University of Maryland, College Park

AMSC 663/664

Pattern Decomposition of Inorganic Materials: Optimizing Computational Algorithm

Author:
Graham Antoszewski
ganto@math.umd.edu

Supervisor:
Dr. Hector Corrada-Bravo
Center for Bioinformatics and Computational Biology
University of Maryland, Department of Computer Science
hcorrada@umiacs.umd.edu

December 16, 2016

Abstract

Phase pattern decomposition of inorganic materials' crystalline structure is extremely important for the unearthing of new properties such as superconductivity. Previously, this process had meticulously been done by hand, so computer algorithms have been developed to try and uncover these phases. They, however, have yet to combine efficiency and accuracy together. The goal of this project is to do just that by extending the Graph-based endmember extraction and labeling algorithm (GRENDEL). Phase one will be to incorporate physical constraints and prior knowledge to increase the accuracy of our phase composition results, and phase two will be to utilize active learning to minimize the number of sample points needed to analyze a given material to increase efficiency.

1 Background Information

Inorganic materials are compounds or mixtures of elements which do not contain any carbon. Of particular interest are combinations of metal alloys called ternary systems, where three different metallic compounds are heated up and combined into one. Because of the heating and cooling process, the crystalline structure of each individual metal has been altered, similar to how an ice cube that is melted and refrozen will not be identical to the initial configuration. This means the phase of the metal has changed, as the phase is defined as a region within a material or compound where the crystal structure and composition is uniform [1]. This means these phases have distinct properties, such as density and index of refraction. Within different areas of the ternary alloy, there can be different phases of each metal as well due to how the atoms restructured and the proportions of each compound at the given point. Each point of this material is made of a different composition of the three input metals, meaning there can be three phases present and at different proportions based on the mixing process of the alloy. An example of a typical thin film sample of a ternary system is seen in Figure 1.

A given phase of a metal, as previously stated, has distinct properties, one of these being a unique diffraction pattern. X-ray diffraction is used to probe a given material, sending in beams of electrons and observing the outgoing spectra [1]. X-ray light has a wavelength that is approximately the same as the distance between atoms in a crystal lattice, giving it a better chance to hit the atoms within the structure. The light will hit an electron in the metal, absorb energy, and bounce back at a given angle. Note that this energy exchange only happens at certain incident angles, which is dictated by the Bragg equation,

$$2dsin\theta = n\lambda, \tag{1}$$

where d is the distance between atoms in the lattice, θ is the incoming scattering angle, λ is the wavelength of the x-ray, and n is an integer. The absorbed energy is seen as diffraction peaks at the given angles which satisfy equation (1). Note that since λ is fixed, the only variable which determined the angle θ is the lattice spacing d.

But for an unknown phase, we do not know the distance d. Thus, both the source of x-ray light and the detector rotate in order to record data over all possible scattering angles $2\theta \in [0°, 90°]$ (2θ is defined as the angle between the detector and the incident beam rather

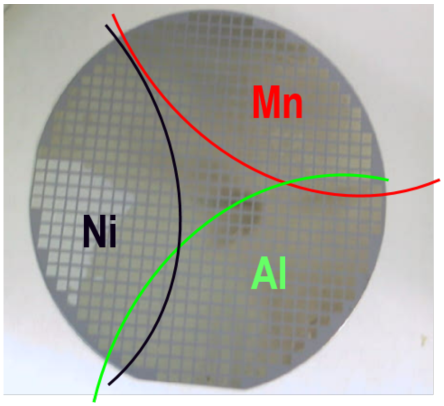

Figure 1: Seen is a thin film of an Al-Mn-Ni ternary system, with the regions specified by each color being the predominant areas of each of the composite metals. That is, where each metal was initially introduced into the alloy and then mixed. This highlights how the mixing throughout the material is not uniform, as we want to find all possible combinations of constituent phases [2].

than the plane of the material, and will be twice that of θ according to Equation (1)). Figure 2 shows an example of an x-ray spectrum for a single sampled point in a material. The given pattern is called a waveform, with peaks in the diffraction spectrum indicating a detection of a certain phase.

The are three main aspects of a given diffraction peak. The scattering angle 2θ tells us about the metal observed and its particular phase, while the height and width of a given peak can tell us about the given proportion of that phase in the mixture of metals that make up the alloy at the sampled point. One can also notice a shifting of the position of seen peaks over different light intensities, which is a source of error and something that has to be accounted for. Using this data, we can recognize the constituent phases seen at each point in the material along with their respective proportions, and a phase diagram can be made like the one seen in Figure 3 [1]. The alloyed material we wish to sample is usually on a circular thin film, yet we transform the data taken from this shape into a simplex, where each vertex corresponding to the locations of the three initial compounds at the start of the mixing process. Different colors represent areas/clusters within the material where similar phase structure is seen, indicating these whole regions will have the same intrinsic chemical properties. Each dot or marker on the simplex corresponds to a probed sample point.

2 Project Objective

Previously, these phase diagrams were done by hand, eying the proportions of the constituent phase composition. This process took so long that a library of materials has already been created which have yet to be analyzed. Thus, the White House Materials Genome Initiative was started in order to encourage an algorithm to take in this structure and composition data as an input and output the desired phase diagrams and phase composition data (more information available at *https://www.whitehouse.gov/mgi*. This algorithm must obey the

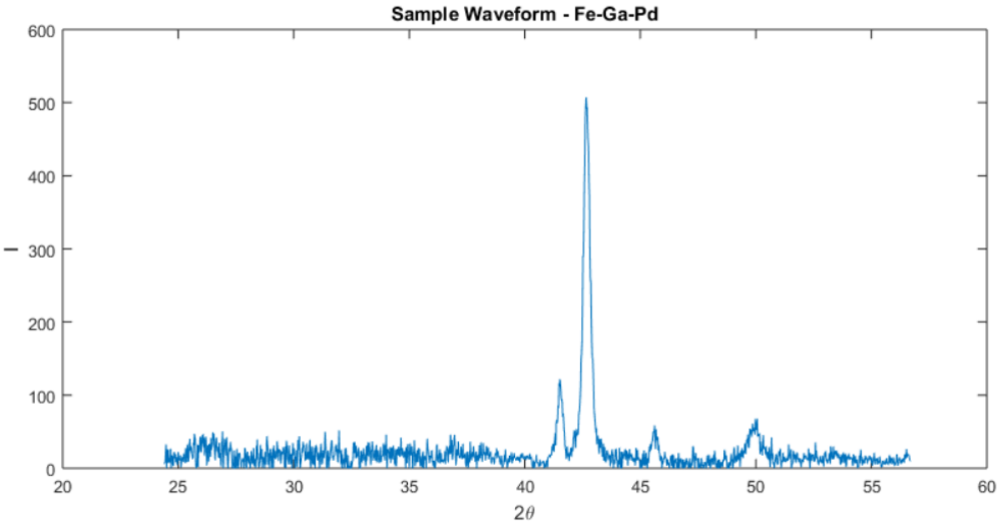

Figure 2: A sample x-ray spectrum from the Fe-Ga-Pd ternary system, with the x-axis being 2θ = the scattering angle observed and the y-axis being the intensity of light detected. Peaks on this plot represent material detection corresponding to given phases of our metals.

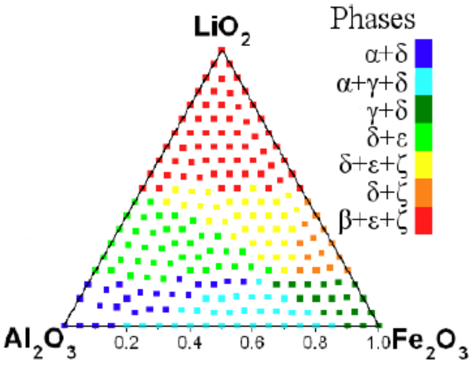

Figure 3: An example of a phase diagram, represented as a simplex. Each vertex corresponds to one of the original compounds in the alloy, colors correspond to similar phase structure between those points, and the Greek symbols in the legend represent the different phases seen in the material [1].

laws of physics while also accurately identifying the individual phases, regions or clusters of similar phase composition, and do all of this in an efficient manner so more materials can be evaluated [3].

Current attempts at an algorithm focus on pattern decomposition. Given a set of diffraction patterns at N points of a given system, it is assumed these can be described as a combination of K basis patterns. We seek to resolve these basis patterns, which in this case are the K constituent phases that contribute to the diffraction patterns seen in the material. In other words, if we think of the entire material's diffraction spectrum as a vector space, we wish to find the basis vectors of the space. There are two main steps, the first being spectral clustering. Here a similarity matrix is constructed to group points in the material with analogous diffraction patterns, with each group of points being called a *cluster*. This splits up our entire dataset into smaller subproblems, allowing the algorithm to run more efficiently. Second, nonnegative matrix factorization is used to identify the constituent phases and their proportions within each cluster [1]. These steps will be explained in detail in Section 3.

