2017

XIC Clustering By Baseyian Network

Kyle J. Handy
Computer Sciences

XNET CLUSTERING BY BAYESIAN NETWORK

By

KYLE JEFFREY HANDY

B. S. Computer Science, Gonzaga University, Spokane, WA, 2015

Thesis

presented in partial fulfillment of the requirements
for the degree of

Master of Science
in Computer Science

The University of Montana
Missoula, MT

July 2017

Approved by:

Scott Whittenburg, Dean of The Graduate School
Graduate School

Dr. Rob Smith, Chair
Computer Science

Dr. Travis Wheeler,
Computer Science

Dr. Chris Palmer
Chemistry

Handy, Kyle, M.S., July 2017 Major
Computer Science

XIC Clustering by Bayesian Network

Chairperson: Dr. Rob Smith

In mass spectrometry (MS) based proteomics, the identification and quantification of the molecules in a sample is possible with the accurate delineation of isotope signal groups known as *isotopic envelopes*. Many techniques attempt to discover isotopic envelopes with searches for known isotope signal patterns. An emerging approach, however, is to modularize the problem by first delineating individual isotope signals known as *extracted ion chromatograms (XICs)*, then clustering XICs into isotopic envelopes. In both cases, existing approaches suffer from their dependence on user parameters and hard decision thresholds. We present XIC Clustering by Bayesian Network (XNet), a machine learning approach that uses a Bayes network to cluster XICs. XNet doesn't require user parameters, and performs comparably with optimized alternatives. XNet's learning model can be extended with additional ground truth data. We demonstrate XNet's clustering performance against three prominent XIC clustering solutions*: OpenMS Feature Finder Centroided*, *msInspect* and *MaxQuant*.

Table of Contents

Introduction

Mass spectrometry (MS) is a popular technique capable of identifying and quantifying many constituent molecules in a physical sample. MS is an excellent technique for chemical and biological investigations, such as drug development and biomarker detection. The technique takes place in a mass spectrometer instrument, wherein molecules are ionized and separated by mass. The electric current induced by the ions are detected alongside their masses, with current strength proportionate to ion abundance. Each ion produces a signal referred to as an *extracted ion chromatograms (XIC).* MS sample analysis yields 3-dimensional signals comprised of molecular intensities at given mass-to-charge (*m/z*) ratios per retention time (*RT*). In a raw MS output, points coalesce in the form of *isotopic envelopes* for each detected molecule (Figure 1) at each charge state (*z*). An isotopic envelope comprises a collection of signal groups referred to as extracted ion chromatograms, with each XIC corresponding to a particular isotope of the molecule.

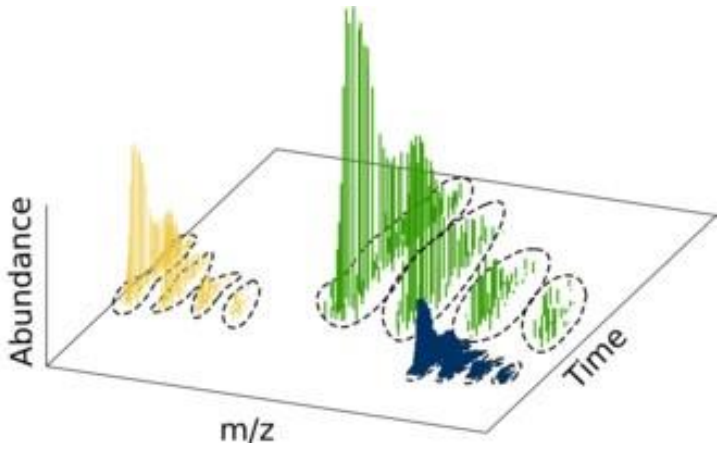

Figure 1: An isotopic envelope (shown in yellow, green and blue) consists of a set of extracted ion chromatograms (XICs) shown with dashed lines. XICs form along the RT (time) axis. Isotopic envelopes comprise XICs along the *m/z* axis linearly.

Segmentation of raw MS data points into isotopic envelopes yields a more accurate molecular quantification than other techniques, and may provide additional information to assist

in molecular identification. To date, MS identification and quantification is performed via labeling, targeting or ad-hoc manual segmentation. The profile view of an isotopic envelope presents an *isotope pattern* (Figure 2). An isotope pattern is a signature of a molecule, given by the masses (m/z) and naturally occurring relative intensities of the molecule's isotopes. By matching theoretical, pre-computed isotope patterns (green pikes, Figure 2) with experimentally measured isotopic envelopes (black waveform, Figure 2) the identity of the molecule can be ascertained. Additionally, an integration of the isotopic envelope's intensity measurements yields the overall abundance of associated molecule.

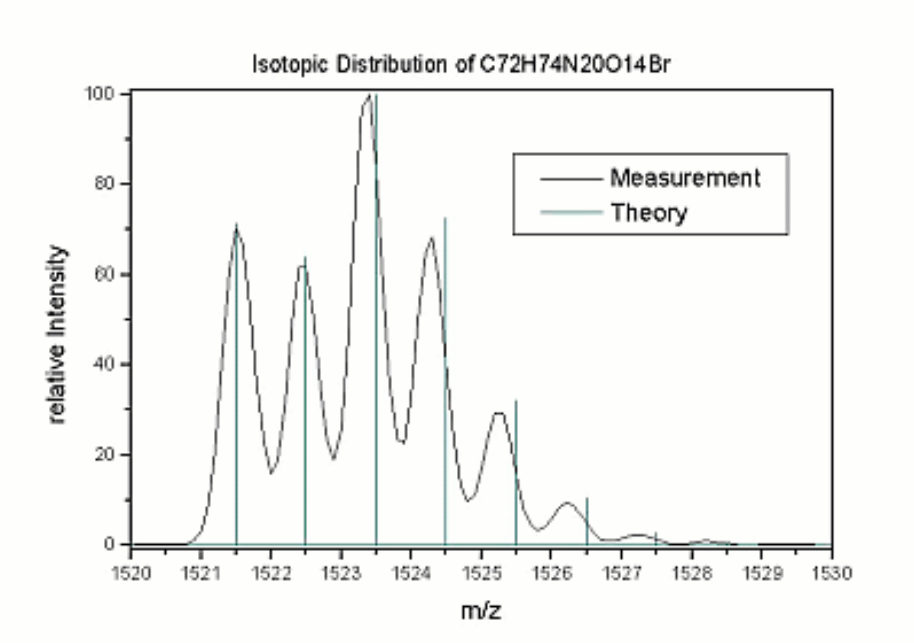

Figure 2: Matching experimental (black waveform) and theoretical (green pikes) isotope patterns. Relative XIC intensity and XIC masses provided a molecular signature.

Labeling techniques (e.g. SILAC[1], iTRAQ[2]) enable quantification of a specific set of compounds in a sample in various ways. SILAC[1], for example, allows for the quantification of a *labelled compound* in a sample by introducing a *label compound* prior to mass spectrometer analysis. The label compound is a slightly modified version of the labelled compound, creating a marginally translated isotopic envelope amongst the labelled compound's isotopic envelop. A specific quantity of the *label compound* is introduced, enabling absolute quantification of the expected compound by measuring via the intensity ratios between label and labelled isotopic envelope[1]. Labeling techniques such as SILAC are very limited. Generally, each labelled

compound must be identified and analyzed prior to quantification[1], excluding labelling techniques from identification applications. Label compounds tend to be expensive, constraining the number of labelled compounds per experiment. Additionally, there are technical limitations on the amount of labeling that can be performed per experiment. Both factors restrict the coverage of labelling techniques to a small percentage per experiment.[3]

Targeting techniques occur at the instrumental level by diverting a subset of molecules for additional analysis based on local intensity maxima or otherwise. Tandem MS (MS/MS)[4] is the primary targeting technique in MS data analytics. In some applications, such as proteomics, the mass of some precursor molecules is not informative enough to derive the molecule's identity.[4] For instance, proteins of identical molecular composition may have differing structures, presenting two or more different molecules with nearly or exactly the same masses. Many MS/MS applications attempt so enhance molecular information through *collision induced dissociation (CID)*, in which *precursor molecules* are diverted into a secondary analyzer and fragmented by CID, creating *product molecules*.[4] The behavior of CID is well understood, and so products of CID are more readily identified than their precursor molecules via database matching algorithms.[4] Targeting is an attractive option for MS segmentation, however it too is constrained by technical and experimental limitations. Primarily, diversion of precursor molecules is very limited per experiment—a select class or set of molecules must be specified for diversion.[4] Next, the typical inclusion of multiple precursor molecule signals in each MS/MS spectrum make identification of targeted compounds challenging, particularly for low abundance molecules. As a result, segmentation coverage is typically limited to around 10% of the whole sample.[5]

High-coverage segmentation is possible with ad-hoc manual segmentation. However, labor requirements render manual segmentation nonviable for high-throughput, high-coverage MS segmentation. Typical MS data files measure on the order of tens of gigabytes, containing hundreds of millions of data points.[6] Manual segmentation requires manually processing every data point within a dataset; even with the expectation of a human being able to process groups of points at once (e.g. 100), and a liberal processing rate of 3 seconds per group, manual segmentation requires on the order of one person-year per dataset. MS datasets are expected to increase in size,[6] so manual segmentation will continue to decrease in feasibility.

