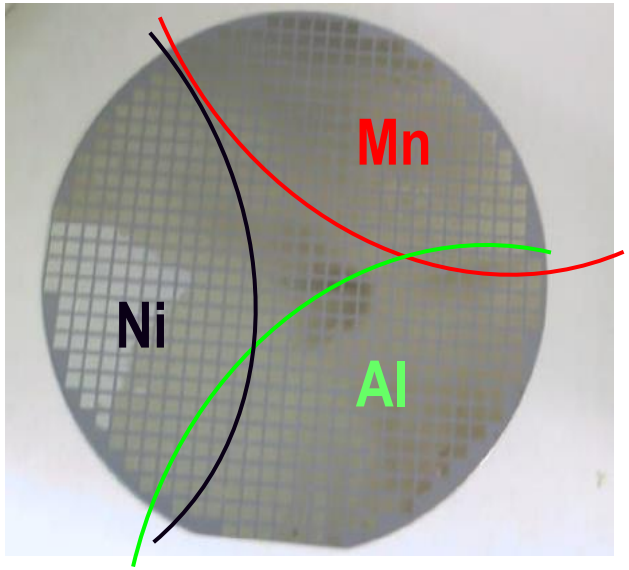
Pattern Decomposition of Inorganic Materials: Optimizing Computational Algorithm

Graham Antoszewski
ganto@math.umd.edu

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

March 9, 2016

Background Information – Materials Sciences

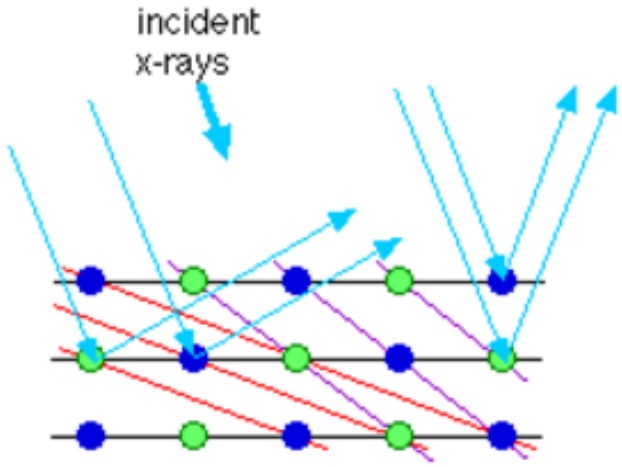


- Mixtures of metal alloys – ternary systems
  - Composition varies through material
  - Different composition = unique crystalline structure
  - Different chemical properties

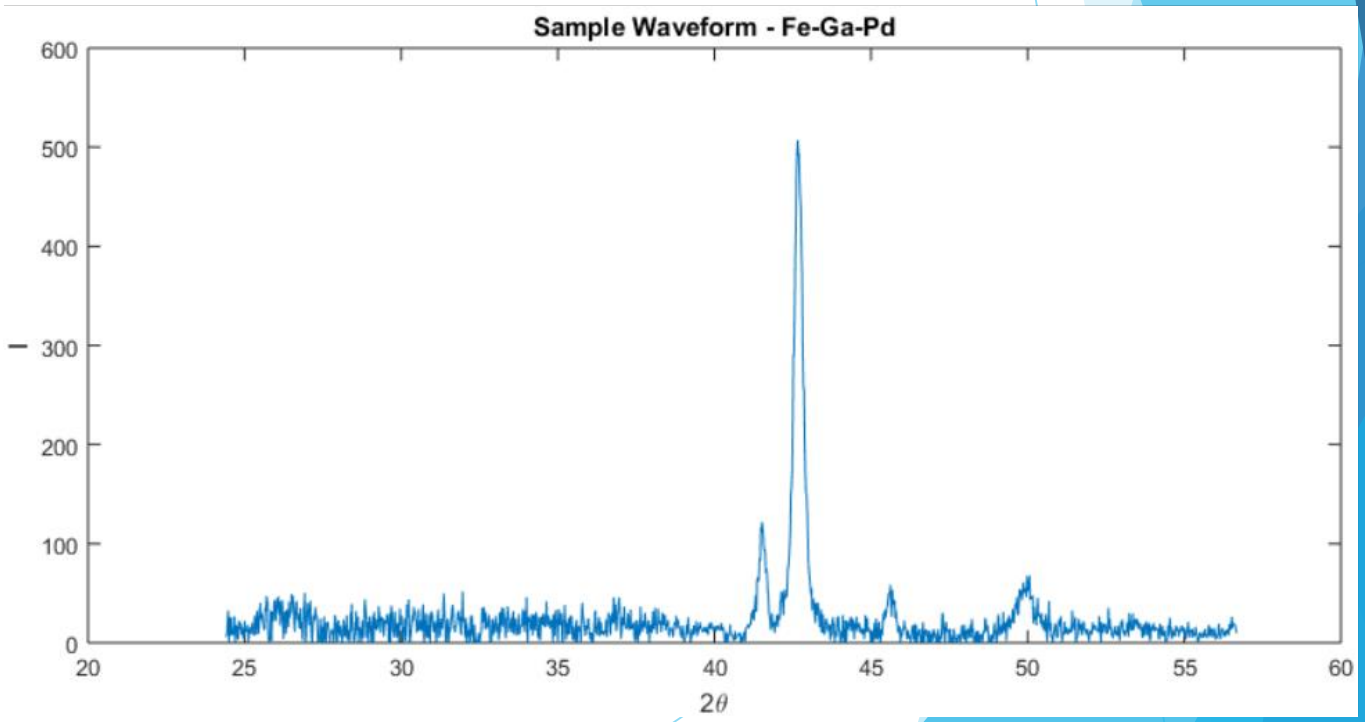
- Pattern Decomposition
  - Given a system of N sample points of numeric data (Ex: light intensity)
  - Want to find K basis "phase patterns" that describe data at all points
  - Like finding basis of a vector space
  - Phases tell us about the chemical properties of the material

Takeuchi I. (2016) MRS Meeting

Background Information – Pattern Decomposition



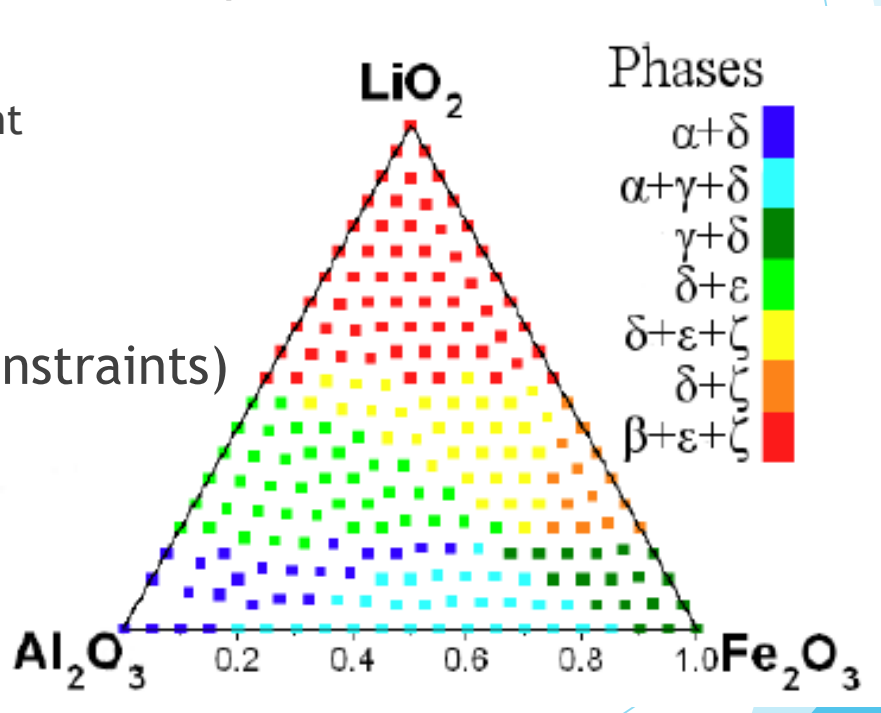
- Given material is sampled using electron probe
  - X-ray light is diffracted back at a certain angle
  - Based on lattice spacing
- Output is a continuous waveform
  - X axis - Scattering angle
  - Y axis - Intensity of diffracted light
- Determine composition via waveform
  - Like human fingerprint
  - Combination of basis waveforms



Top figure: http://physics.bu.edu/py106/notes/Resolution.html

Background Information – Phase diagrams

- After probing all sample points of a material, a simplex can be created
  - Illustration of phase composition at a given point
  - Colors = clusters (similar phase structure)
- Results must uphold to laws of physics (constraints)
  - Gibbs phase rule
  - Connectivity (continuity of phases in space)
  - Peak Shifting (effect of alloying process)



LeBras et al (2011) AAAI CP'11 508-522

What is the Computational Problem?