One example of such an algorithm is Graph-based endmember extraction and labeling (GRENDEL). *Endmember* is another word for the basis constituent phase which make up the diffraction pattern of a given point or region within the material, so both terms can be used interchangeably. This method seeks to minimize an objective function during the pattern decomposition process, which looks at how well our estimated phase proportions match up with the raw diffraction patterns both within the clusters and over the entire material. GRENDEL runs very fast, with computation times under a minute, but fails to properly take into account physical constraints which leads to inaccuracy [3]. Another attempt at an algorithm, Alternating Mixed Integer Quadratic Optimization, uses mixed integer quadratic problems to minimize the error. Essentially, this boils down to minimizing norms regarding residuals between the original structure and our hypothesized phase structure, yet this method uses prior knowledge to add in physical constraints. It recognizes certain pairs of points and phases that Must-Link and Cannot Link together, which leads to extremely accurate results [4]. AMIQO runs on the order of days, however, making it too slow for an ideal method. The latest attempt of a pattern decomposition algorithm is AgileFD, which relies on convolutive nonnegative matrix factorization, physical constraints, and lightweight update rules of the basis phases derived from the Kullback-Leibler divergence loss function to obtain accurate estimates of the basis phases in an efficient manner [5],[6]. Yet AgileFD omits the clustering step of isolating regions of similar phase structure, so we wish to combine the accuracy of AgileFD with the speed and clustering of GRENDEL.

Furthermore, it takes about a half hour to obtain a full x-ray diffraction spectrum for each point in a material, meaning it can take over a week to go through an entire material. Another goal then must be to find a way to have our algorithm use a minimal number of data points to evaluate an entire material, as an algorithm that runs quickly does no good if the input data cannot be collected quickly too. This requires a program that, rather than taking entire material's spectrum as input, iteratively takes in individual sample points and suggests the most informative point in the material to be the next to be probed.

In summary, current approaches at an algorithm are missing a key goal of the Whit House Materials Genome Initiative. In addition, the sampling process of retrieving new x-ray diffraction data takes too long so even if our program runs quickly, we still cannot do rapid analysis of new materials. Our project objective is to address these issues in one algorithm to combine speed, accuracy, and an optimized sampling process. To do so, will be working to extend the GRENDEL algorithm. GRENDEL begins with a spectral clustering step in order to create initial cluster assignments for all of our sample points within our given material dataset. Then, an iterative two-step process of nonnegative matrix factorization and the Graph Cut package is run to find a local minimum of the objective function while simultaneously updating cluster assignments for the entire material [7],[9],[8],[10]. Once convergence is attained, GRENDEL will output cluster assignments for each of the sample points within the material as well as a set of constituent basis phases/endmembers for each cluster [3]. From this output, phase diagrams outlining regions of similar phase structure and constituent phase compositions of each cluster can be generated, with an example of a phase diagram seen in Figure 3.

The two new components which we will add to GRENDEL is constraint programming and active learning. Constraints can be added to the objective function, which can be laws of physics or properties of the material that we know prior to computation, to make our solution more accurate and physically realistic. The objective function already has certain constraints, and the challenge of this optimization strategy is uncovering the most important constraints to use, as adding constraints tends to increase the run time of GRENDEL. Further explanation of these new features of GRENDEL are explained in Section 4.1.

Active learning pertains to optimization of the sampling process. Rather than using all sample points of a given material, we now realize that our algorithm will be running in conjunction with the sampling process, meaning the input spectral data will be read in iteratively point by point. We will take a small subset of sample points, run GRENDEL to output the current guess of clusters and phases/endmembers, then generalize this analysis to create a predictive phase diagram for the whole material. To estimate the basis phase composition of unknown regions of the material where we have not probed, we do a distance-weighted average from the results at the data points we do have. The next point to sample is then chosen by determining the area of highest cluster uncertainty, coinciding with this region's estimated basis phase composition conflicting with the cluster compositions output by GRENDEL. This will typically fall at cluster boundaries, as the endmembers of neighboring clusters will differ and overlap to create error [11]. The goal is to probe these areas of uncertainty and skip probing areas that may be superfluous, such as a middle of a large cluster whose basis phase composition is already known, to reduce the number of points we need to achieve minimization of the objective function. The active learning component of the project is to be implemented in the upcoming semester, and thus will not be focused on in this mid-year report. More discussion can be found in Section 8.4. Before going any further into the implementation of the constraint programming done so far, however, the current GRENDEL algorithm must be explained in detail.

3 Algorithm - GRENDEL

Figure 4 illustrates the flow of the current implementation of GRENDEL. As an input, both structure and composition data from the Inorganic Crystal Structure Database and other material libraries can be utilized. If X is the input diffraction waveforms (labeled in Figure 4 as 'structure' data) for the whole material of N sample points, we will look at each individual

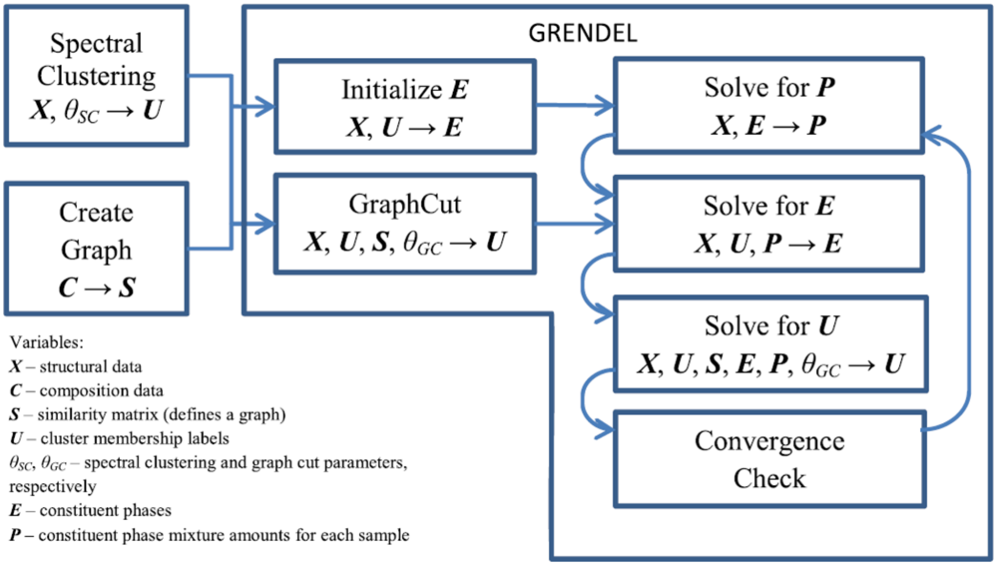
Spectral Clustering
$\boldsymbol{X}, \theta_{SC} \rightarrow \boldsymbol{U}$
Create Graph
$\boldsymbol{C} \rightarrow \boldsymbol{S}$
GRENDEL
Initialize $\boldsymbol{E}$
$\boldsymbol{X}, \boldsymbol{U} \rightarrow \boldsymbol{E}$
GraphCut
$\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{S}, \theta_{GC} \rightarrow \boldsymbol{U}$
Solve for $\boldsymbol{P}$
$\boldsymbol{X}, \boldsymbol{E} \rightarrow \boldsymbol{P}$
Solve for $\boldsymbol{E}$
$\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{P} \rightarrow \boldsymbol{E}$
Solve for $\boldsymbol{U}$
$\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{S}, \boldsymbol{E}, \boldsymbol{P}, \theta_{GC} \rightarrow \boldsymbol{U}$
Convergence Check
Variables:
$\boldsymbol{X}$ – structural data
$\boldsymbol{C}$ – composition data
$\boldsymbol{S}$ – similarity matrix (defines a graph)
$\boldsymbol{U}$ – cluster membership labels
θ_{SC}, θ_{GC} – spectral clustering and graph cut parameters, respectively
$\boldsymbol{E}$ – constituent phases
$\boldsymbol{P}$ – constituent phase mixture amounts for each sample

Figure 4: A flow chart of the GRENDEL algorithm. Initial structure X and composition C data is input, and spectral clustering is done to find areas of similar diffraction spectra. Then, an iterative process of the Graph Cut algorithm and NMF (seen here in the last column of the flow chart) is implemented in order minimize the objective function which resolves cluster boundaries and the constituent phases within each cluster. Final output plots are the phase diagram and constituent phase plot of the basis phase waveforms [3].

sample diffraction waveform X_i. X_i is a vector with dimensions $1 \times D$, where D is the number of scattering angles observed. For example, we know $2\theta \in [0, 90]$, so if data is taken with resolution of $0.1°$, then $D = 900$. A given element $X_{i,j}$ is itself a scattering intensity value seen at the given j^{th} scattering angle by the detector. To graphically see where each of the N sample points are in terms of our phase diagram, each marker seen on the simplex of Figure 3 is a sample point. Composition data C of the material is used in Section 3.2 to place a given sample point i in the correct position on the simplex, whereas the spectral data X_i is used to determine the cluster assignment and basis phase composition of each cluster, seen as the different colors for each of the markers [3].