Labelling methods cannot perform high-throughput quantification, both labeling and targeting methods cannot perform high-coverage identification/quantification, and manual segmentation techniques are intractable. Automated techniques for high-coverage MS segmentation are therefore needed.

There are two major techniques for automated segmentation: isotope pattern searching and two-stage segmentation. Existing software packages that attempt to solve the problem of high-coverage automated MS segmentation are OpenMS Feature Finder Centroided (FFC),[7] SuperHIRN (discontinued),[8] MaxQuant[9] and msInspect.[10] FFC and SuperHIRN both employ isotope pattern searching. In both products, candidate signal sets are compared to a database of precomputed isotope patterns. Precomputed isotope patterns are assigned a similarity score based on the difference in m/z and intensity values between corresponding peaks in the candidate and precomputed isotope patterns. The similarity score of the closest matching precomputed isotope pattern is used to assess the candidate signals. Searching for isotope patterns is a high-level approach to MS segmentation–XICs are isotopic envelopes are delineated in tandem rather than individually. Isotope pattern searching algorithms suffer from combinatoric complexity in the number of raw data points (*N*) and number of isotope patterns (*M*). Give the average isotope pattern cardinality (*K),* each combination of *K* data points must be must be compared to *M* isotope patterns. The resulting time complexity is $M\binom{N}{K}$.

In two-stage segmentation, there are two modular steps: XIC segmentation and XIC clustering. MaxQuant and msInspect are two software packages that have adopted this approach. The first module segments raw MS data into XICs. The second module clusters the XICs into isotopic envelopes. Two-stage segmentation allows the user to choose the best-performing algorithm for each problem. In addition, the two-stage approach is far less computationally complex than pattern searching. Linear complexity solutions to XIC segmentation exist, and XIC clustering can be performed in an agglomerative manor with, at worst, quadratic complexity.

Regardless of the approach, most automated MS segmentation software packages suffer from the same two flaws: reliance on empirical data and hard thresholds. FFC, SuperHIRN, msInspect use empirically-derived, static datasets–such as an isotope pattern database–to approve, score or otherwise evaluate raw MS data. Employing a database enables the recognition of expected signals, but additionally determines a recognition boundary. Signals that are beyond the boundary (i.e. are not recorded within the database) will not be recognized, even if the signals

are legitimate. Insufficient database coverage directly translates to poor segmentation coverage; unrecognized signals are discounted or ignored.

Next, each software package is heavily parametrized. Each of FFC, SuperHIRN, MaxQuant and msInspect have many user parameters for MS segmentation, many of which perform as hard thresholds. For example, FFC exposes the *minimum feature score* parameter to the user, a threshold that excludes candidate isotopic envelopes (features) with insufficient scores. In most cases, users will rely on default settings for parameters without verification,[11] likely resulting in a sub-optimal configuration. Presenting many parameters is dangerous because sub-optimal configurations will often degrade experimental performance.[11] Optimal configurations– ones resulting in the highest possible accuracy—are theoretically possible with user-settings, but the performance of parametrized algorithms is unlikely to translate to practice.[11]

We present XIC Clustering by Bayesian Network (XNet), an XIC clustering module designed to participate in two-stage segmentation. XNet is a machine learning approach to XIC clustering that is designed to be adaptable, flexible, and independent of user parameters or hard thresholds. XNet uses a Bayes network to infer the likely composition of isotopic envelopes. As a machine learning model, the Bayes network in XNet is trainable on fully annotated ground truth data. Training makes XNet extensible, allowing XNet to adapt and improve as ground-truth MS segmentation data is obtained. In addition, extensibility allows XNet to train for specific applications. For portability, XNet is implemented in Java (version 8).

Methods

Latent Properties of an Isotopic Envelope

XNet is designed to make clustering decisions based on the latent properties of isotopic envelopes. The following properties are characteristic of all isotopic envelopes, providing a foundation for isotopic envelope recognition. The first two properties constrain the positioning of *adjacent XICs*—this term refers to the pairs of XICs nearest one another within an isotopic envelope.

1. **Valid XIC Separation:** Each pair of adjacent XICs has an *m/z* separation of

$1/z, z \in Z_{>0}$.

2. **Consistent XIC Separation:** Each pair of adjacent XICs has the same *m/z* separation throughout the isotopic envelope.
3. **Concurrent XIC Emergence:** The profile (intensity trace along the RT axis) of each XIC should correlate, i.e. onset, apex and attenuate concurrently.

An *XIC neighborhood* for a given XIC can be determined using isotopic envelope properties 1 and 3. An XIC neighborhood for an XIC *x* is the set of all XICs that could feasibly be adjacent to *x* within an isotopic envelope, as defined by isotopic envelope properties 1 and 3. Property 1 constrains adjacent XICs to be no further than 1 *m/z* apart ($z = 1$), with variance tolerance. Property 2 constraints adjacent XICs to emerge concurrently. This property is enforceable by requiring potential adjacent XICs to at least have overlap on the RT axis. Altogether, the XIC neighborhood for a given XIC is the set of all XICs that are within 1.1 Daltons on the *m/z* axis (maximum *m/z*-separation with variance tolerance) and have overlap on the RT axis.

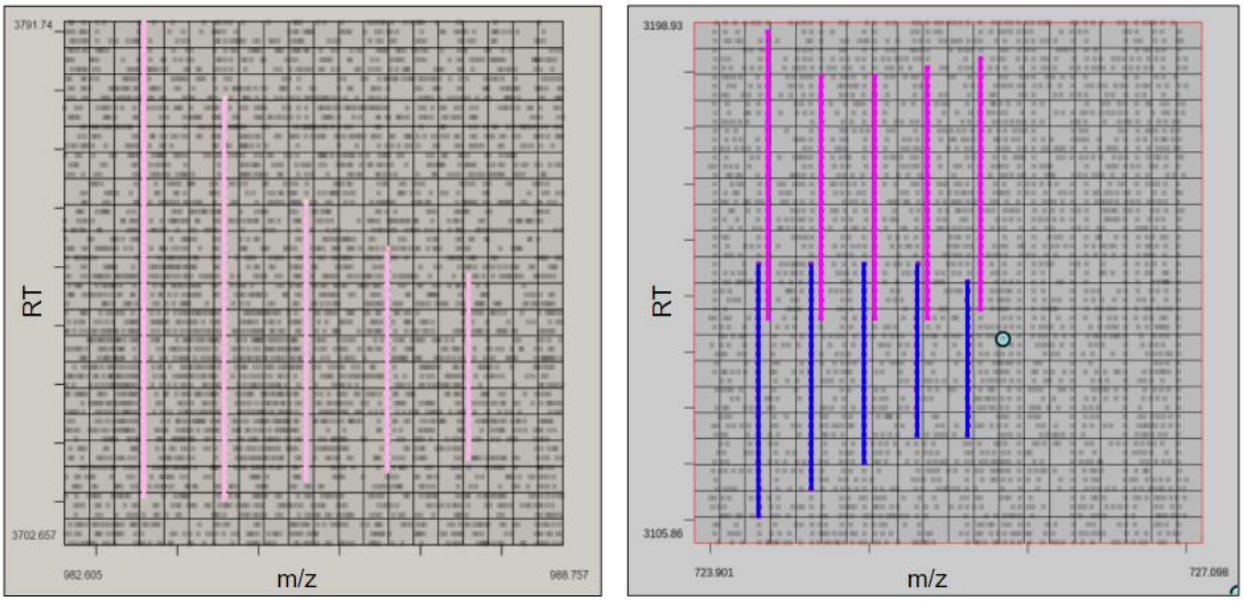

Figure 3: The left isotopic envelope is standalone–clustering XIC neighborhoods alone would result in the correct envelope. The right image shows two isotopic envelopes with significant overlap in both the m/z and RT dimensions.

Step 1: Enumerate Edges

Clustering XICs into isotopic envelopes is a trivial process when considering a standalone envelope—one without nearby or overlapping envelopes (Figure 3, left). In this case, the union of each XIC's neighborhood can determine the correct cluster. However, in many instances, isotopic envelopes emerge with significant overlap or adjacent to each other (see Figure 3. right). For this reason, XIC clustering algorithms must be capable of handling standalone, overlapping or adjacent isotopic envelopes.

XNet approaches the clustering problem graphically, modelling XICs as vertices. Edges are formed between XIC vertices that are potentially adjacent to each other (Figure 4B). The standalone/overlapping problem is approached by first creating preliminary clusters based on XIC neighborhoods. For each XIC in a dataset, an edge is enumerated between the XIC and all potentially adjacent XICs. For each XIC within the neighborhood with RT overlap and an *m/z*-separation less than 1.1 Daltons, an edge is enumerated. Edges are stored in an undirected, weighted graph with XICs as nodes (see Figure 4B). Using connected component analysis, the graph is decomposed into preliminary clusters.