- White House Materials Genome Initiative
  - Develop algorithm to take in diffraction/composition data, output phase structure of materials

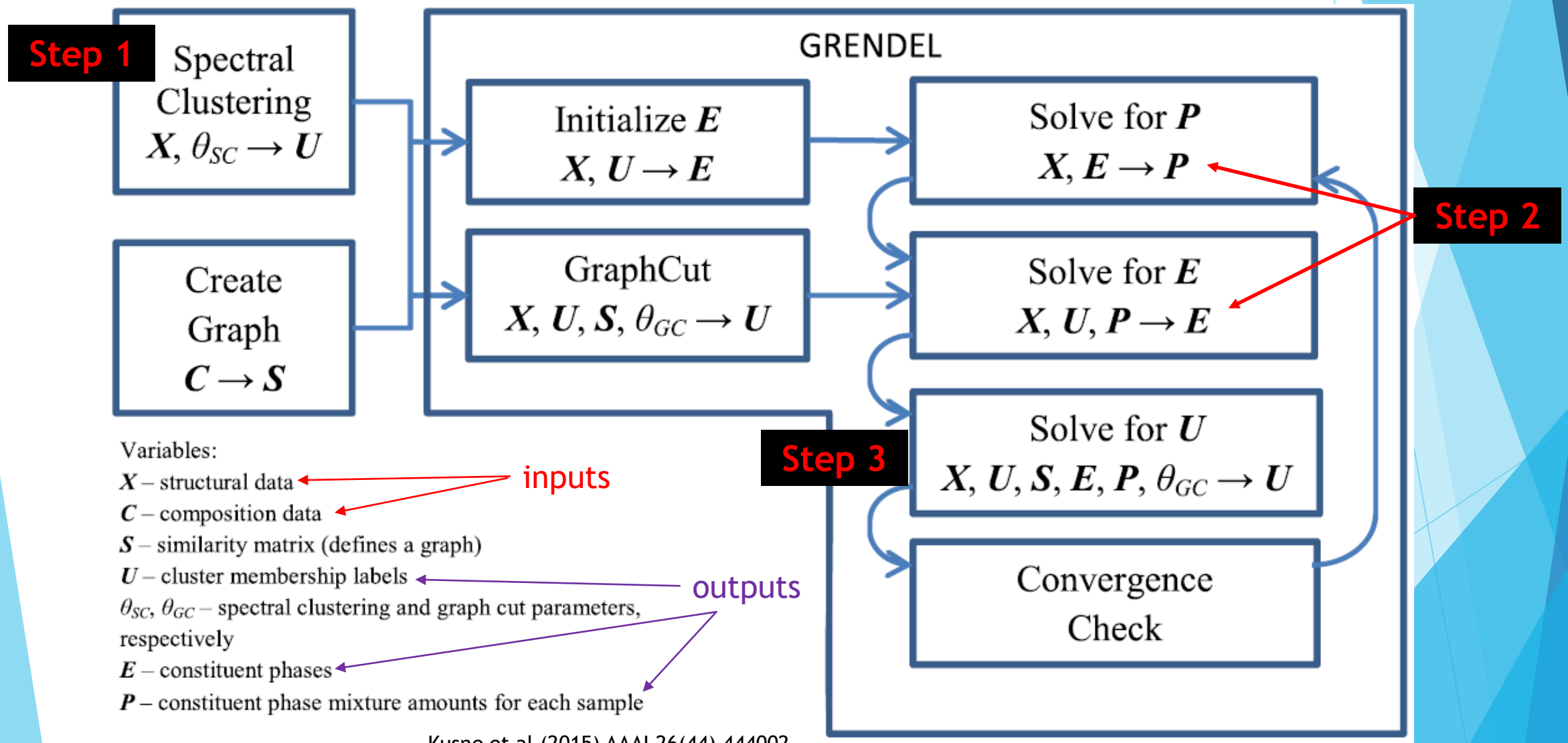
- Algorithm must:
  - Obey physical constraints (laws of physics)
  - Identify regions/clusters of similar phase composition within material
  - Identify basis phases accurately (≤3 per cluster)
  - Be efficient – short run times so more materials can be analyzed

Project Goal– Extending GRENDEL

- Take existing GRENDEL (Graph-based **End**member **E**xtraction and **L**abeling) code, apply strategies to make the algorithm more accurate and precise
  - GRENDEL does not adhere to physical laws and phenomenon, yielding inaccurate results

- Increase accuracy of clustering and basis phase detection results by incorporating constraints
  - Laws of physics
  - “Expert” prior knowledge of material
  - Affects cluster analysis and overall phase composition

GRENDEL Algorithm

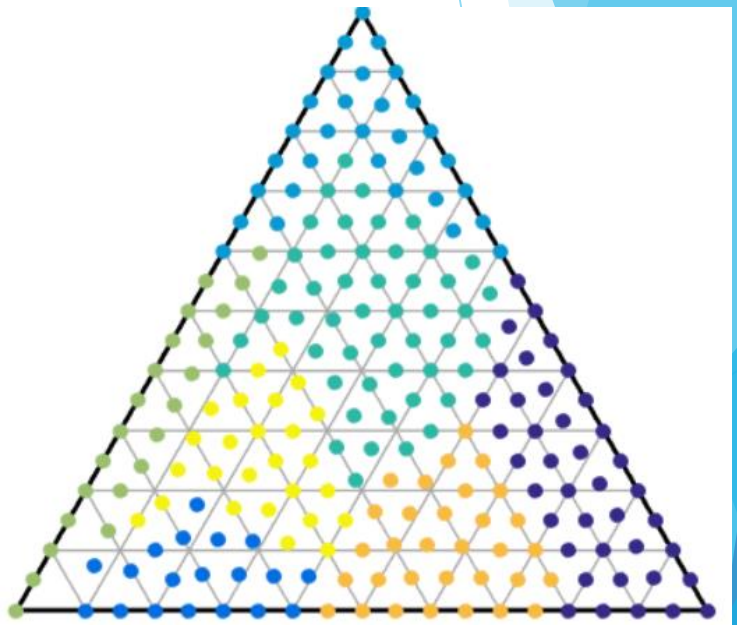


Kusne et al (2015) AAAI 26(44) 444002

Algorithm – GRENDEL
Step 1 – Spectral Clustering

- Input diffraction data – **X**, NxM matrix
  - N = # of data points
  - M = # of scattering angles sampled (length of waveform)
- Takes in diffraction data, creates a similarity matrix **W**
  - i,j – sample points
  - $\delta_{cos}(X_i, X_j)$ – cosine distance (1 – cosine of waveform vectors)
  - σ – spectral clustering bandwidth parameter (θ_{sc})
- Spectral Clustering Algorithm:
  - **G** = diagonal matrix summing rows of **W**
  - Find *k* smallest nontrivial eigenvectors of Graph Laplacian, **L** = $\mathbf{G^{-1}W}$
  - use MATLAB k-means on **X** to group points into k clusters corresponding to eigenvectors
  - **U** (kxN) – cluster membership matrix, **U**(c,i) = 1 if point i is in cluster c

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



Algorithm – GRENDEL
Step 2 – Nonnegative Matrix Factorization

- The goal of GRENDEL is to minimize an objective function:

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

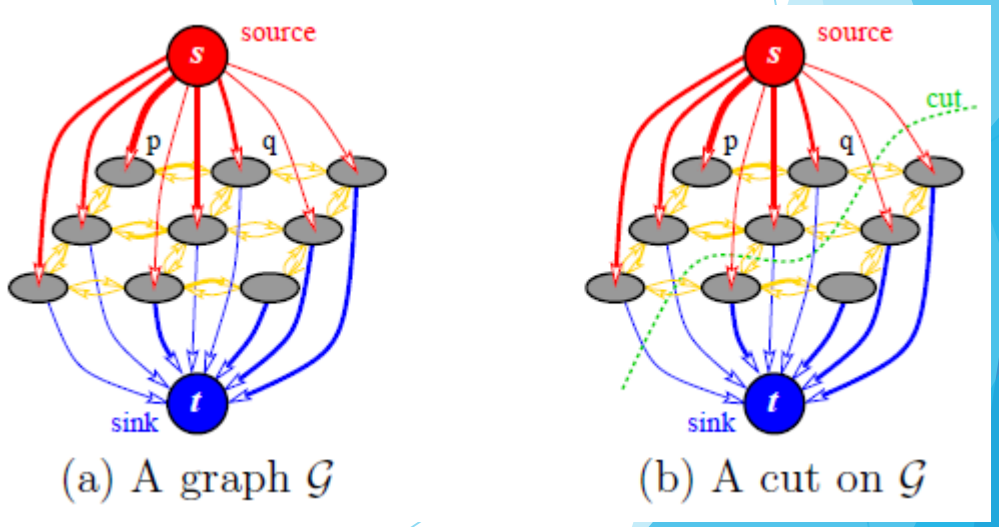
- **E** (DxM) – basis phases of ith cluster (unknown), e_{ij} is jth row of $\mathbf{E_i}$
- **P** (NxD) – phase proportions of ith cluster for jth sample point (unknown)
- **U** (KxN) – cluster membership
- Assume **X** can be approximated/reconstructed by **P*E**
- Set derivative of J with respect to **E,P** to update/output these matrices
- CURRENT WORK – REPLACING THE OBJECTIVE FUNCTION ABOVE FOR PEAKSHIFTING

Algorithm – GRENDEL
Step 3 – Graph Cut

- General "cost" equation to minimize: $V = \lambda_d \sum_i V^i(L_i) + \lambda_s \sum_{i,j \in N} V^{i,j}(L_i, L_j)$ (1) (2)
  - Require **X**, **P**, **E**, and **U** as inputs
- Smoothness cost (2) is 0 if cluster labels match, 1 otherwise, Data cost matrix (1):

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

- Minimize **V** through Max Flow Algorithm
  - Minimizes the entirety of **V**, not for each data point
  - Figure: Thickness of arrows = less cost to be in that colored cluster ('source' and 'sink')
  - Finds 'border' between clusters where cost to be in either adjacent cluster is most similar



(a) A graph $\mathcal{G}$ (b) A cut on $\mathcal{G}$

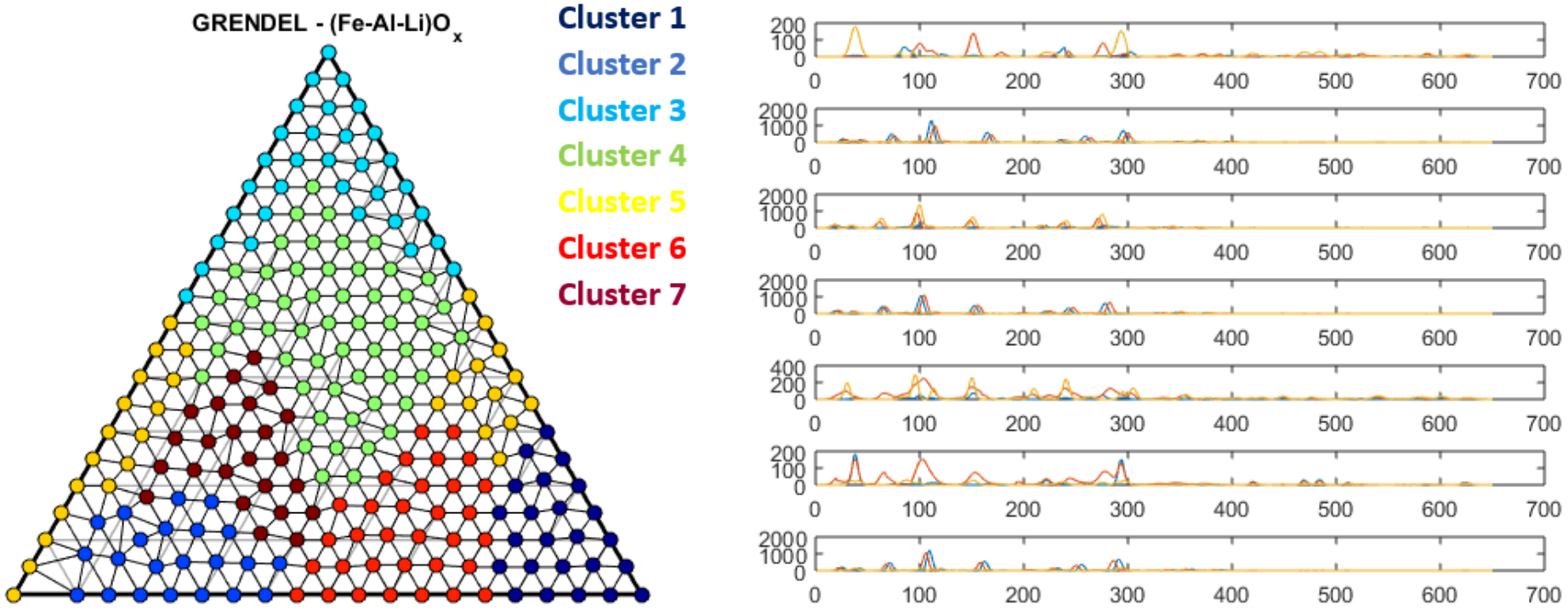
Boykov et al (2004) PAMI 26(9) 1124-1137

Implementation

- Language - MATLAB R2015a
- Hardware - personal computer
  - ASUS, 8 GB RAM
- Data sets:
  - Inorganic Crystal Structure Database (Fe-Ga-Pd, from Kusne et al.)
    - Synthetic diffraction, structural data from previous research efforts ((Fe-Al-Li)O_x from Gregoire et al.)
    - X – input spectral waveform data (diffraction patterns)
    - C – input composition data (spatial coordinates)
  - NEW - Synthetic Spectral Data from ShiftNMF (Morup M. and Madsen K. H.)

Results – Original GRENDEL

- Plot to the left is ternary diagram (showing the 7 different clusters/colors)
- Plot to the right are the spectral (waveform) plots of the constituent phases for each cluster



Recap of Last Semester – Cannot Link Constraint Algorithm

- Analysis of algorithm – NMF updates of **E** and **P** are what violate cluster connectivity requirement
- Algorithm:

  Compute cosine distance between all pairs

  Assign top p% dissimilar pairs to 'Cannot Link' array

  After initial Graph Cut:

    Remove pairs in CL which are initially clustered together

  After all subsequent Graph Cut iterations:

    Loop through all CL pairs:

    If pair in same cluster

      If 1st point changed cluster

        Revert cluster assignment of 1st point to old cluster

      Else

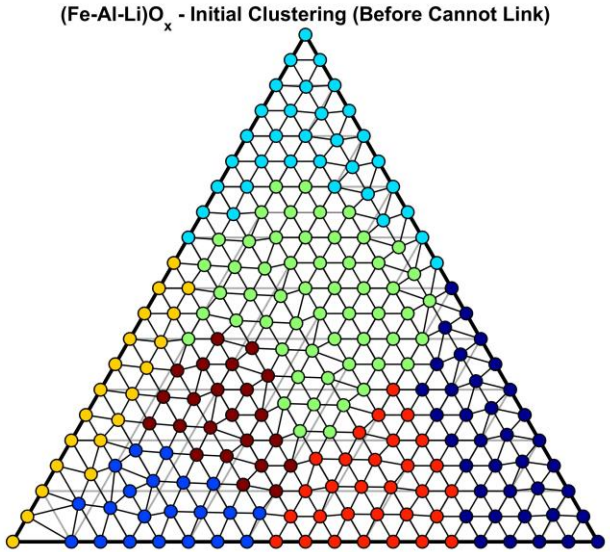
        Revert 2nd point's cluster assignment