3.1 Spectral Clustering

Spectral clustering seeks to separate the material data into regions or clusters of similar structure, thus allowing the proceeding steps to be run on smaller subsets to speed up computation. These clusters will also be areas of analogous chemical properties. First, a similarity measure is used to compare how close the diffraction patterns are between two given sample points [12]. This metric is the cosine distance between 2 sample point waveform vectors X_i and X_j, given by

$$\delta_{\cos}(X_i, X_j) = 1 - \cos(X_i, X_j), \tag{2}$$

where $\cos(X_i, X_j)$ is the cosine similarity between the two vectors, defined by

$$\cos(X_i, X_j) = \frac{X_i \cdot X_j}{||X_i||\ ||X_j||}. \tag{3}$$

Here, $\cdot$ is the dot product and $||\cdot||$ is the L2 norm. Thus, the cosine distance between our points X_i and X_j will be near zero if the diffraction patterns match well, near 1 if they are orthogonal, and near 2 if they are completely contradictory [3]. Then, we create a similarity matrix from these cosine distances,

$$W_{ij} = e^{\frac{-\delta_{\cos}(X_i, X_j)}{2\sigma^2}}, \tag{4}$$

where σ is the spectral clustering bandwidth parameter specific to the given material we are observing. Thus, W is a $N \times N$ matrix.

With this, a diagonal matrix G is created by summing the rows of W. Then, eigenvalue decomposition used upon the Graph Laplacian defined by

$$L = G^{-1}W \tag{5}$$

to find the eigenvectors corresponding to the K smallest nontrivial eigenvalues of L. Here, K is an input parameter of how many different clusters we expect to have. This varies from material to material, and is usually determined by the user in conjunction with analyzing results of previous experiments with the materials. For example, it is advised to select the number of clusters K to be similar as the number of basis phases you expect to see, between 5-7.

With these K eigenvectors of length N, we utilize the K-means function within MATLAB to identify individual clusters of points with similar structures. The eigenvectors are assigned to columns of a given matrix Z (so each row of Z corresponds to a data point in the material), and we initialize the clustering by finding the K cluster means (note these are vectors of length K as well). The first is chosen at random from the rows of Z. The other initial means are then chosen from the remaining rows of Z randomly with a probability weighted proportionally to the cosine distance metric between each row and its most-similar cluster mean already selected. We denote the mean initial composition of the k^{th} cluster to be $\bar{v}_k$. With these, an initial clustering assignment can be made for all N points by assigning each row of Z, where the i^{th} row of Z is z_i, to the cluster whose mean composition $\bar{v}$ is most similar by means of cosine distance:

$$C(i) = \underset{1\le k\le K}{\arg\min}\, \delta_{\cos}(z_i, \bar{v}_k), \tag{6}$$

where $C(i)$ is the current clustering assignment of the i^{th} data point. $C(i)$ ranges from 1 to K.

Then, the cluster means $\bar{v}$ are updated by minimizing the total cosine distance sum between the each z_i in the k^{th} cluster C_k and the desired cluster mean composition v_k,

$$\min_{C, \{\bar{v}_k\}_1^K} \sum_{k=1}^{K} N_k \sum_{C(i)=k} \delta_{\cos}(z_i, \bar{v}_k), \tag{7}$$

where N_k is the number of points assigned to the k^{th} cluster currently. This two-step process defined by Equations (6) and (7) is repeated until the cluster assignments no longer change. The final output of this spectral clustering step is the $K \times N$ cluster membership matrix U, where $U_{k,i} = 1$ if the i^{th} data point belongs to the k^{th} cluster [12]. Furthermore, we can also find the mean spectral composition $\bar{X}_k$ of each cluster now by averaging the spectral data of all points within the k^{th} cluster.

3.2 Creating the Simplex

The 'Create Graph' portion the flow chart in Figure 4 references the transformation of the input spatial composition data C for each data point into respective coordinates on the simplex. The simplex is created using the Delaunay tessellation function in MATLAB, then only including edge connections to nearest neighbors of each point. This transforms the circular thin film shape of our structure data into a simplex via triangulation. The vertices of the simplex correspond to the three initial metal compounds used to make the ternary system, meaning points closer to these vertices implies the primary component in the mixture at this point will be this particular compound [3].

3.3 PCOMMEND - Lightweight Nonnegative Matrix Factorization

The main portion of GRENDEL is minimizing the objective function, defined as follows:

$$J(E,P) = \sum_{i=1}^{K} \Big(\sum_{j=1}^{N} u_{ij}(X_j - p_{ij}E_i)^T (X_j - p_{ij}E_i) + \alpha \sum_{h=1}^{M-1} \sum_{l=h+1}^{M} (e_{ih} - e_{il})^T (e_{ih} - e_{il}) \Big) \quad (8)$$

Here, X_i is the spectral data for the i^{th} sample point in the material, K is the number of estimated clusters, N is the number of sample points, and u_{ij} is an element of the binary cluster membership matrix U that is 1 if j^{th} point belongs to cluster number i and 0 otherwise. In addition, M is the number of endmembers (another word for constituent basis phases) in a given cluster, E_i is a $M \times D$ matrix where the rows are the individual basis phase waveforms that make up the set of basis phases of the i^{th} cluster. That is, the h^{th} column of E_i, symbolized as e_{ih}, is the diffraction spectra of the h^{th} phase of the k^{th} cluster. Moreover, p_{ij} is a $1 \times M$ vector of the proportion values for each endmember used for the i^{th} sample point. Thus, p_{ij} is the row vectors of proportion weights for the basis phases of the i^{th} cluster of the j^{th} sample point α is a parameter set to 0.0001 to balance the importance of each of the summations [7].

The key notion here is that we are assuming that the input diffraction data at each sample point, X_i, can be approximated as a linear combination of the basis phases. Thus, p_{ij} are the set of proportion weights applied to the basis phases E_i to make up this combination. That is, represented as linear combinations of the basis phase patterns we wish to find:

$$X_j \approx p_{i,j}\ E_i. \quad (9)$$

The first summation term in (8) corresponds to the norm of the residual between our input diffraction patterns X and the linear combination we seek to obtain, $p_{ij}E_i$, while the second summation can be thought of as a volume constraint on the basis waveform vectors themselves. As stated previously, the position/scattering angle of the peaks within the waveform are determined by the lattice spacing term d in the Bragg Equation (1). Since d is a fixed value at a given point in the material, the position of the peaks within each basis phase should also be approximately similar. If this is not the case, the second summation will be large, implying error in the estimate of the basis phases.

Note that we need matrices X, U, P, and E. We already have X and an initial guess at U from spectral clustering. We create an initial guess for P by setting all proportions equal to $1/M$, and an initial E is obtained using the *nnmf* function of MATLAB, which outputs a guess at the basis phases themselves for each cluster by seeking to minimize the first summation in Equation (8). This function call takes up CPU time, which is why instead of using this each iteration we apply lightweight update rules for the matrices E and P from the PCOMMEND method of finding the local minimum of the objective function [7]. First we solve $\partial J/\partial E_k = 0$ to update our guess for the basis phase matrix of the i^{th} cluster, yielding the equation

$$E_i = \Big(\sum_j u_{ij} p_{ij}^T p_{ij} + 2\alpha(MI_{M\times M} - 1_{M\times M})\Big)^{-1}\Big(\sum_i u_{ij} p_{ij}^T X_j\Big), \tag{10}$$

where I and 1 are the $M \times M$ identity and ones matrices, respectively. We assume endmembers/basis phases must be positive in order to resemble a physically-accurate diffraction pattern, so if an element of E_i is negative, that value is set to zero and the matrix is recomputed via Equation (10).