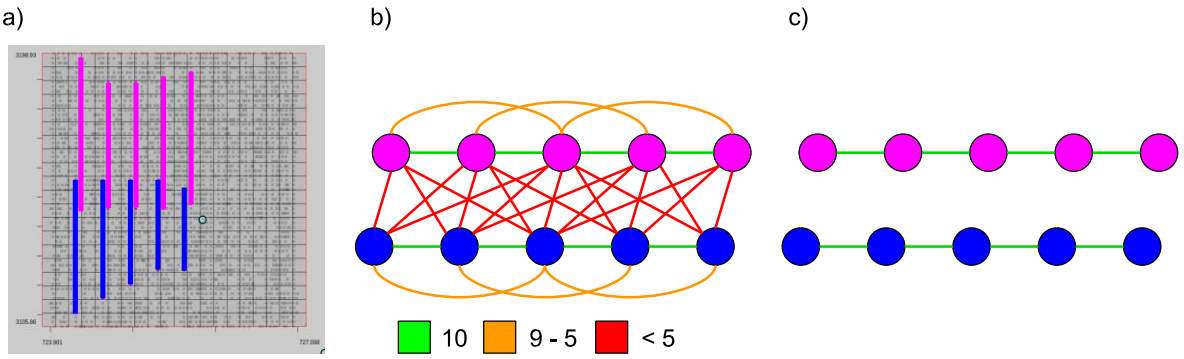

Figure 4: XIC clustering groups nearby/overlapping XICs (A) into a preliminary cluster (B); each edge in the cluster is scored on its likelihood of connecting truly adjacent XICs. Culling and consistency analysis refine the preliminary cluster into isotopic envelopes (C).

The XICGrid object (Figure 5) is a data structure used to facilitate constant-time access to XIC neighborhoods. The XICGrid is statically configured with a cell-width (*m/z*-axis) equal to the maximum *m/z*-separation of adjacent XICs, plus 10% tolerance. The cell-height (RT axis) is

set to distinguish overlapping XICs from non-overlapping XICs. Each cell contains a list of XICs that overlap the cell's data range (Figure 5C). An XIC's neighborhood can be retrieved by collecting all other XICs within the XIC's neighborhood cells. An XIC's neighborhood cells are its containing cells, the left-adjacent cells and the right-adjacent cells (Figure 5D).

For any given MS data file, if k is the average XIC neighborhood size, each XIC must compare to k other XICs on average. Using the XICGrid on n XICs, XIC clustering has a linear complexity of $O(kn)$, a vast improvement on the cubic and exponential complexities of standard clustering techniques.

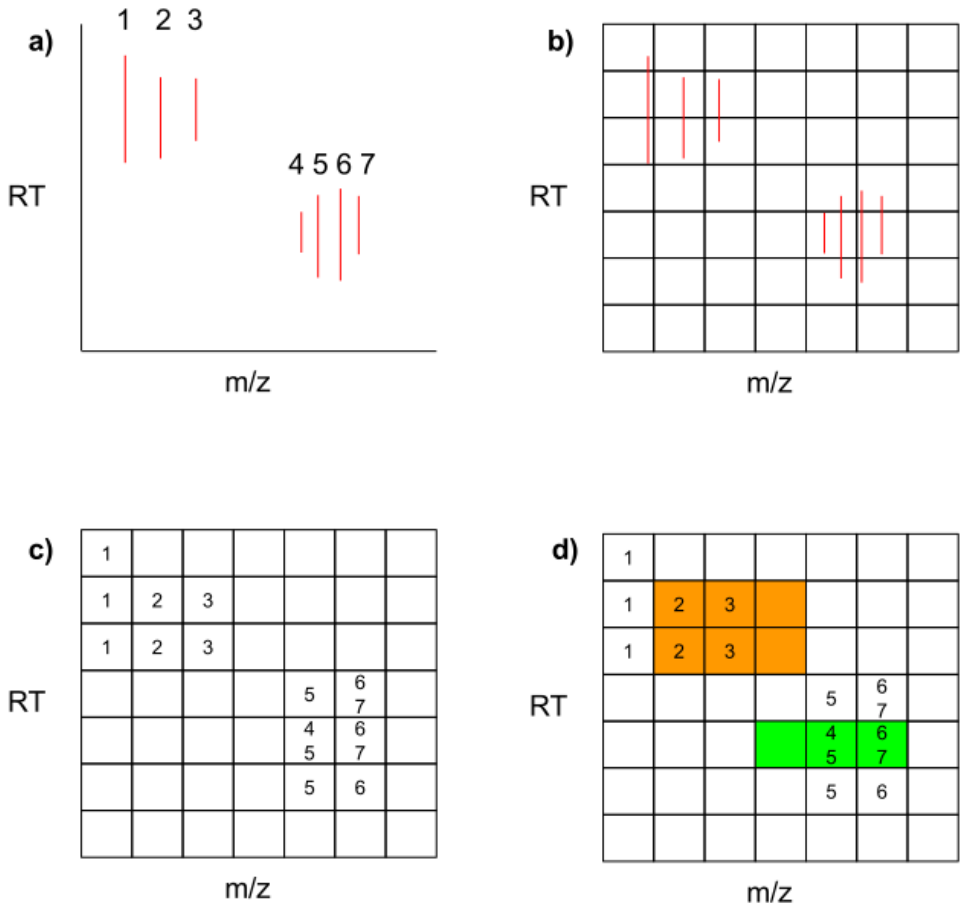

Figure 5: An XICGrid allows for constant-time retrieval of XIC neighborhoods. The isotopic envelopes in (A) are loaded into an XICGrid (B). All cells in which an XIC appears collects the XIC's GUID (C). The neighborhood cells of XICs 3 and 4 are shown in (D) in orange and green, respectively.

Step 2: Score Edges–Bayesian Network

Preliminary clusters are likely to contain XICS from more than one isotopic envelope. Each edge is scored on its likelihood of connecting truly adjacent XICs. Edges are scored by a Bayesian network tailored to the problem of XIC clustering.

A Bayesian network is a machine learning model that captures the likelihoods of, and influences between, a set of random variables.[12] Bayesian networks are useful for their ability to infer "most probable explanations"[12] based on a set of observations. Bayesian networks illuminate the likely state of hidden (unobserved) variables given the states of evidence (observed) variables.[12] In most settings, a Bayesian network is used as a query interface for predicting outcomes.

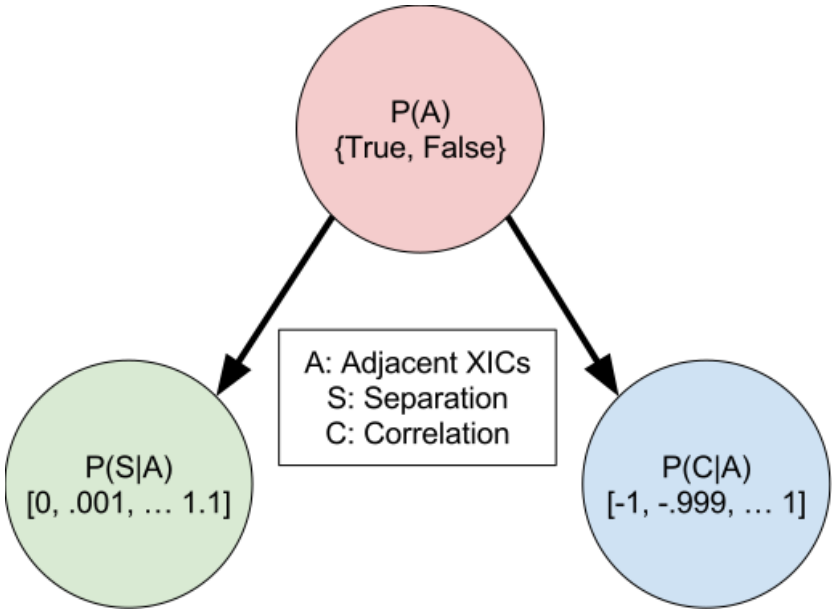

Figure 6: The Bayes Net used for inferring the likelihood of two XICs being adjacent. Each node represents a random variable, whose outcomes are shown. *m/z*-separation (S) and correlation (C) are observable random variables, influenced by the hidden adjacent XICs (A) random variable.

A Bayesian network is represented as a directed acyclic graph (DAG), where nodes are random variables and arcs represent influences between random variables.[12] Influencing nodes (arc source nodes) are referred to as parent nodes. Internally, a node's random variable is

maintained as a conditional probability table (CPT);[12] each parent random variable is a condition in the random variable's CPT.[12] In a well-crafted Bayesian network, random variables with theoretical or empirically observed influences are positioned as parents to the random variables they influence.

The Bayesian network shown in Figure 6 infers the likelihood of two XICs being adjacent. It consists of three random variables: *adjacent XICs (A)*, *m/z-separation (S)* and *XIC correlation (C)*. With these variables, the Bayesian network can accept the *m/z*-separation and correlation of two XICs, then return the likelihood of the two XICs being adjacent. Each edge generated during enumeration is scored by the likelihood its two XICs are adjacent ($A = true$).

Influences were assigned from A to S and A to C (Figure 6). Truly adjacent XICs will have an *m/z*-separation near to values $1/z$ and a high XIC correlation, whereas nonadjacent XICs will have an an *m/z*-separation other than $1/z$ and poor XIC correlation. These theoretical influences motivate the Bayesian network configuration. The resulting CPTs for this configuration are $P(A)$, $P(A|N)$ and $P(A|N)$.