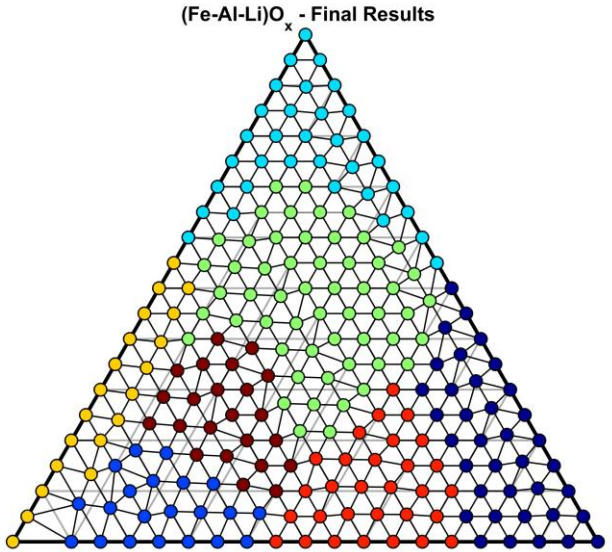
      end

    end

Validation of Cannot Link

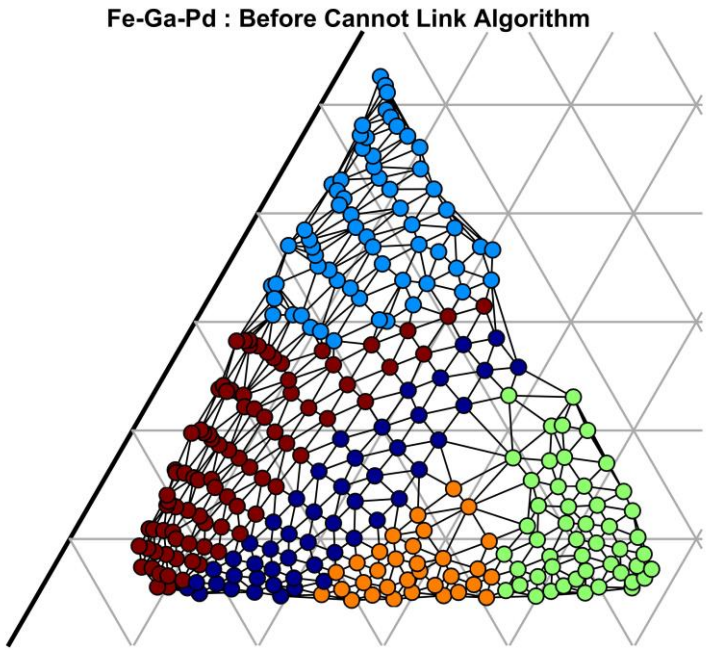
- Two different local minimums at GRENDEL converged to
- Had 3.92% and 4.01% of CL pairs deleted after initial Graph Cut
- After every iteration, 0% of remaining CL pairs were in the same cluster
- Data replicated for 50 trials

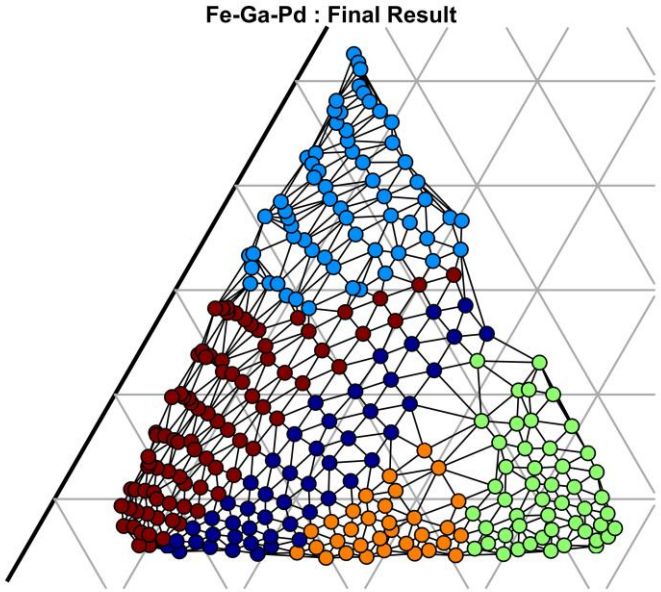




Validation of Cannot Link

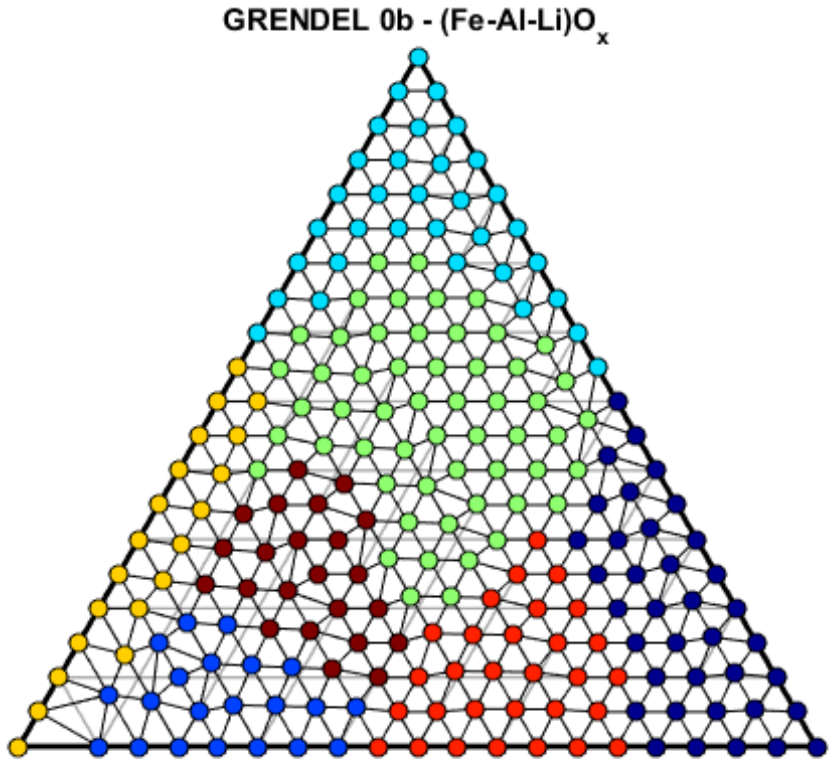
- One local minimum, 3.30% of CL pairs removed after initial Graph Cut
- Over 50 trials, after every iteration, 0% of CL pairs were in the same cluster
- Cluster connectivity constraint adhered to again