Second, we try to minimize Equation (8) for p_{ij}. Our proportions of endmembers in cluster i must sum to 1 for each sample point X_j in the cluster in order to be physically realistic as well, $\sum_{h=1}^{M} p_{ijh} = 1$. To ensure this, we use a Lagrange multiplier λ_i. Proportions must be nonnegative as well, so our update of p_{ij} becomes

$$p_{ij} = \max\Big((E_i E_i^T)^{-1}(E_i X_j^T - \lambda_i 1_{M\times 1}), 0\Big) \tag{11}$$

with

$$\lambda_i = \frac{1_{1\times M}(E_i E_i^T)^{-1} E_i X_j^T - 1}{1_{1\times M}(E_i E_i^T)^{-1} 1_{M\times 1}} \tag{12}$$

If a particular proportion value is chosen to be 0 because the first term in Equation (11) is negative, then the other proportions for the i^{th} sample point must be normalized in order to have them sum to one.

These two updates are repeated for all K clusters and over multiple iterations along with Graph Cut, to be explained in Section 3.4, until convergence is reached for matrices E and P, meaning we have minimized our objective function (8) [7]. By finding a local minimum, we understand that the steepest descent-like nature of our updates mean that we cannot guarantee our updating process converges to the absolute minimum of the objective function, only that the objective function is minimized within a certain neighborhood of potential solutions for E and P. Depending on different initial seed guesses at U, P, and E, our PCOMMEND update procedures may converge to different final results, although in our experience these different results vary only slightly.

3.4 Graph Cut Algorithm

Now that we have updated our basis phases E and proportions P, we use Graph Cut to compute the update of these cluster membership matrix U each iteration of GRENDEL. Note that this is completely independent of the objective function (Equation (8)), contrary to what had been asserted in the proposal. The PCOMMEND updates described above also have an update rule for U along with E and P; however, we choose to use Graph Cut to make our cluster membership guess U more accurate. We use a specific MATLAB wrapper available online at *http://www.wisdom.weizmann.ac.il/∼bagon/matlab.html* [3],[9],[8],[10]. The update of U is done by minimizing a cost function, V.

The general cost V of the cluster labeling of all input spectral data $X_i, i \in [1, N]$, is described as:

$$V = \lambda_d \sum_j V^j(L_j) + \lambda_s \sum_{j,l \in N} V^{j,k}(L_j, L_k), \tag{13}$$

where $V^j(L_j)$ is the data cost for a point i, or the cost to assign a cluster label L_j to j, and $V^{j,k}(L_j, L_k)$ is the smoothness cost, or the cost to assign the labels L_j and L_k to the neighboring points j and k. Note that the values of L range from 1 to K, corresponding to the K clusters. λ_d and λ_s are data cost and smoothness cost weights, respectively, which are parameters chosen to balance the smoothness cost, which emphasizes connectivity of clusters so they are all closed regions, and data cost, which emphasizes the similarity of points within a given cluster.

The data cost in Equation (13) is given by

$$V^j(L_j = i) = \frac{3}{4}\delta_{\cos}(X_j, \bar{X}_i) + \frac{1}{4}\frac{||X_j - p_{ij}E_i||_2}{\sum_i ||X_j - p_{ij}E_i||_2}, \tag{14}$$

where $\delta_{\cos}(X_j, \bar{X}_i)$ is the cosine distance between diffraction peaks of sample point j and the mean spectra of the currently assigned cluster $L_j = i$, $|| \cdot ||_2$ is the L2 norm, and E_i and p_{ij} are defined as in Section 3.3. The first terms makes sure that the spectral data (diffraction pattern) of a point X_i matches with the assigned cluster's mean spectra, similar to the spectral clustering step. The second term makes sure that this cluster's basis phase composition correctly represents the sample point's spectral data X_i, similar to the first summation of the objective function in Equation (8).

The smoothness cost $V^{j,k}(L_j, L_k)$ is 0 if points j and k belong to the same cluster and 1 if they do not. While it is not exactly noticeable in Equation (13), the smoothness cost summation is restricted to only neighboring points j and k rather than summing over all possible pairs of points. This makes sense, for if we want smooth and continuous clusters, we expect most of the adjacent data points of sample point i to also be in the same cluster unless it is on a boundary. Adding these two terms together, V is minimized and all sample points are reassigned into the clusters based on this minimized result.

To minimize V, however, we utilize something called the Max Flow Algorithm [9]. This iterates over all K clusters, and looks at all N data points at one time. In one iteration, if we are looking at cluster κ, for each point data point X_j, it looks at the cost of this point being assigned into cluster κ versus its current cluster assignment. Specifically, it takes the *residual* between these two costs, and uses this to determine if it should switch the current

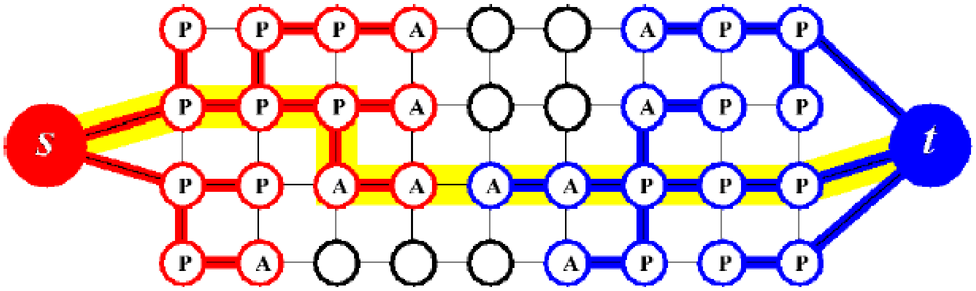

Figure 5: A diagram illustrating the methodology of Graph Cut to update cluster assignments of our sample data points. The red points belong to the source tree corresponding to cluster κ, while the blue points belong to the sink tree corresponding to all other cluster assignments. Note the highlighted path of 'max flow' between the source and sink parent nodes [9].

cluster assignment to cluster κ. By residual, we mean the difference in the total costs (data cost + smoothness cost) between a given point j being in its current cluster assignment versus being assigned to cluster κ. Thus, if the residual for point j is positive, we would say that it is more costly to keep the point in its current cluster, and we should change the assignment of point j into cluster κ. But if the residual is negative, we should keep j in its current cluster.

The novel idea though is to think of the points in cluster κ as belonging to a 'source' tree of flow (positive residuals), and the points remaining in their original cluster assignment as belonging to a 'sink' tree (negative residuals). There must be a continuous path from the highest-level parent nodes of the source and sink tree. This idea is illustrated in Figure 5 [9]. The 'A' and 'P' labels of the points correspond to whether or not this point is a active or passive node in the tree, which is just terminology to say whether o not the node is on the boundary of their respective trees (or the boundary of the cluster itself, thinking about the ternary diagram like in Figure 1). Note, however, that while this does enforce connectivity of cluster assignments of the entire material, there are certain points (seen as the white points in Figure 5) are can potentially been disconnected in terms of cluster assignment. This warrants more connectivity constraints to ensure the laws of physics are obeyed.

As stated before, this is done using the MATLAB wrapper referenced above [3]. This, along with the nonnegative matrix factorization updates in Section 3.3, are repeated over a certain amount of iterations until the convergence of the *condition number*. The condition number for iteration *iter* is just the summation of the norms of the difference between the E, P, and U matrices from iteration *iter* and $iter - 1$, that is, how much our guesses at these three matrices have changed in one update.

One might be noticing in column 2 of the flowchart for GRENDEL (Figure 4), that there seems to be a performance of Graph Cut prior to going into this iterative loop. A simpler version Graph Cut is run prior to this, yet without an initial guess of the matrices E and P. For this run of Graph Cut we only take the data cost matrix to be the cosine distance metric:

$$V^j(L_j = i)_{\text{simple}} = \delta_{\cos}(X_j, \bar{X}_i), \tag{15}$$

with all other aspects of Graph Cut described above remaining unchanged. This is meant to create a better initial guess at U prior to running the nonnegative matrix factorization portion of GRENDEL.

4 Approach to Extend GRENDEL

4.1 Constraint Programming

First, we introduce constraint programming within GRENDEL. Using both the laws of physics and prior knowledge, we will add constraints on certain matrices and variables which will create more accurate results for our experimental results for the basis phases E, its proportion/abundance matrix P, and cluster membership U.

4.1.1 Gibbs' Phase Rule

One example of a physical constraint is the Gibbs phase rule. Our material is considered to be in equilibrium or steady-state. That is, it is not undergoing any chemical processes such as melting or evaporation, and the chemical composition is stable. At equilibrium, a compound or element must be in a set crystalline structure, corresponding to a set phase. Thus, within our ternary system there can only be three phases seen at a given point due to the three input compounds. Every point assigned to a given cluster i should also be represented by the same set of endmember basis phases E_i, so this means that at most 3 phases can be seen in a given cluster [1]. As M is defined as the number of endmembers seen in a given cluster, this constraint is written as

$$M \leq 3. \tag{16}$$

Thus, this law of physics is already applied in GRENDEL by us setting $M = 3$ (so the matrix of basis phases E_i, is $3 \times D$, and p_{ij} is a 1×3 vector) during our updates as defined in Section 3.3. No validation is required, although if our sample material only had 2 input metal compounds, note the M would change to 2 as a result.