Finally, each random variable must be populated with a set of outcomes. A is boolean in nature–two XICs are, or are not, adjacent–so A has the outcomes of *true* or *false*. The quantities recorded by S or C are numeric, however. S measures the separation between two XICs on the *m/z* axis, constrained by the XIC neighborhood to a maximum of $1.1 m/z$. S also has an inherent minimum of 0 *m/z*. C is measured using Pearson's correlation coefficient, which has a range of $[-1,1]$. C's outcomes reflects this range. Both S and C outcomes have a step size of $.001$. This step size is theoretically sufficient for distinguishing significantly distinct observations in S and C. The Bayesian network can infer the likelihood of two XICs being adjacent with the query $P(A = true|S = s, C = c)$, given m/z-separation s and correlation c. Resulting from the query is a probability value in the range $[0,1]$, the result of which is assigned to the edge as its edge score.

Determining $P(A = true|S = s, C = c)$ is not immediately obvious, especially since the Bayesian network stores only $P(A)$, $P(S|A)$ and $P(C|A)$. Bayesian inference is the process by which responding to the query $P(A = true|S = s, C = c)$ becomes possible. In a Bayesian network, inference begins with the Conditional Probability Formula,[13] which is shown for the query $P(A = true|S = s, C = c)$ in equation 1. The following is a derivation of the query $P(A = true|S = s, C = c)$ expressed in terms $P(A)$, $P(S|A)$ and $P(C|A)$, starting from equation 1.

$$P(A = true|S = s, C = c) = \frac{P(A = true, S = s, C = c)}{P(S = s, C = c)} \quad (1)$$

By the chain rule[13] the numerator in equation 1 can be rewritten as:

$$P(A = true, S = s, C = c) = P(A = true)P(S = s|A = true)P(C = c|S = s, A = true) \quad (2)$$

Due to the common cause relationship[13] between S and C, conditional independence is granted between S and C given A.[13] In the case of equation 2, C is independent of S given A, implying the equivalence:

$$P(C = c|S = s, A = true) = P(C = c|A = true) \quad (3)$$

By substitution, equation 2 can be rewritten as:

$$P(A = true, S = s, C = c) = P(A = true)P(S = s|A = true)P(C = c|A = true) \quad (4)$$

Each of the terms in the right hand side of equation 4 is within the known distributions $P(A)$, $P(S|A)$ and $P(C|A)$. Derivation of the numerator can halt.

Determining the denominator $P(S = s, C = c)$ in equation 1 requires summing $P(S = s, C = c)$ over all values for the nuisance variable A,[13] i.e. evaluating the expression:

$$\sum_a^A P(A = n, S = s, C = c) \quad (5)$$

As we have just demonstrated with the chain rule and conditional independence, the summed term can be transformed to:

$$\sum_a^A P(A = a)P(S = s|A = a)P(C = c|A = a) \quad (6)$$

Each term in equation 6 is known, completing derivation of the denominator. Substituting the derived numerator and denominator into equation 1 results in:

$$P(A = true|S = s, C = c) = \frac{P(A = true)P(S = s|A = true)P(C = c|A = true)}{\sum_{a \in A} P(A = a)P(S = s|A = a)P(C = c|A = a)} \quad (7)$$

Each CPT in equation 7 is stored in the Bayesian network, and so servicing the query $P(A = true|S = s, C = c)$ is a matter of accessing the necessary probabilities and computing the result. By using equation 7, the likelihood of two XICs being adjacent can be assessed.

Probability Models

There are three probability models available to populate the CPTs contained in the Bayesian network. Normally, machine learning models are trained on pre-existing ground truth data. Unfortunately, fully annotated ground truth MS1 data is quite scarce in Mass Spectrometry. The only way to attain fully annotated ground truth data is by manual segmentation, a very time intensive process. We have fully annotated ground truth data collected from the industry recognized UPS2 dataset[14] by hand-labeling 1776 isotopic envelopes comprising 6682 XICs, from which the Bayesian network can be trained. This is not enough fully annotated ground truth data to effectively train the Bayesian Network. The number of observable outcomes is 2.2M (1100 separation outcomes * 2000 correlation outcomes), most outcomes would have a recorded likelihood of zero. To accommodate the lack of fully annotated ground truth data, XNet is equipped with three different probability models from which the Bayesian network can be populated.

Bayesian Probability

In Bayesian probability theory, prior knowledge is used to form reasonable expectations on outcome likelihoods. XNet is equipped with a Bayesian probability model that does not require ground truth MS segmentation data to populate the CPTs in XNet's Bayesian network. This model is founded on isotopic envelope properties 1 and 3 (the prior knowledge). It is reasonably expected for adjacent XICs ($N = true$) to have an m/z-separation of $1/z$ (property 1), and to have a high correlation (property 3). To reflect these expectations, a reasonably expected $P(S|N = true)$ should favor values nearer to $1/z$, and a reasonably expected $P(C|N = true)$ should favor values nearer to 1.

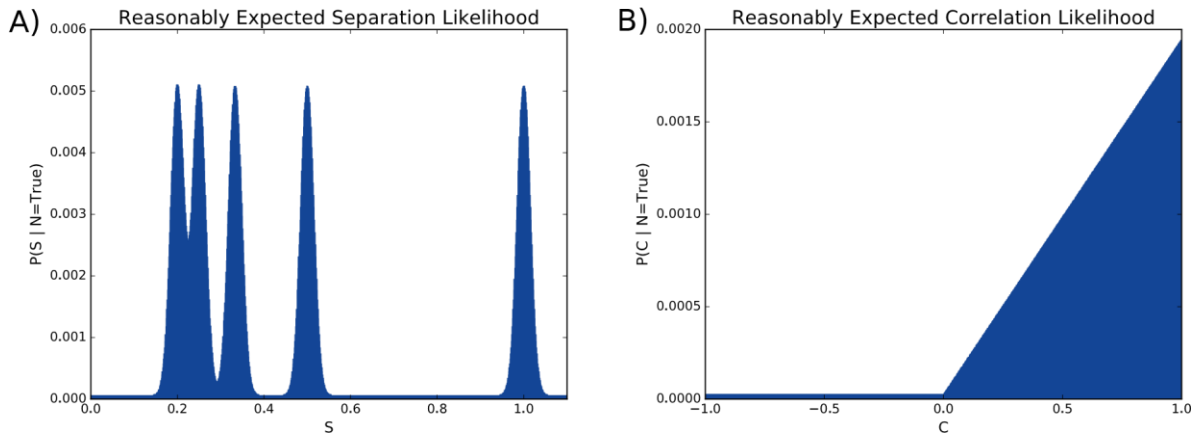

Figure 7: Reasonably expected CPTs for m/z-separation S (A) and correlation C (B), given adjacent XICs A is true. Given $A = true$, S is expected to likely measure near $1/z$ for $z \in \{1,2,\ldots,5\}$, and C is expected to likely measure near 1 with zero and negative measurements expected to be unlikely.

Figure 7 displays reasonably expected CPTs $P(S|A = true)$ and $P(C|A = true)$. $P(S|A = true)$ is populated with a series of normal curves, each with a mean of $1/z$ and standard deviation of 0.01. There is one normal curve per $z \in \{1,2,\ldots,5\}$, each with a corresponding mean at $1/z$. Normal curves were chosen to emulate reasonably expected dissipation in probability as S departs from $1/z, z \in \{1,2,\ldots,5\}$. The standard deviation 0.01 was selected so that subtle deviations in m/z-separation received an adequate probability penalty, and so that interference between normal curves was minimized (see the normal curves at $1/4$ and $1/5$ in Figure 7). Semantically, this instantiation of $P(S|A = true)$ implies that given truly adjacent XICs $(A = true)$, m/z-separation outcomes near $1/z$ are most likely, with likelihood dissipating as m/z-separation departs from $1/z$. $P(C|A = true)$ is populated proportionately to the rectified linear unit function–a popular activation function for neural networks[15]–because it emulates the reasonably expected probabilities of $P(C|A = true)$: given a truly adjacent XICs $(A = true)$, higher correlations are more likely, and negative correlations are just as unlikely as no correlation $(C = 0.0)$. To avoid zero-scores, all outcomes in both CPTs are initialized with a small starting value.

Frequentist Probability

In frequentist probability theory, outcome likelihood is based on the outcome's observed propensity, i.e. the proportion of times the outcome has been observed. Frequentist theory is the backbone of all machine learning models, where prediction models are trained on prelabeled ground truth data. The frequentist probability model in XNet is no different; given fully annotated ground truth data, XNet uses the contained observations to initialize the Bayesian network's CPTs. An XNet user can instruct XNet to train on such a dataset.

XNet is able to persistently store, load and update a frequentist probability model derived from fully annotated ground truth data in the form of a JSON file. In the event of training, XNet will output a JSON file containing the network's CPTs. The persisted model can be reloaded for further XIC clustering, or further trained in the event of ground truth data. XNet comes pre-packaged with a default JSON probability file, storing the frequentist probability model observed from the fully annotated ground truth UPS2 dataset. The model contained within this file is ready to be used in XIC clustering, and can be extended.

Hybrid Probability

Finally, XNet allows for a hybrid probability model combining both the Bayesian and frequentist approach. The hybrid model allows the reasonably expected CPTs to be extended by ground truth observations. The intent of this approach is to compensate for the scarcity of fully annotated ground truth data with the Bayesian model, and use whatever ground truth data is available for fine-tuning.