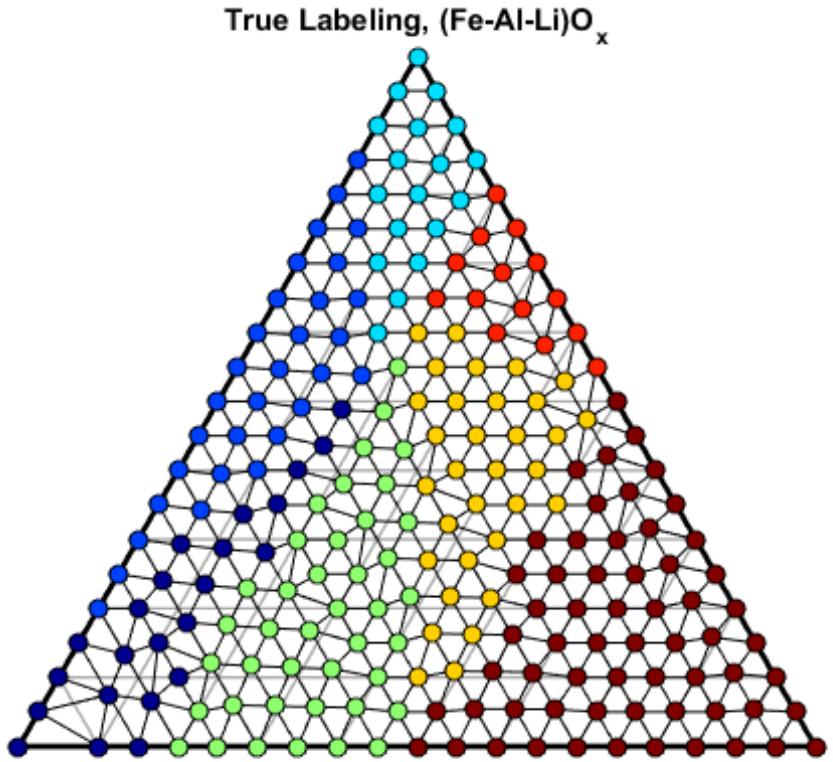




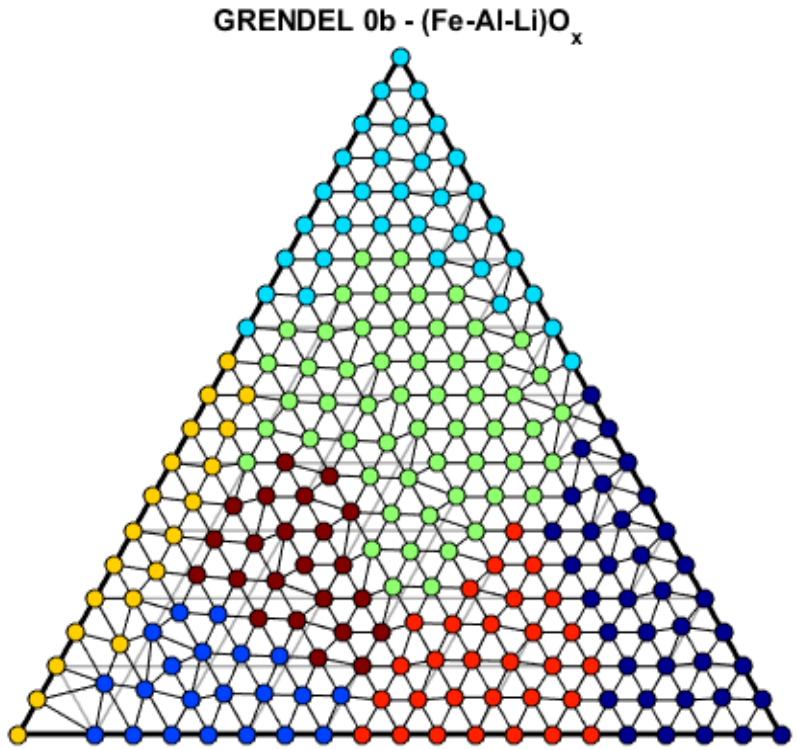
Comparison to True Values

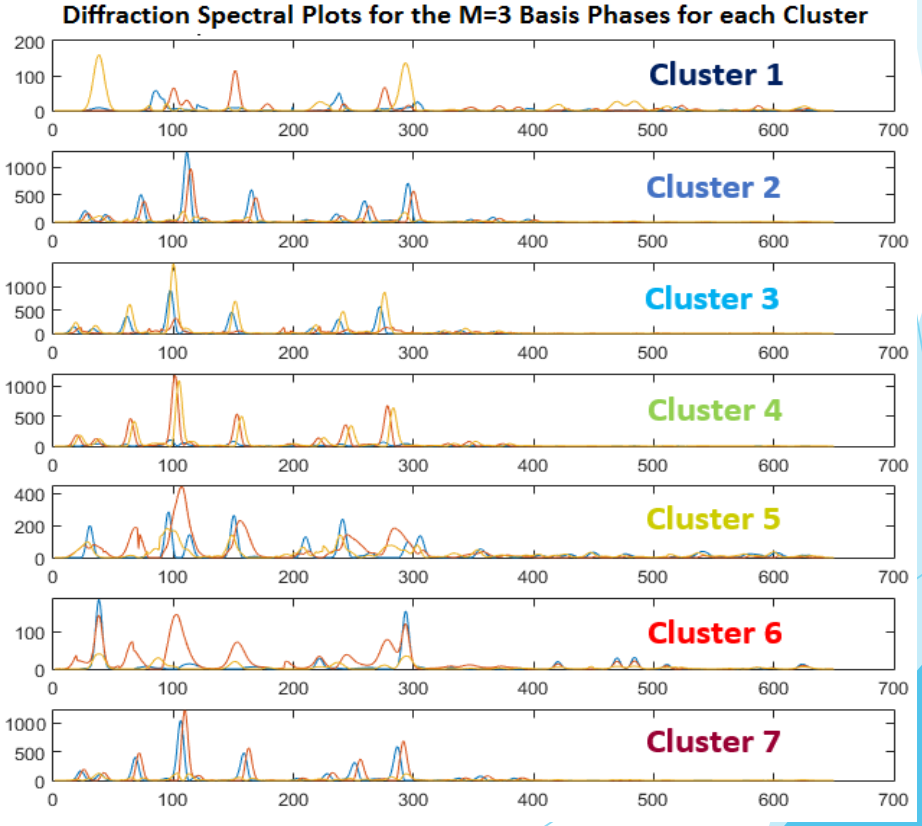
- Basis phases (E), Proportions (P), and Clustering (U) are previously known with synthetic data set
- GRENDEL – poor agreement with true clustering





Current Work – ‘Peakshifting’ Expert Constraint





Peakshifting – ShiftNMF Algorithm

- Novel idea – **T** (NxD) is matrix of 'shifting delays' at each data point for each basis phase
  - Apply delays in Fourier space ($\boldsymbol{X_f}$, $\boldsymbol{P_f}$, $\boldsymbol{E_f}$ = Fourier transform of **X**, **P**, and **E**)
- New Least Squares objective function (ω is frequency vector in Fourier space):

$$J_{LS}(P, E, T) = \frac{1}{2}||X - PE||_F^2 = \frac{1}{2M}||X_f - P_f E_f \circ \exp(i\omega T)||_F^2$$

  - Parceval's identity – allows us minimize error in both spaces
- Run iteratively until convergence is reached
- General Method to find update **P**, **E**, and **T**:
  - Apply Fast Fourier Transform (fft), add time delays **T** to either **P** or **E**
  - Find the derivative(s) of **J** with respect to the matrix we wish to update
  - Use gradients to create multiplicative update rules

ShiftNMF Algorithm – E and P update

$$E_{f,T} = E_f \circ \exp(i\omega T)$$

$$\text{grad}_P = \frac{-1}{M}(X_f - P_{f,T}E_f)E_{f,T}^H$$

$$\text{grad}_P^- = \frac{1}{M}X_f E_{f,T}^H$$

$$\text{grad}_P^+ = \frac{1}{M}P_f E_{f,T}E_{f,T}^H$$

$$G^+ = \text{ifft}(\text{grad}_P^+), \quad G^- = \text{ifft}(\text{grad}_P^-)$$

$$P = P \circ \left(\frac{G^-}{G^+}\right)^\alpha$$

Guaranteed convergence for $\alpha = 1$

$$P_{f,T} = P_f \circ \exp(i\omega T)$$

$$\text{grad}_E = \frac{-1}{M}P_{f,T}^H(E_f - P_{f,T}E_f)$$

$$\text{grad}_E^- = \frac{1}{M}P_{f,T}^H P_{f,T}E_f$$

$$\text{grad}_E^+ = \frac{1}{M}P_{f,T}X_f$$

$$G^- = \text{ifft}(\text{grad}_E^-), \quad G^+ = \text{ifft}(\text{grad}_E^+)$$

$$E = E \circ \left(\frac{G^-}{G^+}\right)^\alpha$$

If $J_{new} \geq J_{old}$, then reduce α until $J_{new} < J_{old}$

ShiftNMF Algorithm – T update

- Utilizes Newton-Raphson method:
  - $\mathbf{T} = \mathbf{T} - \eta \mathbf{B}^{-1}\mathbf{g}$
  - η – step size parameter
  - **B** – Hessian
  - **g** - gradient

$$P_{f,T} = P_f \circ \exp(i\omega T)$$

$$Q_f = P_{f,T} E_f$$

$$Y_f = X_f - Q_f$$

$$g = \frac{-1}{M} \sum_{\omega} 2\omega \Im[Q_f Y_f^*]$$

$$B = \left\{ \begin{array}{ll} \frac{-2}{M} \sum_{\omega} \omega^2 \Re[Q_f Q_f *], & \text{for diagonal entries} \\ \frac{-2}{M} \sum_{\omega} \omega^2 \Re[Q_f (Q_f * + Y_f *)], & \text{else} \end{array} \right\}$$

$$T = T - \eta B^{-1} g$$

$$\text{If } J_{new} \geq J_{old}, \quad \text{then reduce } \eta \text{ until } J_{new} < J_{old}$$

ShiftNMF Algorithm – Cross-Correlation Step

- Due to complexity of the objective function, local minima are abundant
- To avoid these, every 20 iterations we run a ‘cross-correlation step’
- Done for each element in **T** in random permutation to shake up our **T** matrix