4.1.2 Connectivity

Due to the continuity of the mixing and alloying process of creating the ternary system of metal compounds, another law of physics that must be upheld is connectivity of the regions within the material with the same basis phase present. Thus, this entails in GRENDEL that the clusters themselves within the material must be fully connected, as each cluster should compromise of the same 3 basis phases. A way of visualizing this is through mixing colors while painting. If one mixes red and yellow paint loosely together with a paintbrush, knowing that you started with all of the red paint on the left side of the pallet and yellow on the right, one would expect areas of red, orange, and yellow. But one would not see a two regions of red paint completely isolated from each other without at least connecting path of orange paint (a mixture of red and yellow). If you did, then it would mean that these red regions somehow split without leaving some sort of trail between them, which is physically impossible. In that same vein, if we mix three metallic compounds, we expect a continuous distribution/path for each of them throughout the entire alloyed material. The basis phases seen at a given point in the material correspond to these input metal compounds (Hence why Gibbs' phase rule requires us to set $M = 3$ in our algorithm), so we expect connected regions of these basis phases as well.

While the utilization of Graph Cut does a decent job of initially enforcing connected cluster regions, discontinuity of certain clusters begin as we update our E and P matrices through nonnegative matrix factorization. To combat this, we seek to use 'expert' prior knowledge constraints to enforce connectivity of clusters. By prior knowledge, we mean that the constraints used are not exact law of physics, but the methodology utilized makes our results uphold a certain law of physics. We talked about the AMIQO algorithm in Section 2, which applies something called Must-Link and Cannot-Link pairs of data points. Essentially, if the user of the AMIQO algorithm knows prior to analysis that a certain pair of points i and j have spectra X_i and X_j that are in the same cluster, then they say that these two points in the material must be linked together, regardless of what cluster this pair is assigned into. And if we know that a pair of points are not contained in the same cluster, then this pair cannot be linked in the same cluster [4]. Let b_{ij} be a binary variable that is 1 if the j^{th} sample point is in cluster i. The Must-Link and Cannot-Link constraints are written as

$$b_{i,j_n} \in \{0,1\}, i \in [1,K], j_n \in [1,N] \tag{17a}$$

$$b_{i,j_1} = b_{i,j_2} = 1, i \in [1,K], (j_1, j_2) \in MustLink \tag{17b}$$

$$b_{i,j_1} + b_{i,j_2} \leq 1, k \in [1,K], (i,j) \in CannotLink \tag{17c}$$

The issue with this method of constraints is that this requires omniscience regarding cluster assignments of certain data points within the material. Therefore, we append this idea using our own Cannot Link constraints that can be determined from data given in the spectral clustering step described in Section 3.1. Here, we used the cosine distance between two input diffraction waveform vectors, $\delta_{\cos}(X_i, X_j)$, as a similarity metric. To determine which particular pairs of points that cannot be linked together in the same cluster, we assume that the cosine distance between the two waveforms must be large. We assign the top $\rho\%$ of pairs into a Cannot Link array, given by CL (so CL is a $Z \times 2$ matrix, where $Z = \rho\% *$ number of all possible pairs of points), and check to make sure no CL pairs are assigned into the same cluster after the Graph Cut portion in GRENDEL. If they are, whichever point in the pair was the latest to switch into the shared cluster is reverted to the cluster assignment of the previous iteration.

But as stated before, the initial run of the simpler version of Graph Cut creates fully-connected cluster regions, so we take the cluster membership after this step to be correct in terms of knowing if a given pair of points is connected in the same cluster. Thus, we eliminate any of our CL pairs that are actually in the same cluster from the CL array after this initial Graph Cut in order to decrease CPU time.

In summary, to enforce the Cannot Link connectivity constraint, the procedure is described below:

```
Compute $\delta_{\cos}(X_i, X_j) \quad \forall\ (i,j) \in [1, N]$;
Assign top $\rho$% of unique $(i,j)$ pairs into $CL$;
After initial run of Graph Cut:
    for $i = 1 : length(CL)$
        if $CL(i,1)$ and $CL(i,2)$ are in the same cluster
            $CL(i,:) \in deleteCL$;
        end
    end
    $CL(deleteCL,:) = [\ ]$;
$\forall$ successive iterations of Graph Cut
    for $i = 1 : length(CL)$
        if $CL(i,1)$ and $CL(i,2)$ are in the same cluster;
            if point $CL(i,1)$ changed cluster assignment last
                $U(:, CL(i,1)) = U_{old}(:, CL(i,1))$;
            else
                $U(:, CL(i,2)) = U_{old}(:, CL(i,2))$;
            end
        end
    end
```

The parameter ρ was tested for its optimal value, which turned out to be $\rho = 75$. We assumed ρ should be approximately the 1 minus the ratio of the size of the largest cluster to N, the overall number of data points. For our synthetic data set seen in Figure 11, the largest cluster had 61 data points, and with $N = 219$, mean $\rho \approx 1 - 61/219 = .7215$, or 72.15%. Both $\rho = 70\%$ and $\rho = 75\%$ were tested as well as multiple other values to ensure our logic was sound, with $\rho = 75\%$ creating consistently connected results with the best verification statistics (to be explained in Section 8.2).

To validate our constraint works properly, we expect to see full connectivity of the cluster regions in our results. It is understood validation of the Cannot Link algorithm must be done outside of the GRENDEL algorithm. This will be done immediately,as we will take the true labeling for our $(Fe\text{-}Al\text{-}Li)O_x$ data as input and run our Cannot Link algorithm over multiple iterations where we randomly swap a certain amount of cluster labels each time. Validation will be achieved if the algorithm successfully keeps Cannot-Link pairs of data points outside of the same cluster assignment over all iterations.

4.1.3 Peak Shifting

The last physical constraint needed to be implemented regards the peak shifting of the input waveforms in X. It was described in Section 1 how the scattering angle of a given peak an x-ray diffraction pattern is dependent upon the lattice spacing of the material's crystalline structure. But due to the alloying process, the material may not have a perfect lattice structure once it has cooled back down, and thus the lattice spacing may be a little bit off or nonuniform at a given point in the material. This creates a shifting of the peaks in the waveform, and this generates error considering the exact location of the peaks of the waveform for data point i, X_i is needed to determine its exact basis phases.

An analogy to help understand this can be done with ice cubes. Say you have a perfect ice cube, completely uniform in atomic structure at every point. It is then melted into water, and this water is put into a ice cube tray to be put back in the freezer. Once the ice is solidified again though, the new ice cube may not be exactly the same as the old one - air bubbles may have been trapped in the water during the freezing process, or the tray may not have been completely level when put into the freezer. This would distort the ice cube's atomic structure slightly. A similar idea occurs when alloying the three ternary compounds together. Slight shifts in the atomic lattice structure, while unnoticeable to the naked eye, can be seen through the shifting of peaks in the waveform. This can cause GRENDEL to incorrectly say we have two separate basis phases present in the material, when in reality they are just shifted versions of the same one. This error must be accounted for in order to have accurate clustering diagrams as well as accurate guesses at the basis phase patterns.

The algorithm AgileFD described in Section 2 takes into account peak shifting by allowing for m additional shifted copies of a given basis phase. AgileFD also does not do clustering, so say we have B basis phases present in a material. Then AgileFD allows $B * m$ basis phases, as long as at most one of the shifted copies of a basis phase is present at the given data point i [5],[6]. Note that this is an expert constraint rather than a law of physics, and we will have to do much of the same.

Adding in constraints for peak shifting is still an ongoing project. An initial attempt at doing so utilized topological data analysis, particularly something called mode clustering [13]. Looking at a sample waveform for the i^{th} data point, X_i, the mode of each peak is the scattering angle location, and each peak has a *birth time*, the maximum intensity value of the peak, and a *death time*, the intensity value where the given peak overlaps with the next peak in the waveform. Typically the death time corresponds to a light intensity of 0. Using the mode, birth time, and death time of each peak, we can construct a confidence interval about the mode of each peak.