Logistically, the hybrid model operates nearly identically to the frequentist model; a JSON file persists the probability model and allows for reuse and updating. The only difference is that the model is initialized to have the CPTs of the Bayesian Probability Model.

Step 3: Cull Edges

Preliminary clusters are likely to contain more than one isotopic envelope. More specifically, isotopic envelopes within 1.1 on the *m/z* axis and within 0.5 on the RT axis will be assigned to the same preliminary cluster. Culling is performed on each preliminary cluster to

extract the isotopic envelopes within. Each of the resulting clusters is referred to as a *culled cluster*.

Procedure

Culling iteratively processes a preliminary cluster's edges in descending order of edge score. Each iterated edge is *accepted* (Figure 8), dubbing the edge's XICs adjacent within an isotopic envelope. The highest scoring edge at each iteration is the most likely pair of adjacent XICs; culling uses the score of each edge as a heuristic to determining the most likely isotopic envelopes.

If an XIC is *completed*, the XIC's unaccepted edges are culled from the preliminary cluster (Figure 8). A culled edge represents two XICs that are unlikely to be adjacent. Culling a completed XIC's edges removes one or more unlikely XIC combinations, and prevents the completed XIC from receiving any more accepted edges. An XIC in a cluster is deemed complete if it meets one of three conditions:

1. Has the maximum *m/z* among non-complete XICs and has an accepted edge of lesser *m/z*.
2. Has the minimum *m/z* among non-complete XICs and has an accepted edge of greater *m/z*.
3. Has two accepted edges, one in either *m/z*-direction.

Satisfying any of the above conditions confirms that the XIC has acquired its maximum number of accepted edges, and each of the XIC's unaccepted edges are to be culled. Iteration proceeds until no more edges remain in the preliminary cluster (Figure 8D).

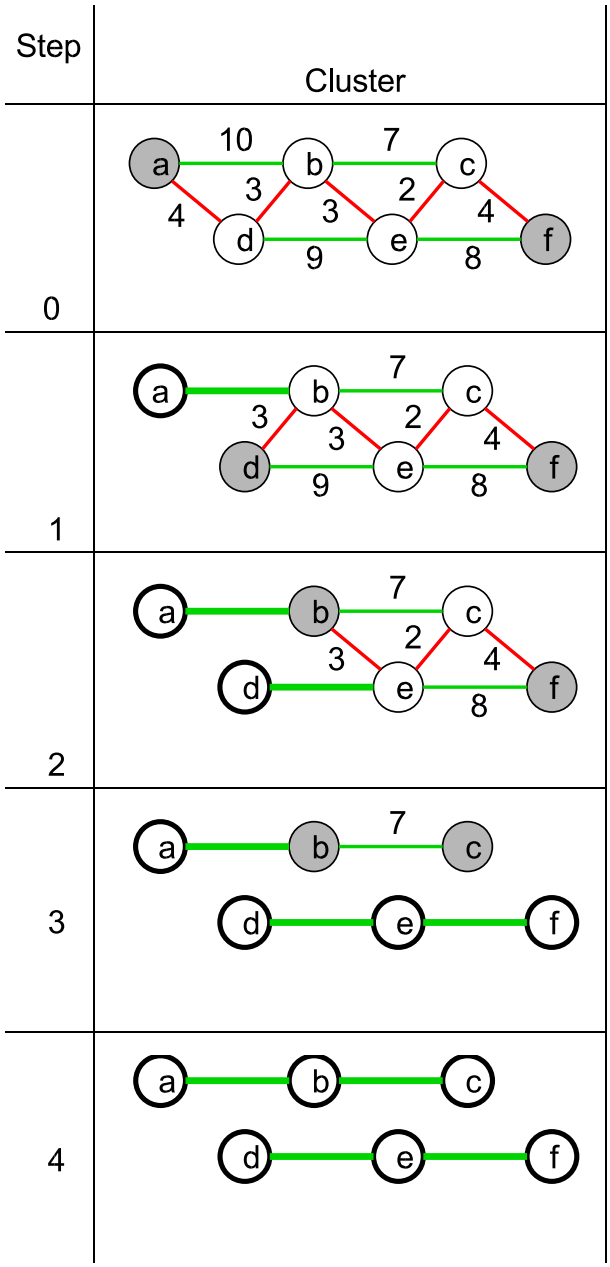

Figure 8: Culling of a preliminary cluster containing two isotopic envelopes with XICs (represented as nodes) {a, b, c} and {d, e, f}. At each step, the edge with the highest edge score is accepted (shown as bold edges). If an XIC is complete (shown as bold nodes), all connected edges are culled. Shaded nodes represent non-complete minimum/maximum *m/z* XICs.

In addition, iterated edges are culled if they create a double adjacency for any XIC. A double adjacency is when an XIC has two accepted edges in an *m/z*-direction. Double

adjacencies are disallowed because an XIC cannot have two adjacent XICs in one m/zdirection within an isotopic envelope. In a double adjacency scenario, the higher scoring edge will be collected by virtue of the descending order of iteration.

Edge Cases

The preceding edge culling algorithm does not capture edge cases satisfying each of the following criteria:

1. The envelope has charge state z_0 and there exists another charge state z_1 and an integer n such that $z_0 = nz_1$.

2. The envelope has XICs x_0 and x_1 with an m/z-separation of $1/z_0$ and correlation c_0.

3. The envelope has XIC x_2 where x_0 and x_2 have an m/z-separation of $1/z_1$ and correlation c_1.

4. $c_0 < c_1$

An example edge case is provided in Figure 9, with $z_0 = 4$, $z_1 = 2$, and $n = 2$. Nonadjacent XICs x_0 and x_2 score higher than adjacent pairs (x_0,x_1) and (x_1,x_2). Figure 9A shows the result of culling on this particular cluster: x_1 is excluded from the resultant envelope.

Modifications to the edge scoring step incorporate these edge cases. First, the precision of edge scores is deliberately reduced from the thousandth to the tenth by rounding to the nearest tenth. The score is multiplied by ten for readability. As a result, each edge's score is now in the set of integers $\{0..10\}$. Obviously, the loss in precision results in many edge score ties (e.g. all edges score 10 after score rounding in Figure 9B). Ties are arbitrated in favor of edges with a lesser m/z-separation.

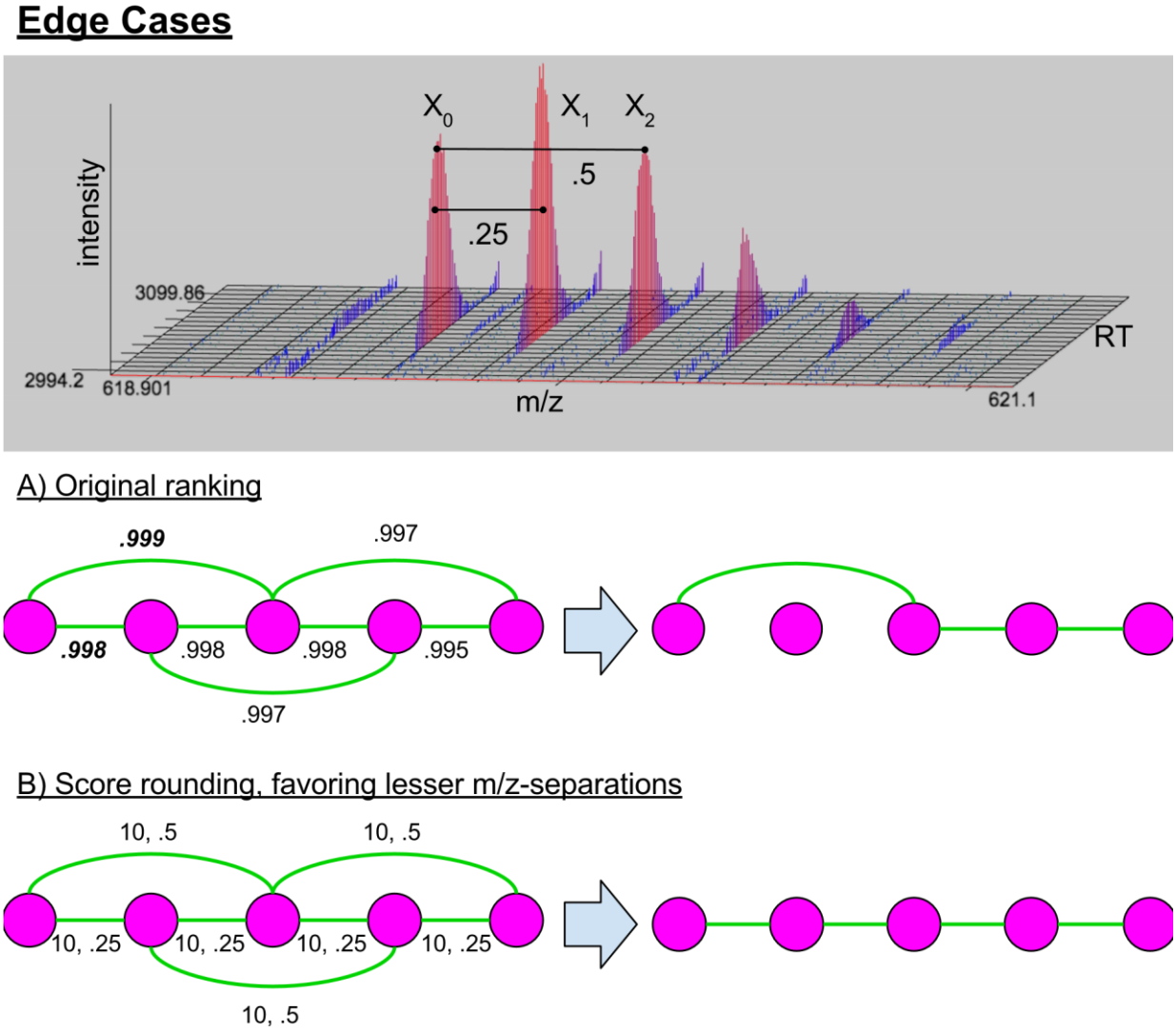

Figure 9: An edge case requiring score rounding and *m/z*-separation favoring. XICs X_0 and X_1 have a valid *m/z*-separation (0.25). X_0 and X_2 also have a valid *m/z*-separation (0.5). X_0 and X_2 have a stronger correlation, therefore a higher edge score. (A) Using the original ranking results in an incorrect cluster. (B) With score rounding and favoring lesser *m/z*-separations, the correct cluster is achieved.

