, M.; van der Hooft, J. J. J.; Tugizimana, F. *Front. Plant Sci.* **2022**, *13*, 920963.

(44) Tinte, M. M.; Masike, K.; Steenkamp, P. A.; Huyser, J.; van der Hooft, J. J. J.; Tugizimana, F. *Metabolites* **2022**, *12*, 487.

(45) Hebra, T.; Elie, N.; Poyer, S.; Van Elslande, E.; Touboul, D.; Eparvier, V. *Metabolites* **2021**, *11*, 444.

(46) Hebra, T.; Pollet, N.; Touboul, D.; Eparvier, V. *Sci. Rep.* **2022**, *12*, 17310.

(47) Sorres, J.; Hebra, T.; Elie, N.; Leman-Loubière, C.; Grayfer, T.; Grellier, P.; Touboul, D.; Stien, D.; Eparvier, V. *Molecules* **2022**, *27*, 1182.

(48) Dork, M.; Carpendale, S.; Williamson, C. *Inf. Vis.* **2012**, *11*, 5−21.

(49) da Silva, R. R.; Wang, M.; Nothias, L.-F.; van der Hooft, J. J. J.; Caraballo-Rodríguez, A. M.; Fox, E.; Balunas, M. J.; Klassen, J. L.;

Lopes, N. P.; Dorrestein, P. C. *PLoS Comput. Biol.* **2018**, *14*, No. e1006089.

(50) Morehouse, N. J.; Clark, T. N.; McMann, E. J.; van Santen, J. A.; Haeckl, F. P. J.; Gray, C. A.; Linington, R. G. *Nat. Commun.* **2023**, *14*, 308.

(51) Lee, B.; Plaisant, C.; Parr, C. S.; Fekete, J.-D.; Henry, N. Task Taxonomy for Graph Visualization. In *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors*; Novel Evaluation Methods for Information Visualization: New York, NY, USA, 2006; pp 1−5.

(52) Eades, P. *Congressus Numerantium* **1984**, *42*, 149−160.

(53) Fruchterman, T. M. J.; Reingold, E. M. Graph drawing by force-directed placement. In *Software: Practice and Experience*; Wiley, 1991; Vol. *21*, pp 1129−1164..

(54) Kamada, T.; Kawai, S. *Inf. Process. Lett.* **1989**, *31*, 7−15.

(55) Kobourov, S. G. *arXiv* **2012**, arXiv:1201.3011.

(56) Purchase, H. *Which Aesthetic Has the Greatest Effect on Human Understanding?*; Graph Drawing: Berlin, Heidelberg, 1997, pp 248−261.

(57) Purchase, H. C.; Carrington, D.; Allder, J.-A. *Empir. Softw. Eng.* **2002**, *7*, 233−255.

(58) Purchase, H. C.; Pilcher, C.; Plimmer, B. *IEEE Trans. Visualization Comput. Graphics* **2012**, *18*, 81−92.

(59) Kobourov, S. G.; Pupyrev, S.; Saket, B.. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*; Bayro-Corrochano, E., Hancock, E., Eds.; Springer International Publishing: Cham, 2014; Vol. *8827*, pp 234−245. Series Title: Lecture Notes in Computer Science.

(60) Parkkinen, J.; Nybo, K.; Peltonen, J.; Kaski, S. Graph visualization with latent variable models. In *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*: New York, NY, USA, 2010; pp 94−101.

(61) Bruckdorfer, T.; Cornelsen, S.; Gutwenger, C.; Kaufmann, M.; Montecchiani, F.; Nöllenburg, M.; Wolff, A. Progress on Partial Edge Drawings; *Graph Drawing*; Springer: Berlin, Heidelberg, 2013; pp 67−78.

(62) Liu, Y.; Safavi, T.; Dighe, A.; Koutra, D. *ACM Comput. Surv.* **2018**, *51*, 1−34.

(63) Elmqvist, N.; Fekete, J.-D. *IEEE Trans. Visualization Comput. Graphics* **2010**, *16*, 439−454.

(64) Zhou, H.; Xu, P.; Yuan, X.; Qu, H. *Edge bundling in information visualization*; Tsinghua Science and Technology, 2013; Vol *18*, pp 145−156..

(65) Gray, K.; Li, M.; Ahmed, R.; Rahman, M. K.; Azad, A.; Kobourov, S.; Börner, K. *IEEE Trans. Visualization Comput. Graphics* **2024**, *30*, 1564−1578.

(66) Wiens, V.; Lohmann, S.; Auer, S. Semantic Zooming for Ontology Graph Visualizations. In *Proceedings of the Knowledge Capture Conference*: New York, NY, USA, 2017; pp 1−8.

(67) Figueiras, A. Towards the Understanding of Interaction in Information Visualization. In *2015 19th International Conference on Information Visualisation*, 2015; pp 140−147. ISSN: 2375−0138.

(68) Kobak, D.; Berens, P. *Nat. Commun.* **2019**, *10*, 5416.

(69) McInnes, L.; Healy, J.; Melville, J. *arXiv* **2018**, arXiv:1802.03426.

(70) McInnes, L.; Healy, J.; Saul, N.; Großberger, L. *J. Open Source Softw.* **2018**, *3*, 861.

(71) Wang, Y.; Huang, H.; Rudin, C.; Shaposhnik, Y. *J. Mach. Learn. Res.* **2021**, *22*, 1−73.

(72) Chatzimparmpas, A.; Martins, R. M.; Kerren, A. *t-viSNE: A Visual Inspector for the Exploration of T-SNE. IEEE Information Visualization (VIS'18)*, Berlin, Germany, 2018.

(73) Schubert, E.; Rousseeuw, P. J. Faster k-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms. In *Similarity Search and Applications*; Springer: Cham, 2019; pp 171−187.

(74) Schubert, E.; *HACAM: Hierarchical Agglomerative Clustering Around Medoids-And its Limitations* Labor and Workforce Development Agency, 2021; pp 191−204.

(75) Zdouc, M. M.; Bayona Maldonado, L. M.; Augustijn, H. E.; Soldatou, S.; de Jonge, N.; Jaspars, M.; van Wezel, G. P.; Medema, M. H.; van der Hooft, J. J. J. *bioRxiv* **2022**.