Thus, looking at the basis phases in each cluster, we created confidence intervals for each significant peak in the basis waveform. If we compared two waveforms and saw that each of the confidence intervals overlapped for these phases, then we labeled these as peak-shifted versions of the same basis phase pattern. Formal mathematical definitions mode clustering using density functions of image pixels [13], yet while attempted to adapt this to our waveform data and doing experiments with identifying cluster confidence intervals, we realized that validation as well as implementation was infeasible in the current construct of GRENDEL. As described in Section 3, we update our E and P matrices within each cluster. That is, we update E_i and P_i for all i clusters, $i \in [1, K]$.

Using the *findpeaks* function in MATLAB to find the peak locations, birth times, and death times, we could construct rudimentary confidence intervals of all peaks for every basis phase. Our method was able to successfully identify phase-shifted basis phases which had peaks of high intensity, yet failed with phases with smaller intensity peaks. Furthermore, there is no good way to correct for shifted peaks after we have identified them. Even allowing multiple copies of peak-shifted basis phases within each cluster similar to AgileFD's constraints would not work, as the main issue is that the initial spectral clustering and Graph Cut steps have already clustered points into regions where the only difference is the peak-shifted basis phases. A better explanation of this issue is found in Section 7 under Figure 10. Major changes to GRENDEL may have to be implemented in order to implement a peak shifting physical constraint, such as updating our E and P matrices for the entire material at once rather than solving the smaller subproblem of E_i and P_i for cluster i for all K clusters.

5 Implementation

The overall GRENDEL algorithm and all of the constraint programming is written in MATLAB R2015a. The Graph Cut portion is coded in C++, yet our goal is to not change this function as it has been optimized over years of research [9],[8],[10]. The code is run on a personal ASUS laptop with a 2.4 GHz Intel processor and 8 GB of RAM. If needed a high-performance computer may be used, although previous implementations have run well on the current computer specifications.

6 Datasets

The datasets to be used to test this algorithm fall into one of two categories. The first group of materials data will be taken from the Inorganic Crystal Structure Database, a large library of material data. The ICSD will give us structural diffraction spectral data and spatial composition data. These materials, however, have not been previously analyzed before and thus validation cannot be done on this dataset. An example of the Fe-Ga-Pd ternary system from the ICSD we wish to test on after validation of the entire constraint programming portion of the project is seen in Figure 6.

The second group of material data we use is a synthetic data set given to us by the creators of the GRENDEL algorithm [3]. This diffraction data has been generated for validation testing purposes, as we know the basis phase patterns in E, the proportion of basis phases P, and the cluster membership U for each given data point in the material. For validation of the connectivity constraint programming done so far, we use the $(Fe$-Al-$Li)O_x$ data set, which is known to have k=7 clusters and 6 basis phases. The true ternary clustering diagram for $(Fe$-Al-$Li)O_x$ can be seen in Section 7 in Figure 11.

7 Validation Methods

Running the original algorithm, prior to applying the connectivity constraint programming, the results of GRENDEL on the $(Fe$-Al-$Li)O_x$ ternary system is seen in Figures 7 and 8. Gibb's phase rule is seen when looking at the spectral plots for each of the k=7 clusters. For each cluster, at most three waveforms are seen (indicated by the orange, yellow, and blue waveforms in each of the plots in Figure 8). The plots are labeled accordingly by cluster corresponding to the colored regions seen in Figure 7.

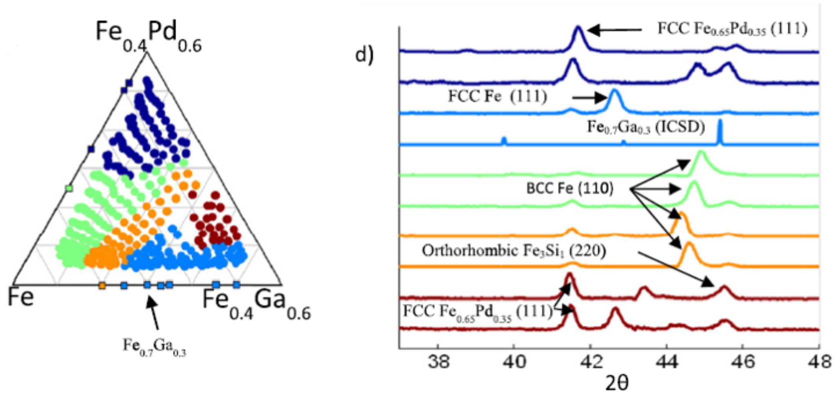

Figure 6: To the left, we have the *Fe-Ga-Pd* system ternary diagram, illustrating the clustering of a previous GRENDEL experiment. To the right is the constituent phase plot of the 10 basis phases seen in the material [3]. Since we do not know true values for this data set, this material will be analyzed after validation procedures are completed.

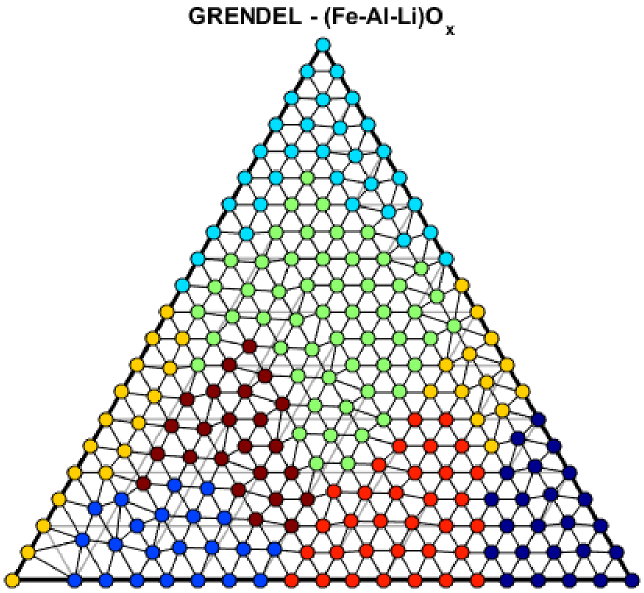

Figure 7: The ternary clustering diagram for the $(Fe\text{-}Al\text{-}Li)O_x$ system for the original GRENDEL algorithm.

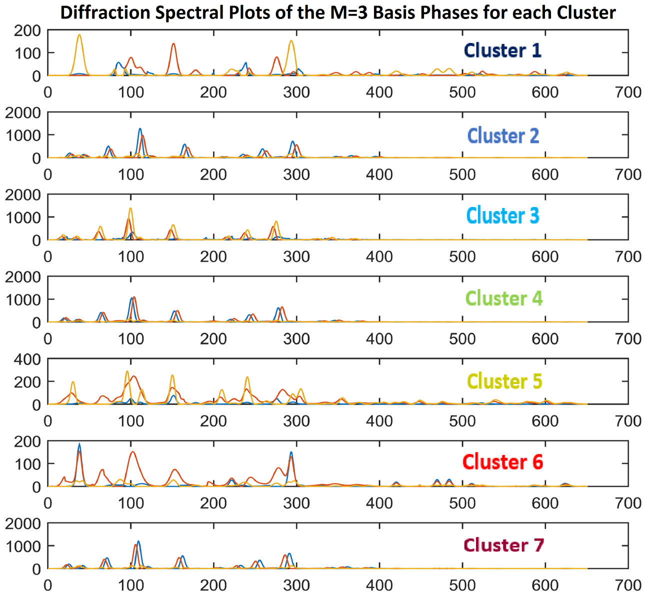

Figure 8: The corresponding basis phase waveforms for each of the k=7 clusters. The legend to the right is to associate the given spectral phase plots to their respective cluster color in Figure 7.

Regarding validation of the cannot-link connectivity constraints, see Figures 9 and 10. As stated earlier, validation of our connectivity constraints upholding the laws of physics can be seen by each of the 7 clusters in the new ternary diagram being fully connected. This is in comparison to the original GRENDEL's clustering, which we see has the disconnected yellow cluster in Figure 7. As stated in Section 4.1.2, verification outside of GRENDEL is to be done as well.

But comparing our results to the true labeling of the synthetic data, seen in Figure 11, we see that while are clusters are connected, they do not exactly match up with the true clustering. Note that the colors between our GRENDEL results and the true clustering do not match up. We can see the true maroon cluster is split up into our experimental red and dark blue clusters. Cluster splitting was very evident in this synthetic data set, which after analysis is a direct result of us not yet applying peak shifting constraints. Over half of the true basis phases are extremely similar to one another, and once this constraint is put in the colors regarding each of the clusters should match up. The key point for now, however, is that the main cluster boundaries are resolved with the connectivity constraints applied to GRENDEL.