Step 4: Consistency

The final step in XNet is to ensure that all culled clusters are consistent with isotopic envelope properties 2 and 3. Culling is designed to dissect preliminary clusters that contain overlapping envelopes. However, due to the circumstantial alignment of XIC neighborhoods, it is possible for preliminary clusters to contain a chain of two or more isotopic envelopes that are

within 1.1 Daltons on the *m/z* axis and do not overlap (Figure 10 A, D). In such cases, it is possible to identify separations inconsistent with isotopic envelope properties 2 and 3. That is, if a culled cluster does not have consistent *m/z*-separation (violating property 2) or has discordant XIC emergence (violating property 3) then the culled cluster contains two or more envelopes.

Consistency analysis is performed on each culled cluster to detect and correct instances of nearby, non-overlapping envelopes. First, *m/z-separation analysis* (Figure 10B) is performed by iterating through the cluster, ensuring that each XIC-separation matches the previous (initialized by the first XIC-separation). If an XIC-separation is encountered that does not match the previous, then the cluster is split at the edge that presented the inconsistent separation (Figure 10B). After a split, the next XIC-separation re-initializes the process.

After *m/z*-separation analysis, each culled cluster is subjected to *apex analysis* (Figure 10E). The apex of an XIC is the most intense point in the XIC. Apex analysis enforces isotopic envelope property 3 (concurrent emergence) without employing arbitrary thresholds via a single criterion: within an isotopic envelope, each XIC's apex must fall within the RT-range of all previous XICs. Due to transitivity, the criterion can be restated: each XIC's apex must fall within the RT-range of the smallest (in terms of RT) previous XIC. Apex analysis is performed by iterating through the cluster, ensuring that each XIC's apex is within the RT-range of smallest, previous XIC (initialized by the first XIC). If an XIC's apex escapes the constraining RT-range, then the cluster is split at the edge between the escaping XIC and the previous XIC (Figure 10E).

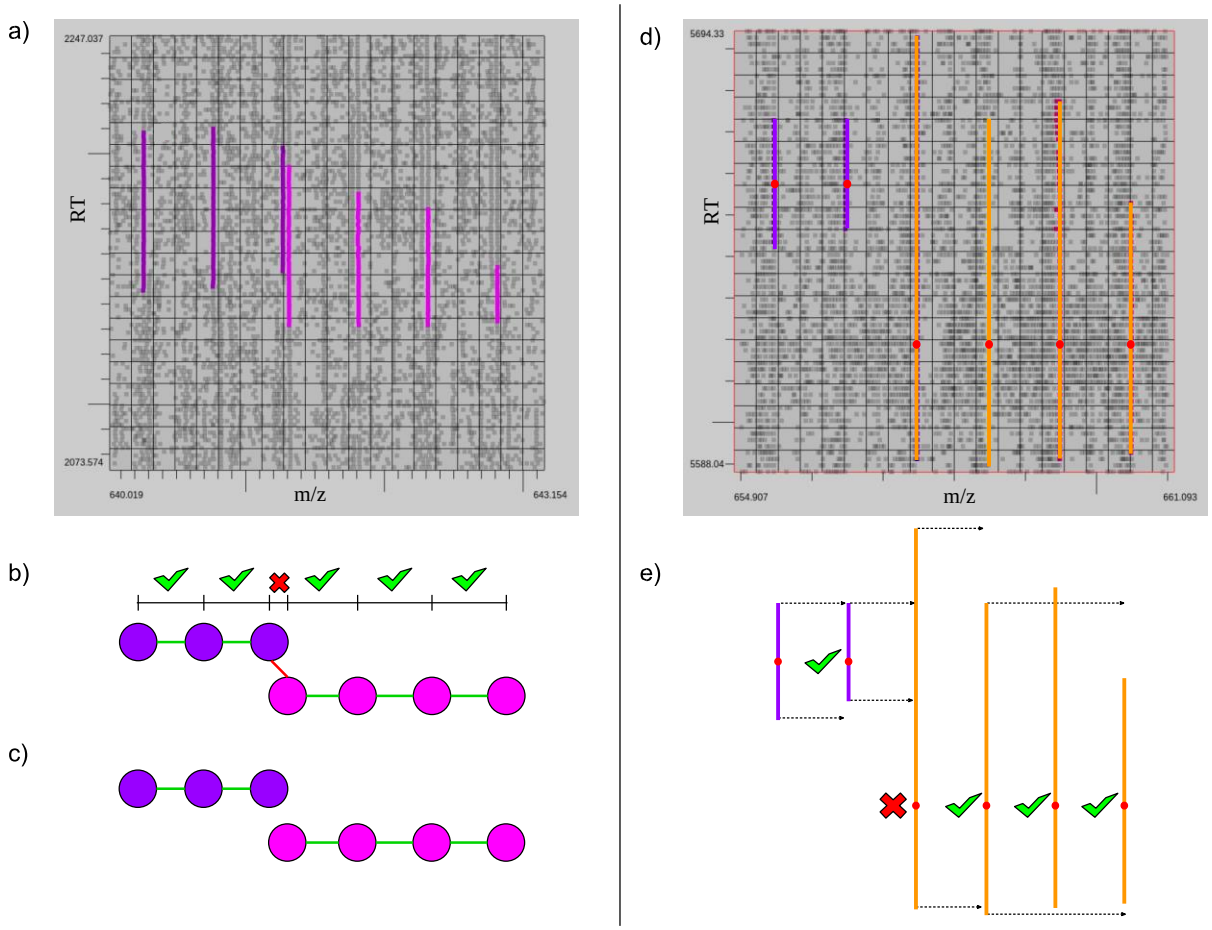

Figure 10: The two cases of culled clusters that are inconsistent with the properties of isotopic envelopes. (A) exhibits nearby, non-overlapping, apex-consistent envelopes composing a single culled cluster (B). *m/z*-separation analysis (B) results in splitting the culled cluster into the two true isotopic envelopes (C). (D) exhibits nearby, non-overlapping, *m/z*-separation consistent envelopes composing a single culled cluster. Apex analysis (E) results in the two true isotopic envelopes.

Results

A hand-labelled version of the UPS2[14] dataset, containing fully annotated ground truth data on 1776 isotopic envelopes comprising 6682 XICs was used for quantitative evaluation. XNet was compared with the XIC clustering modules of MaxQuant,[9] msInspect[10] and FFC[7] in terms of XIC clustering efficacy. XNet was evaluated once for each probability model– Bayesian, Frequentist and Hybrid. SuperHIRN,[8] MzMine[16] and Hardklor were considered for

evaluation; however, SuperHIRN was discontinued, and both MzMine and Hardklor are too involved to be considered automated.

First, each module was evaluated on accuracy of XIC clustering. In this context, accuracy is defined as:

$$Accuracy = \frac{Number\ of\ correctly\ clustered\ XIC_T}{Total\ number\ of\ XIC_T} \tag{8}$$

An important consideration in evaluating XIC clustering is efficacy across various magnitudes of XIC intensity. In many contexts (such as biomarker discovery), low-intensity signals tend to be the most significant. Due to tenuous signal strength and rarity, these signals also tend to be the most difficult to accurately segment. The clustering accuracy of each module across several orders of XIC intensity magnitude was evaluated individually to stratify performance by intensity. Overall accuracy was additionally recorded.

The XIC clustering modules of MaxQuant and FFC both present a number of user parameters that must be set before executing clustering, whereas XNet and MsInspect are free from parameters. Many of MaxQuant and FFC's parameters perform as hard thresholds that control program decisions. While tunable user parameters allow for optimization, it is unrealistic to expect a user to optimize parameters.[11] In most cases, a user will rely on default settings,[11] which are very unlikely to be optimal. If a user decides to attempt manual configuration of user parameters, the optimal value is generally unknown and difficult to derive.[11] In either case, severe performance degradation can result from sub-optimal configurations[11].