Randomly select d’ phase, n’ data point

Let $X_{n',f} = \text{fft(X) at n'}, \quad P_{n',f} = \text{fft(P) at n'}$

Let $E_{d',f} = \text{fft(E) at d'}$

$$R_{n',f} = X_{n',f} - \sum_{d \neq d'} P_{n',f} E_{d,f}$$

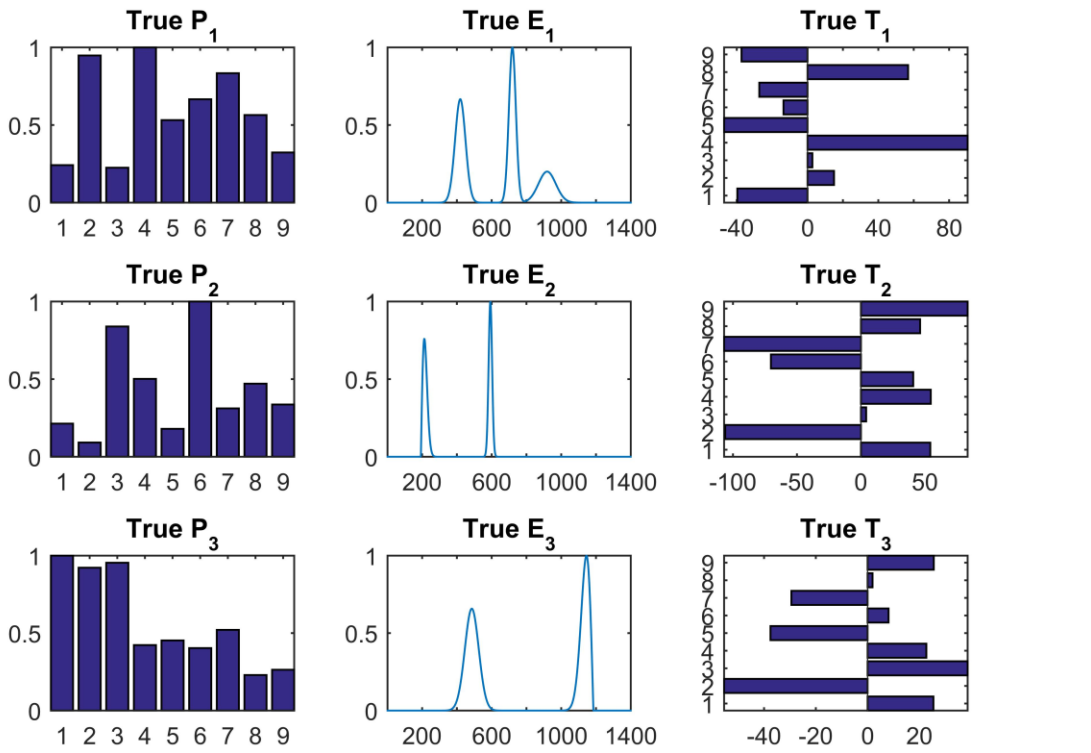
$$C_{n',f} = R^*_{n',f} E_{d',f}$$

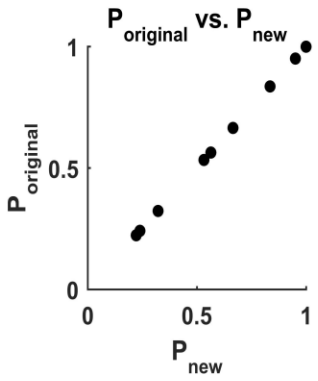
$$t = \arg\max C_{n',f}$$

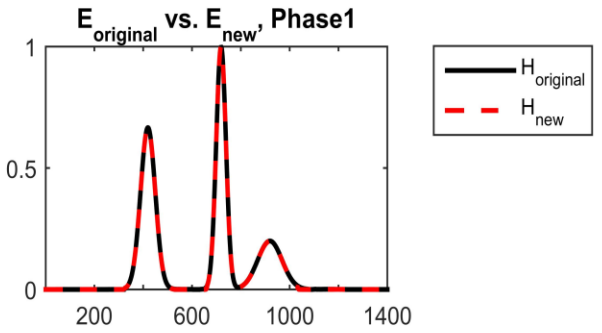
$$T_{n',d'} = t - (M+1)$$

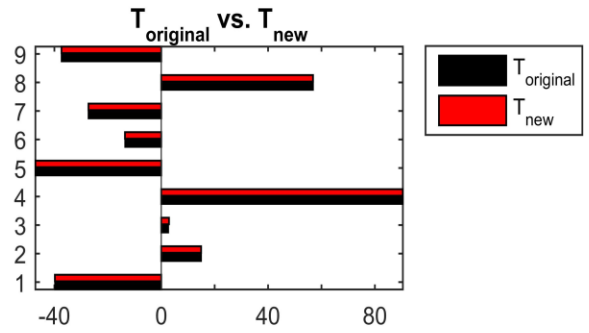
Validation of ShiftNMF – Input # 1

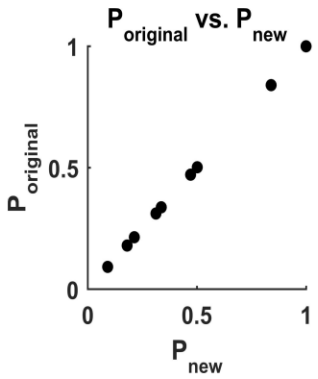
- Utilizing input data from previous authors of ShiftNMF
- We wish to compare original ShiftNMF to my version
  - Seek to achieve comparable convergence statistics and plots
  - Wish to test robustness of the two algorithms
    - Does not always converge to global minimum
  - ShiftNMF does not always 100% reconstruct correct values of **P**, **E**, and **T** for complex diffraction patterns
- N = 12, D = 3, M = 1400 (Best-case scenario of input data)