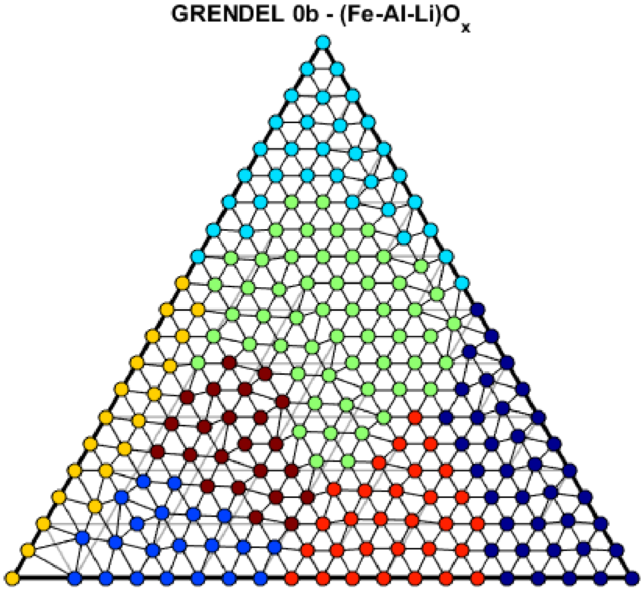

Figure 9: Ternary clustering diagram after adding in cannot link constraints. Note that all k=7 clusters are fully connected now.

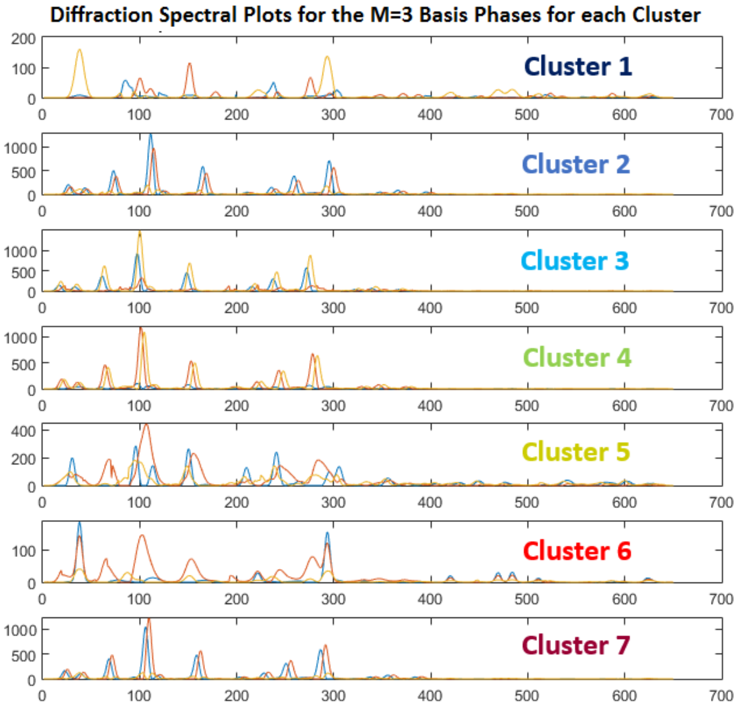

Figure 10: Basis phase waveforms for each of the 7 clusters, labeled accordingly to match up the colored regions in Figure 9. Again, we see Gibb's phase rule upheld.

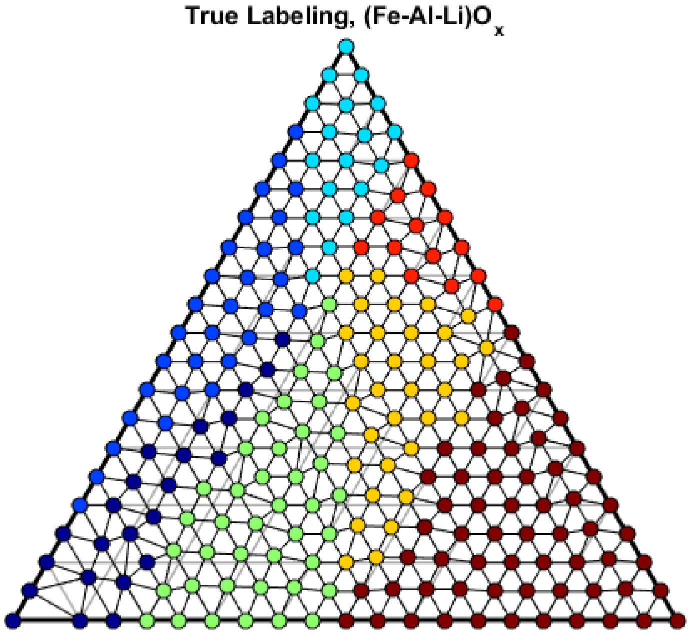

Figure 11: The true ternary clustering diagram for the $(Fe\text{-}Al\text{-}Li)O_x$ synthetic data set.

8 Concluding Remarks

8.1 Parameter Optimization

In order to increase the accuracy of our GRENDEL results, we also made sure our parameter values were optimized for the $(Fe\text{-}Al\text{-}Li)O_x$ data set. Through multiple runs of the GRENDEL algorithm, we were able to determine that the ρ parameter for the percent of Cannot Link pairs of our connectivity constraint should be set to 75. Furthermore, we verified that other parameters such as the spectral bandwidth parameter (σ in Equation (4)), data cost weight, and smoothness weight (λ_d and λ_s of Equation (14)) set in the original GRENDEL code were optimized as well. By optimized we mean to say the ternary clustering diagram upheld the desired physical constraints, and our confusion matrix metrics to be described immediately were at their best values.

8.2 Verification Statistics

Due to cluster splitting in our experimental GRENDEL results due to unresolved peak shifting, there is not a one-to-one correspondence between GRENDEL clusters and the true clustering. Thus, we utilize a confusion matrix through the *confusionmatStats* MATLAB package to take this into account. This program assigns each of our GRENDEL cluster to correspond to the most-likely true cluster. For example, the dark blue and red clusters in Figure 9 would both be paired with the maroon cluster in Figure 11. From this, we utilized the weighted accuracy metric, which gives the percent agreement of correct true positive and true negative cluster assignments. In other words, it calculates the percent of points in the correct label and the percent of points correctly not assigned to an incorrect label, and averages these values. We also used the F-score metric, which solely calculates the percent of points in the material that are in the correct label, without taking into account cluster-splitting.

For the original GRENDEL algorithm, we saw a weighted accuracy of 0.8519, while the addition of connectivity constraints increased this value to 0.8758. The connectivity constraint also increased the F-score from 0.3406 to 0.4562, showing that our cluster assignments did indeed get better even with peak shifting, the main issue, still to be resolved. While the original GRENDEL ran in 37.05 seconds on the ASUS laptop, connectivity constraints only increased the run time to 57.46 seconds, still below the desired threshold on one minute.

8.3 Other Unsuccessful Physical Constraint Algorithms

8.3.1 Connectivity Constraints using Data Costs

Another algorithm developed this semester attempting to satisfy the connectivity constraints utilized appending the cost function to be minimized in Graph Cut. This allowed us to apply this algorithm prior to Graph Cut every time, compared to the Cannot Link algorithm in Section 4.1.2 needing to be implemented after at least one iteration of Graph Cut. It uses the initial spectral clustering step's cluster membership matrix U as the true clustering, so we still require an initialization of clusters. When a Cannot Link pair of points was clustered together, it would look over all other potential cluster assignments of the two points ($2k-2$ clusters) and change the membership of the point with the minimum data cost value to move to another cluster assignment. Next, the data cost of the old cluster assignment was set to ∞ to ensure it would not be reassigned to it. The algorithm applied was as follows:

```
Compute $\delta_{\cos}(X_i, X_j) \quad \forall\ (i, j) \in [1, N]$;
Assign top $\rho$% of unique $(i, j)$ pairs into $CL$;
Prior to every Graph Cut:
    for $i = 1 : length(CL)$
        if $CL(i, 1)$ and $CL(i, 2)$ are in the same cluster
            Find cluster $c$ with minimum data cost to switch either $CL(i, 1)/CL(i, 2)$ to it;
            if DataCost$(CL(i, 1) \in c) <$ DataCost$(CL(i, 2) \in c)$
                $U(c, CL(i, 1)) = 1$
                $U(c_{\text{old}}, CL(i, 1)) = 0$
                DataCost$(CL(i, 1) \in c_{\text{old}}) = \infty$
            else
                $U(c, CL(i, 2)) = 1$
                $U(c_{\text{old}}, CL(i, 2)) = 0$
                DataCost$(CL(i, 2) \in c_{\text{old}}) = \infty$
            end
        end
    end
```

This created terrible results regardless of the choice of parameter ρ. Checking all values of ρ for the best test statistics, we were still only able to achieve an F-score of 0.2179 for $\rho = 30\%$, less than half of the F-score of the Cannot Link algorithm. Figure 12 shows the ternary diagram for if $\rho = 50$. Attempting this method of constraint programming, however, did show us that we needed to trust the initial Graph Cut cluster membership and that the PCOMMEND updates of E and P are what caused a lack connectivity.