A set of configurations were evaluated for both MaxQuant and FFC to discern the impact of sub-optimal parameter settings. MaxQuant's XIC module has 2 integer and 3 continuous user parameters. FFC has 24 user parameters total: 12 integer, 10 continuous, and 2 nominal. Integer parameters were tested on a range from 0 to double the default value (i.e. +/- 100% of the default). Continuous user parameters were tested on a range of 5 values. Each range spanned from 0 to double the default value. Both integer and continuous parameter ranges were bounded by any provided minimum/maximum constraints. Nominal parameters were tested on all provided values.

The resulting set of configurations for MaxQuant contained 3500 configurations, each of which was tested. The resulting set of configurations for FFC is vast, however, with

approximately 10^{25} configurations. Evaluation of one FFC execution requires on the order of an hour to complete, and so thoroughly evaluating FFC's configuration set is intractable. To compensate, 80 randomly selected configurations were chosen from the configuration set and tested.

This section relies on a number of terms describing similar entities, repeated frequently. For clarity and brevity, the following acronyms will be used in reference:

1. ***IE_R* (Resultant Isotopic Envelope)**: An isotopic envelope resulting from the completed XIC clustering process.

2. ***IE_T* (True Isotopic Envelope)**: An isotopic envelope existing and segmented within the fully annotated ground truth dataset.

3. ***XIC_T* (True XIC)**: An XIC existing and segmented within the fully annotated ground truth dataset.

Assessing the number of correctly clustered XIC_T is not trivial. With inaccuracies expected, an IE_R might not match any IE_T exactly, and it might contain XIC_T from multiple IE_T (see Figure 11). Each IE_T must be paired with an IE_R that best represents it. Then, each XIC_T within an IE_R can be assessed by comparing its latent IE_T to the IE_R's paired IE_T. An XIC_T is considered correctly cluster if it's IE_R is paired with IE_T, otherwise the XIC_T is incorrectly clustered.

Pairing IE_T to IE_R is a matter of majorities. For each IE_R, each contained XIC_T contribute a vote for its IE_T. The IE_R is paired with the elected IE_T. It is possible for multiple IE_R to attempt to pair with the same IE_T, however an IE_T cannot pair with more than one IE_R; ties are settled in favor of the IE_R with more votes for the contended IE_T. The conceding IE_R is unpaired. Any XIC_T contained in an unpaired IE_R are considered incorrectly clustered (IE_R 3, Figure 11).

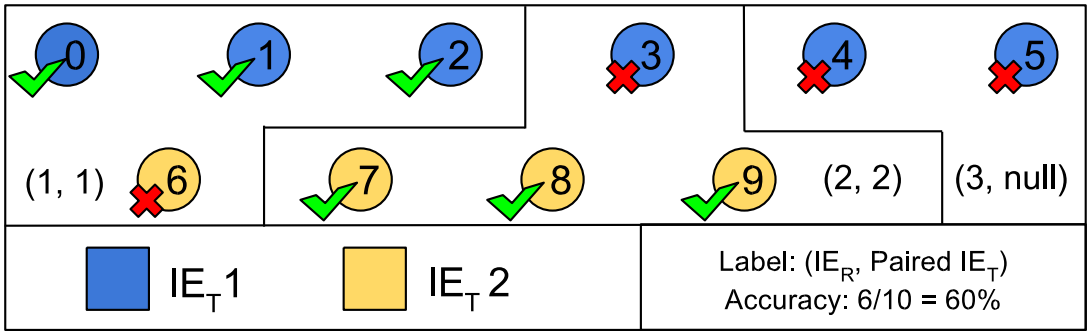

Figure 11: Evaluation of a hypothetical XIC clustering scenario with two ground truth isotopic envelopes (IE_T) {0, 1, 2, 3, 4, 5} and {6, 7, 8, 9}. Each partition represents a resultant isotopic envelope (IE_R) assigned by a clustering module. Each IE_R pairs with an IE_T by majority vote by its XIC_T (shown as nodes). IE_R 3 has a majority IE_T 1, however IE_R 1 has more votes for IE_T 1; IE_R 3 is left unpaired (denoted as null). An XIC_T is correctly clustered if its IE_R is paired with its IE_T. The resulting accuracy is 60%.

Evaluation of XNet, MaxQuant and msInspect concentrated solely on each software package's XIC clustering module. Each module was given as input all XIC_T within the fully annotated ground truth dataset so that the resulting IE_R could be evaluated against the IE_T using the above procedure.

OpenMS FFC does not employ a modular approach to automated signal segmentation; there is no XIC clustering module where XIC_T could be inputted. Instead, the entire unlabelled UPS2 dataset had to be inputted into OpenMS FFC. The result is a featureXML file containing a set of IE_R, each comprising a set of resultant XIC (XIC_R). In order to evaluate this result, each XIC_T must be paired with an XIC_R. This pairing assigns each XIC_T to an IE_R, each of which can be evaluated using the procedure described above.

Pairing an XIC_T with an XIC_R entails searching for the closest matching XIC_R. The following match metric was designed to determine match quality.

$$XIC\ Match\ Quality = \frac{\%\ RToverlap}{|m/z_1 - m/z_2|} \tag{9}$$

The match quality metric promotes XIC pairs that show high overlap in the RT dimension and nearness in the *m/z* dimension. Each XIC_T is paired with the XIC_R with the highest match quality.

If there is contention over an XIC_R, the contest is resolved in favor of the XIC_T with higher match quality. The conceding XIC_T is left unpaired. From here standard clustering assessment resumes, with one minor difference: XIC_T with a null pair are considered incorrect.

All computer resources were dedicated when performing comparisons. Hardware configuration: Dell XPS 8900, 8-processor Intel Core i7-6700K CPU @ 4.00GHz, 256GB SSD, Xubuntu 16.04 (all evaluations except MaxQuant, performed on Windows 10), 32GB RAM.

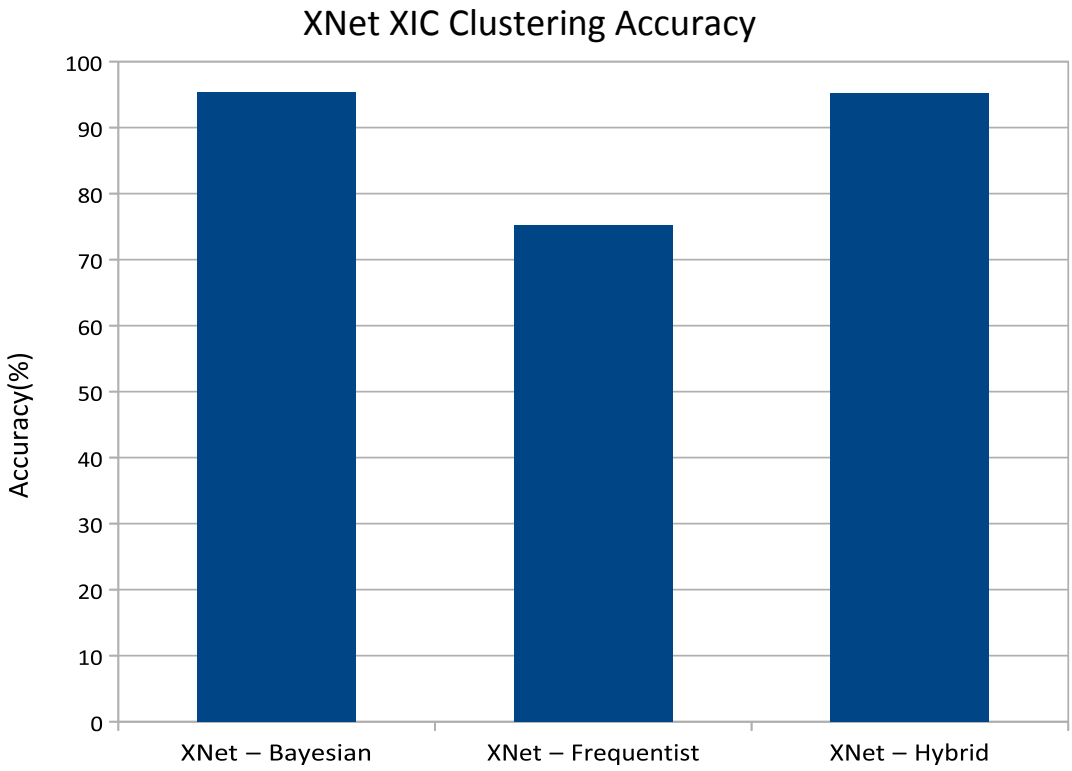

Figure 12: XIC clustering accuracy for each of XNet's probability models. The Bayesian model scores the highest at 95.3%, followed closely by the hybrid model at 95.2%, and finally the frequentist model measures 75.2% accuracy.

Figure 12 displays the overall XIC clustering accuracy for XNet using each probability model (Bayesian, frequentist, hybrid). The Bayesian probability model scored the highest at 95.3%, followed closely by the Hybrid model at 95.2%. The frequentist model is less effective, recording an accuracy of 75.2%. The Bayesian model proved to be the most effective probability model with the available quantity of fully annotated ground truth.

Figure 13 displays the number of correctly clustered XICs per order of XIC intensity. Each data point for MaxQuant and OpenMS FFC represent a different user parameter configuration. The XNet measurements presented represent the top-performing Bayesian probability model.