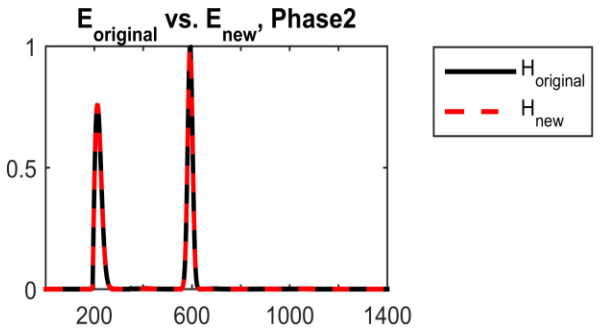


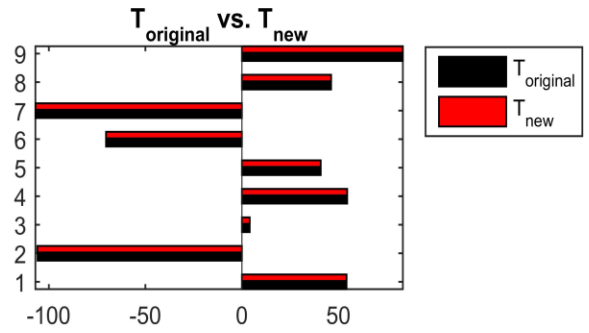

P original vs. P new
P original
P new
0
0.5
1

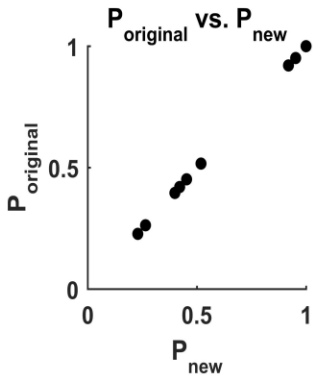

E original vs. E new, Phase1
H original
H new
0
0.5
1
200
600
1000
1400

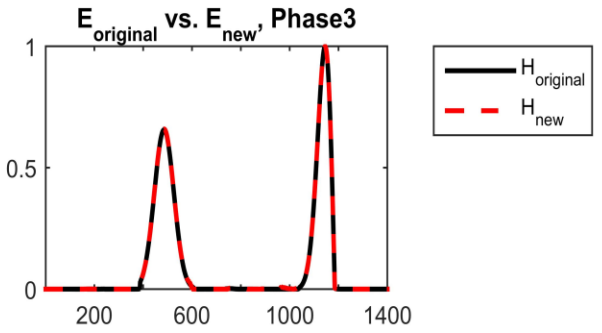

T original vs. T new
T original
T new
1
2
3
4
5
6
7
8
9
-40
0
40
80

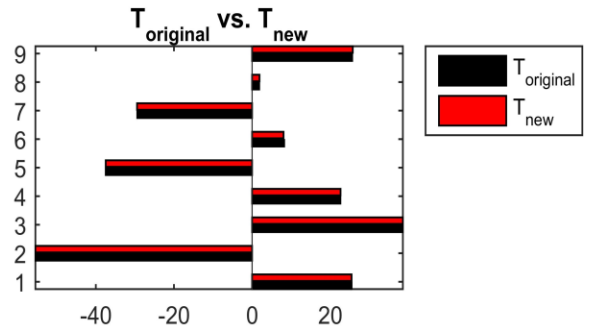

P original vs. P new
P original
P new
0
0.5
1


E original vs. E new, Phase2
H original
H new
0
0.5
1
200
600
1000
1400


T original vs. T new
T original
T new
1
2
3
4
5
6
7
8
9
-100
-50
0
50

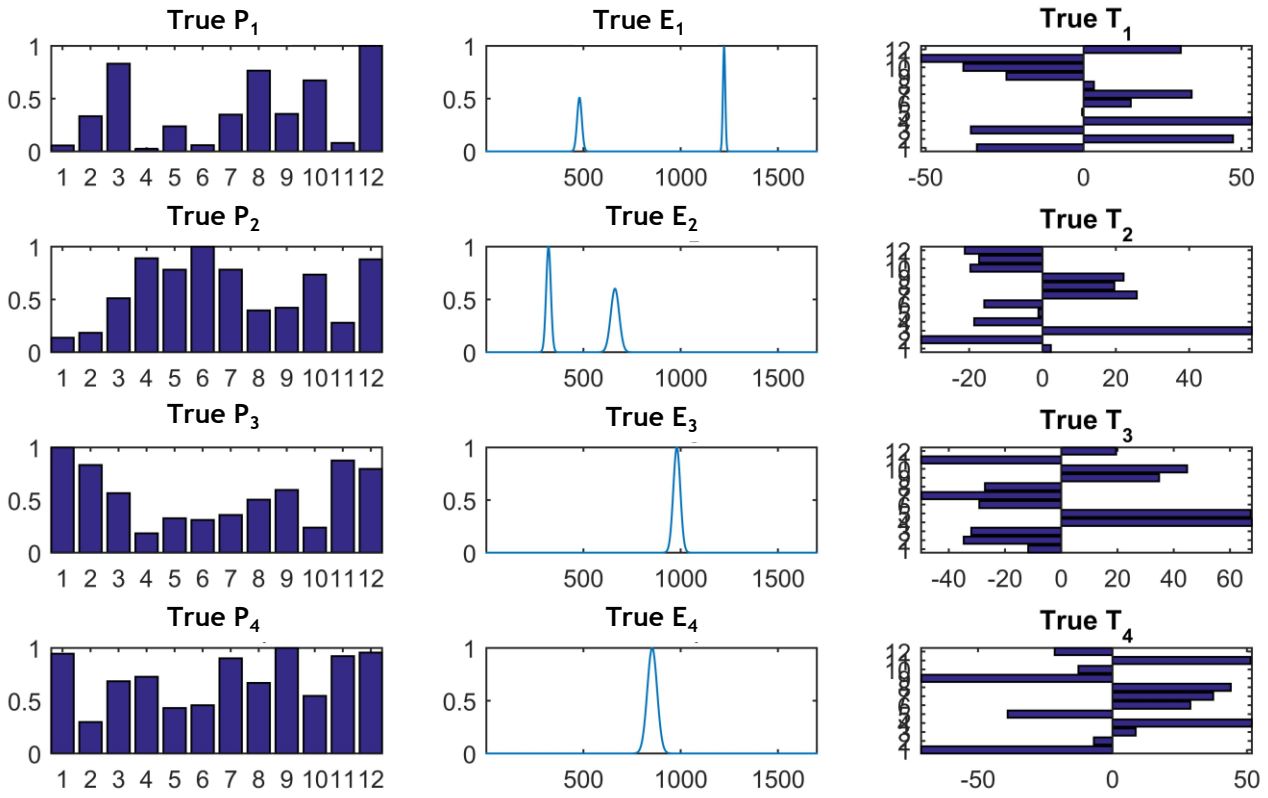

P original vs. P new
P original
P new
0
0.5
1

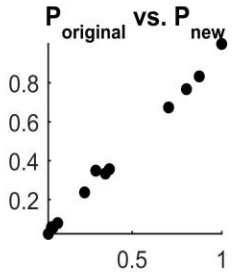

E original vs. E new, Phase3
H original
H new
0
0.5
1
200
600
1000
1400

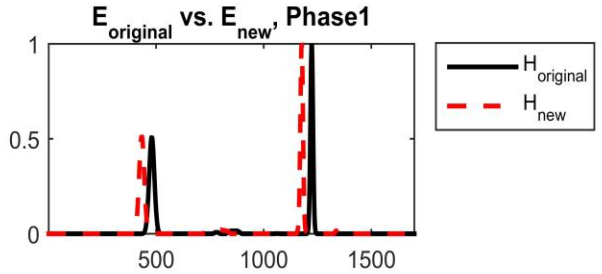

T original vs. T new
T original
T new
1
2
3
4
5
6
7
8
9
-40
-20
0
20

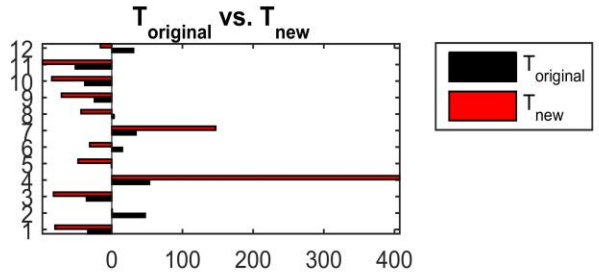
Validation of ShiftNMF – Input # 2

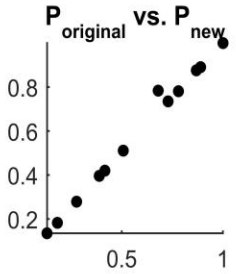
- N = 12, D = 4, M = 1700 (Good scenario of input data)

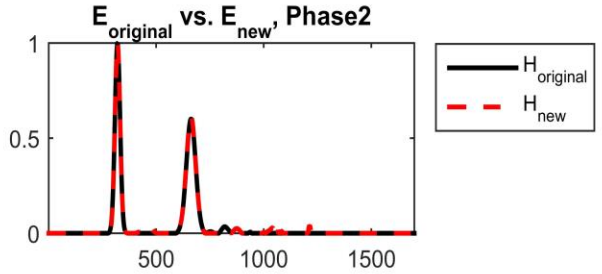


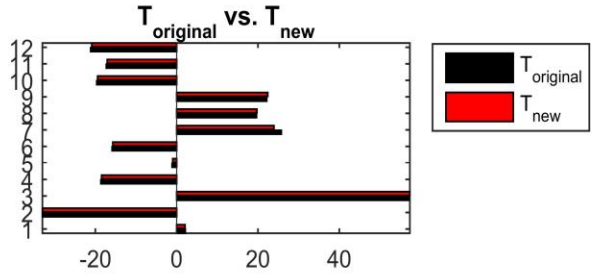

P original vs. P new
0.8
0.6
0.4
0.2
0.5
1

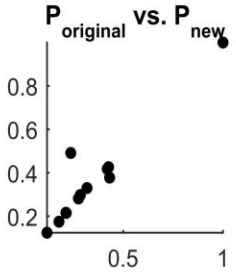

E original vs. E new, Phase1
1
0.5
0
500
1000
1500
H original
H new

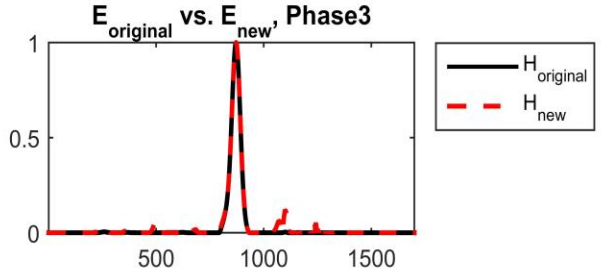

T original vs. T new
12
11
10
9
8
7
6
5
4
3
2
1
0
100
200
300
400
T original
T new