8.4 Active Learning

The probing process employed to create the x-ray diffraction spectra of a given material takes half an hour per sample point. To allow for fast analysis of a large set of unknown materials, one of the major goals of this project, we need to minimize the number of data points necessary to create accurate analysis with our algorithm. Previously, GRENDEL used the spectra of the entire sample as an input, but for this approach we must change the input to iteratively read in single sample points of the material, so our value of N increases by one for each run of GRENDEL. This simulates the algorithm running in conjunction with the diffraction process. We start with a initial number of sample points of spectral data, and GRENDEL is run on this small subset of points within the material to initially estimate clusters and cluster basis phase composition as described in Section 3.

Then, these findings are utilized to predict the constituent basis phase composition over the entire material. To do so, an unknown point in the material is assumed to have an basis phase composition that is a Euclidean distance-weighted average of the phase/endmember

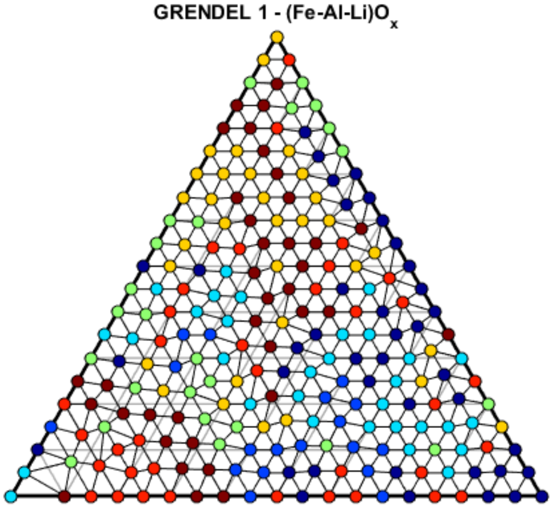

Figure 12: Ternary diagram for our Data Cost Cannot Link algorithm. Note terrible results.

compositions of the already-sampled data. We can then assign a dissimilarity metric to isolate a specific point within the material that is least similar to the mean phase compositions of the current clusters, and take that as the next point to sample [11].

We currently believe we will utilize cosine distance again between hypothetical phase composition and cluster basis phases as this metric. For example, if we define $X_{hyp,j}$ to be the estimate of the diffraction data at point j in the material through the distance-weighted averaging of the $\bar{X}_i$ cluster mean diffraction patterns for all i clusters, $i \in [1, K]$ (for a refresher on what this is, see Section 3.1 on spectral clustering), then we wish to sample point $j^* \in [1, N]$ which has the largest cosine distance value between it and its most-similar mean cluster composition (meaning it does not match with any cluster well):

$$X_{j^*} = \arg\max_{j \in [1,N]} \left(\arg\min_{i \in [1,K]} ||X_{hyp,j} - \bar{X}_i|| \right) \tag{18}$$

.

At this stage in the project a formal mathematical equation/algorithm to describe this process has yet to be formulated. While we understand we wish to use Euclidean distance as the similarity metric to create the continuous phases distribution/composition over the whole material, the exact methodology will depend on the peak shift constraint programming described in Section 4.1.3, as the structure of the GRENDEL algorithm may change. In addition, how we represent the basis phases and shifting in the matrix E may change the mathematical process we want to use. A validation method must also be determined.

8.4.1 Harder Connectivity Constraints

We also wanted to see if our Cannot Link algorithm in Section 4.1.2 was strict enough. In other words, we wanted to see if adding an additional constraint would help increase the accuracy of our clustering output. We wanted to see if the Cannot Link algorithm correctly made sure that all clusters were connected during every iteration of Graph Cut. After every iteration of graph cut, we checked the cluster membership of each data point to make sure at least one of its neighboring points in the simplex had the same cluster membership. If indeed there was a point j disconnected from its cluster, we then went through the same Data Cost procedure as described above to find the most-likely cluster membership of the clusters of j's neighbors, reassigned point j to that cluster, and set the Data Cost of the old cluster assignment to ∞.

Implementing this new constraint only showed though that our Cannot Link algorithm does a satisfactory job of keeping connectedness of clusters over all iterations of GRENDEL. While during the validation process we were able to isolate disconnected points in the sample material, when applying this constraint along with Cannot Link there was never any disconnected points that Cannot Link did not already address. Thus, the ternary diagram of this harder connectivity constraint is the same as that in Figure 9, and had the same F-score of that of the Cannot Link algorithm. Yet this did prove that we have adequately addressed the physical law of connectedness of clusters in order to focus on creating an algorithm to constrain peak shifting.

9 Timeline

The project timeline had to be appended by this mid-year support, as we recognize finishing the constraint programming portion is of the utmost importance. The project will still be divided into three separate stages, with the first stage complete:

1. Fully understand GRENDEL, replicate the previous results (*mid/late October*) **completed**
2. Constraint Programming
   (a) Add connectivity constraints and expert prior knowledge (*November*) **completed**
   (b) Add constraints for peak shifting (*January*)
   (c) Potential addition of other physical laws if necessary (*February*)
3. Active Learning
   (a) Have an algorithm to predict the next best point to sample (*March*)
   (b) Optimize the sampling algorithm for one material (*mid April*)
   (c) Optimize algorithm for all given material data (*late April*)

10 Deliverables

I will be delivering my final algorithm, which will compute the constituent phase composition and clustering. This code will also include the active learning component, so the number of sample points needed to generate the desired results will be outputted as well. When available, percent agreement between our output and previously-generated phase decomposition data will be available. Phase diagrams, spectral graphs, and numerical phase composition proportions will be generated for each of the given sample materials. In addition, an end of the year report and presentation will be given per requirements of the course.

References

[1] Lebras R., Damoulas T., Gregoire J.M., Sabharwal A., Gomes C.P., and van Dover R.B., *Constraint reasoning and kernel clustering for pattern decomposition with scaling*, AAAI **CP'11** (2011), 508–522.

[2] Takeuchi I., *Data Driven Approaches to Combinatorial Materials Science*, Materials Research Society Spring Meeting presentation, University of Maryland, College Park, 2016.

[3] Kusne A.G., Keller D., Anderson A., Zaban A., and Takeuchi I., *High-throughput determination of structural phase diagram and constituent phases using GRENDEL*, Nanotechnology **26** (2015), no. 44, 444002.

[4] Ermon S., LeBras R., Suram S., Gregoire J.M., Gomes C.P., Selman B., and van Dover R.B., *Pattern decomposition with complex combinatorial constraints: application to materials discovery*, AAAI Conference on Artifical Intelligence (1972), available at `http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/10020`.

[5] Ermon S., LeBras R., Suram S., Gregoire J.M., Gomes C.P., Selman B., and van Dover R.B., *Automated Phase Mapping with AgileFD and its Application to Light Absorber Discovery in the V-Mn-Nb Oxide System*, 2016.

[6] Xue Y., Bai J., LeBras R., Rappazzo B., Bjork J., Longpre L., Suram S., van Dover R.B., Gregoire J.M., and Gomes C.P., *Phase-Mapper: An AI Platform to Accelerate High Throughput Material Discovery*, CoRR **1610** (2016).

[7] Zare A., Gader P., Bchir O., and Frigui H., *Piecewise Convex Multiple-Model Endmember Detection and Spectral Unmixing*, IEEE Transactions on Geoscience and Remote Sensing **51** (2013), no. 5, 2853–2862.

[8] Boykov Y., Veksler O., and Zabih R., *Efficient Approximate Energy Minimization via Graph Cuts*, IEEE Transactions on PAMI **20** (2001), no. 12, 1222–1239.

[9] Boykov Y. and Kolmogorov V., *An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision*, IEEE Transactions on Geoscience and Remote Sensing **51** (2013), no. 5, 1124–1137.

[10] Kolmogorov V. and Zabih R., *What Energy Functions can be Minimized via Graph Cuts?*, IEEE Transactions on PAMI **26** (2004), no. 2, 147–159.

[11] Settles B., *Active Learning*, 18th ed., Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan and Claypool, 2012.

[12] Hastie T., Tibishirani R., and Friedman J., *The Elements of Statistical Learning - Data Mining, Interference, and Prediction*, 2nd ed., Springer, 2013.

[13] Wasserman L., *Topological Data Analysis*, 2016.