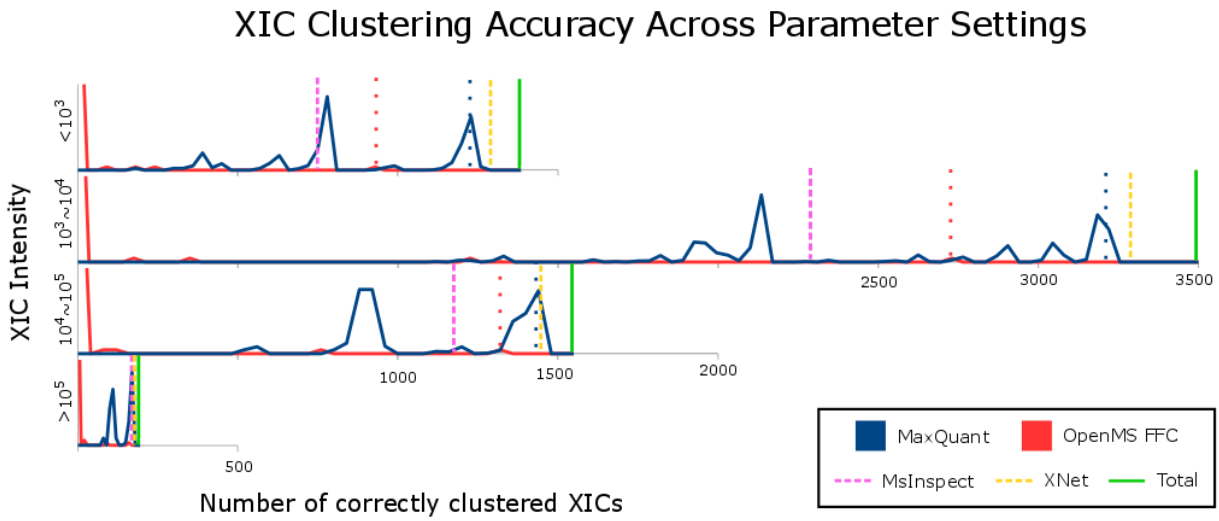

Figure 13: XIC clustering accuracy across orders of XIC intensity, evaluated with modulated parameter settings. The total number of XICs per order of intensity is shown by a solid vertical line. MaxQuant and OpenMS FFC have many user parameters, plots for either software are histograms over clustering accuracy. MaxQuant and OpenMS FFC show a wide range of accuracies across all configurations, with each software's default configuration accuracy shown by dotted lines. XNet and MsInspect do not have user parameters; only a single configuration can be evaluated per software, represented by dashed vertical lines. XNet outperforms OpenMS FFC, MsInspect, and nearly all configurations of MaxQuant.

There are two observations to behold in Figure 13. First, regardless of order of intensity, XNet consistently outperforms MsInspect, XNet outperforms FFC under all configurations, and MaxQuant under almost all configurations. Second, user parameter settings play a major role in determining XIC clustering performance. While MaxQuant can be configured to perform at upwards of 90%, misconfiguring MaxQuant can lead to accuracies below 30%. FFC suffers from user parameters more dramatically. Most configurations resulted in 0 correctly clustered XICs. The maximum recorded overall accuracy for OpenMS FFC occurred once at 77%.

Future Work

Currently, XNet is capable of providing client applications with a confidence metric on each resultant XIC cluster; however, this functionality was not prioritized. Depending on the intent of the confidence metric, it could be calculated as the average edge score, minimum edge score, sum of edge scores, or other collective formula. If more specificity was desired, each edge could be returned with its score. With both a collective cluster score and edge scores, suspect clusters could be manually inspected with edge scores highlighting low-confidence edges.

The notion of manual feedback on a subset of instances is not novel, it is a concept know as active learning.[17] In machine learning, the active learning technique is one in which a machine learning model queries an oracle (usually a human) for a label on selected instances.[17] The results of the query can then be used to improve the machine learning model. XNet is a prime candidate for active learning equipped with uncertainty sampling,[17] where instances with the least certainty are selected for query. Using the edge confidence metric described above, XNet could iteratively improve the frequentist or hybrid probability models, both general and domain-specific. Training on low confidence (or certainty) instances alleviates the difficulty of obtaining ground truth data, while maintaining a schedule for improvement.[17]

Discussion and Conclusion

XNet is a machine learning approach to XIC clustering based on a Bayesian network. XNet is designed around the latent properties of isotopic envelopes to capture the statistical propensity of isotopic envelope composition. This propensity is modelled in three ways. The first model is constructed in accordance with Bayesian probability theory, where reasonable expectations determine likely outcomes. Next, fully annotated ground truth data populates the frequentist probability theory approach, using observed outcomes to determine likelihood. Finally, a hybrid of the two allows for the frequentist model to be initialized with the Bayesian model, such that the Bayesian model can be fine-tuned.

XNet is the first XIC clustering module based on a trainable machine learning model. The intended result is that XNet can, and will, improve as more fully annotated ground truth data becomes available. Upon acquiring and training on additional fully annotated ground truth data, XNet's statistical understanding of XIC clustering will improve. We anticipate that given enough

ground truth MS segmentation data, XNet's frequentist or hybrid probability model will surpass the Bayesian probability model in terms of XIC clustering performance.

XNet can leverage this adaptability in order to specialize to specific domains. If it were to appear beneficial, a multitude of probability files could be developed, each with a domain of aptitude. The advent of a fully annotated ground truth dataset would train the probability file corresponding to the dataset's domain, and could additionally contribute to a general probability file. The dynamic nature of a machine learning approach allows for growth in applicability that cannot be achieved by a static design.

XNet does not employ hard thresholds. XNet's internal parameters are limited, based on the properties of isotopic envelopes, and data-invariant. XICGrid's cell width is based on isotopic envelope principle 1, and the cell height does not affect clustering performance. The reasonably expected CPTs in the Bayesian probability model are crafted by the properties of isotopic envelopes, and can be replaced by CPTs observed by ground truth data. XNet is averse to static constants and configurations, and where they must be used they are data-invariant.

XNet with the untrained Bayesian probability model performs comparably to MaxQuant under optimized user parameters, both of which are the top-performing XIC clustering modules. XNet is distinguished from MaxQuant because its efficacy will translate into the real world. Since XNet is essentially parameterless–the only parameter is the choice of probability model, and the Bayesian model should remain selected–the high accuracy recorded herein will translate automatically to further experimentation. The performance recorded for MaxQuant, and other parameter-laden modules, will not automatically translate to the real world. We've demonstrated the catastrophic effect that sub-optimal parameters can have on performance, and users are very unlikely to use optimal settings.[11] XNet's performance is noteworthy, even before considering its consistency.

References

(1) Mann, M. *Nature reviews Molecular cell biology* 2006, *7*, 952–958.

(2) Wiese, S.; Reidegeld, K. A.; Meyer, H. E.; Warscheid, B. *Proteomics* 2007, *7*, 340–350.

(3) Horvatovich, P.; Mischoff, R. *European Journal of Mass Spectrometry* 2009, *16*, 101.

(4) Nesvizhskii, A. I.; Vitek, O.; Aebersold, R. *Nature methods* 2007, *4*, 787–797.

(5) Michalski, A.; Cox, J.; Mann, M. *Journal of Proteome Research* 2011, *10*, 1785–1793.

(6) Röst, H. L.; Schmitt, U.; Aebersold, R.; Malmström, L. *PloS one* 2015, *10*, e0125108.

(7) Bertsch, A.; Gröpl, C.; Reinert, K.; Kohlbacher, O. *Data Mining in Proteomics: From Standards to Applications* 2011, 353–367.

(8) Mueller, L. N.; Rinner, O.; Schmidt, A.; Letarte, S.; Bodenmiller, B.; Brusniak, M.-Y.; Vitek, O.; Aebersold, R.; Müller, M. *Proteomics* 2007, *7*, 3470–3480.

(9) Cox, J.; Mann, M. *Nature biotechnology* 2008, *26*, 1367–1372.

(10) Bellew, M.; Coram, M.; Fitzgibbon, M.; Igra, M.; Randolph, T.; Wang, P.; May, D.; Eng, J.; Fang, R.; Lin, C. *Bioinformatics* 2006, *22*, 1902–1909.

(11) Gatto, L.; Hansen, K. D.; Hoopmann, M. R.; Hermjakob, H.; Kohlbacher, O.; Beyer, A. *Journal of proteome research* 2015, *15*, 809–814.

(12) Guo, H.; Hsu, W. A survey of algorithms for real-time Bayesian network inference. AAAI/KDD/UAI02 Joint Workshop on Real-Time Decision Support and Diagnosis Systems. 2002.

(13) Rish, I.; Singh, M. *IBM Watson Research Center* 2000,

(14) Tsou, C.-C.; Avtonomov, D.; Larsen, B.; Tucholska, M.; Choi, H.; Gingras, A.-C.;

Nesvizhskii, A. I. *Nature methods* 2015, *12*, 258–264.

(15) Dahl, G. E.; Sainath, T. N.; Hinton, G. E. Improving deep neural networks for LVCSR using rectified linear units and dropout. Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. 2013; pp 8609–8613.

(16) Katajamaa, M.; Miettinen, J.; Orešič, M. *Bioinformatics* 2006, *22*, 634–636.

(17) Settles, B. *University of Wisconsin, Madison* 2010, *52*, 11.