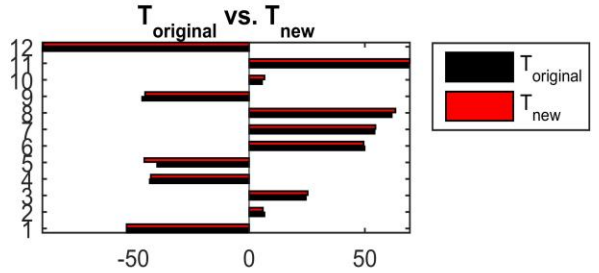

P original vs. P new
0.8
0.6
0.4
0.2
0.5
1

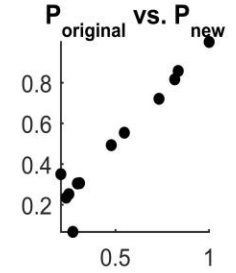

E original vs. E new, Phase2
1
0.5
0
500
1000
1500
H original
H new

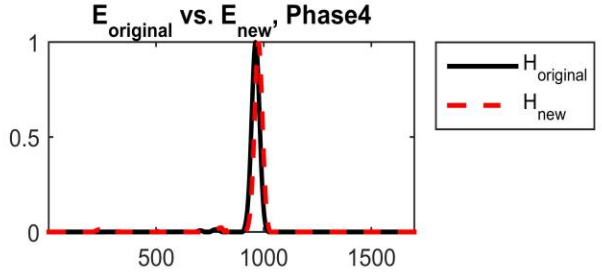

T original vs. T new
12
11
10
9
8
7
6
5
4
3
2
1
-20
0
20
40
T original
T new

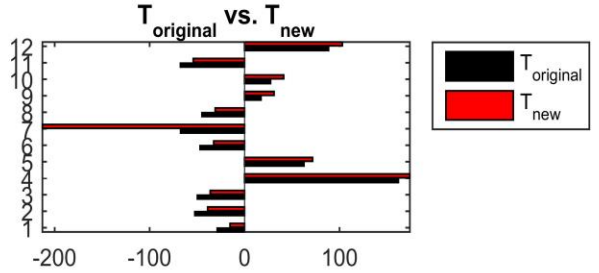

P original vs. P new
0.8
0.6
0.4
0.2
0.5
1


E original vs. E new, Phase3
1
0.5
0
500
1000
1500
H original
H new


T original vs. T new
12
11
10
9
8
7
6
5
4
3
2
1
-50
0
50
T original
T new

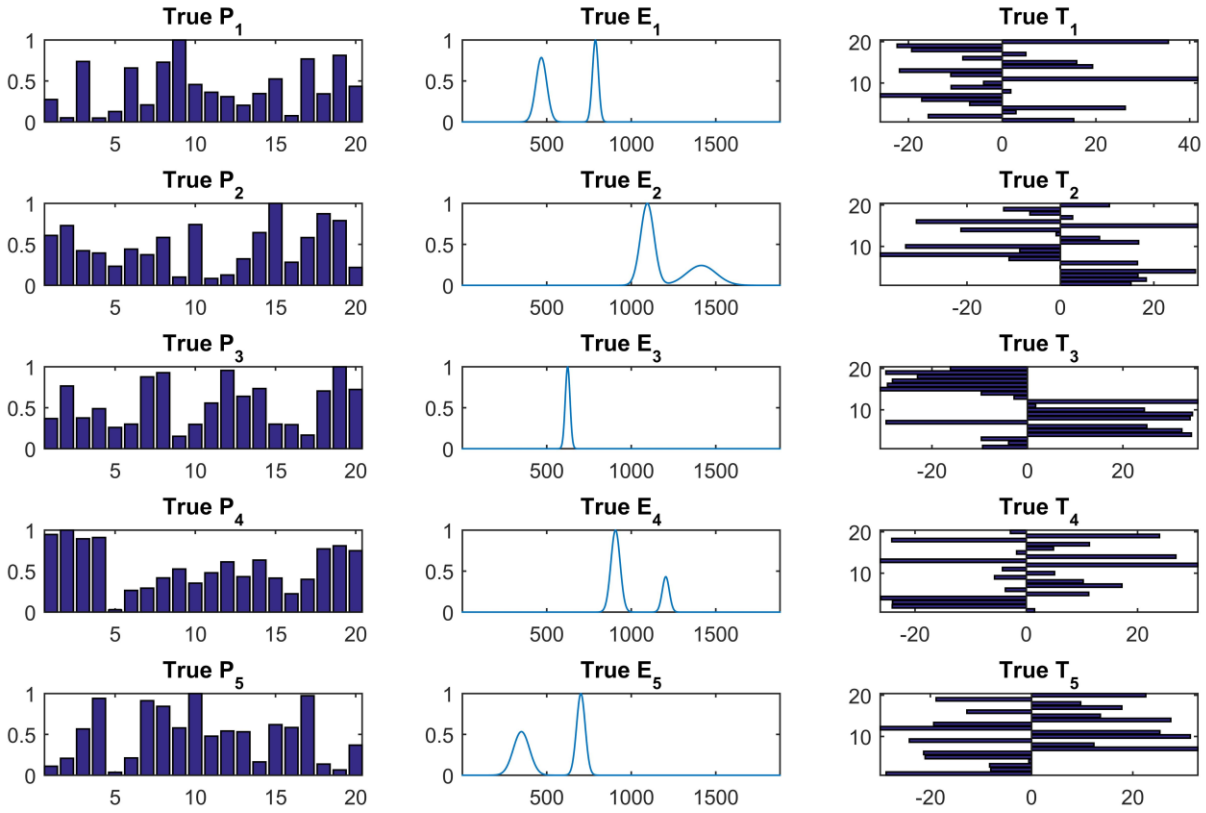

P original vs. P new
0.8
0.6
0.4
0.2
0.5
1

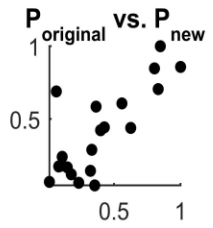

E original vs. E new, Phase4
1
0.5
0
500
1000
1500
H original
H new

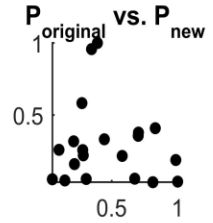

T original vs. T new
12
11
10
9
8
7
6
5
4
3
2
1
-200
-100
0
100
T original
T new

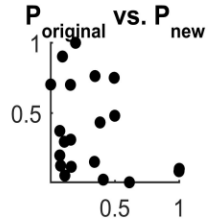
Validation of ShiftNMF – Input # 3

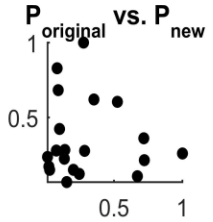
- N = 20, D = 5, M = 1880 (Poor/messy scenario of input data)

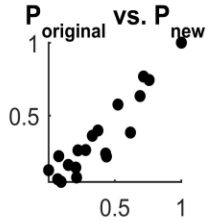


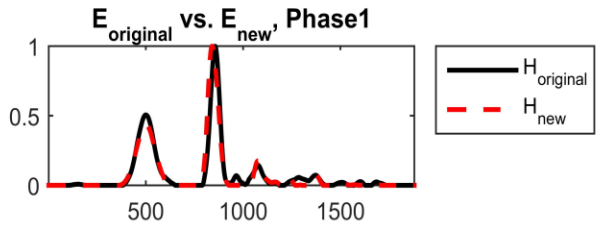

P original vs. P new
1
0.5
0.5
1

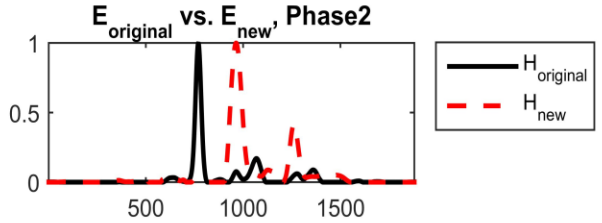

P original vs. P new
1
0.5
0.5
1

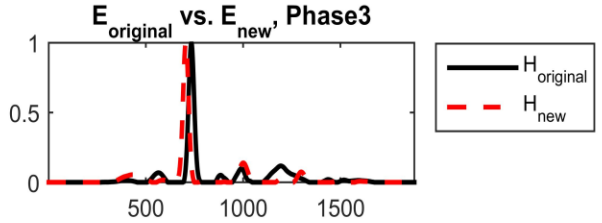

P original vs. P new
1
0.5
0.5
1

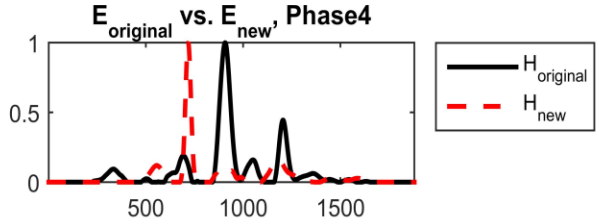

P original vs. P new
1
0.5
0.5
1

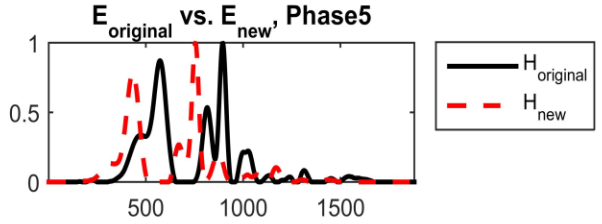

P original vs. P new
1
0.5
0.5
1

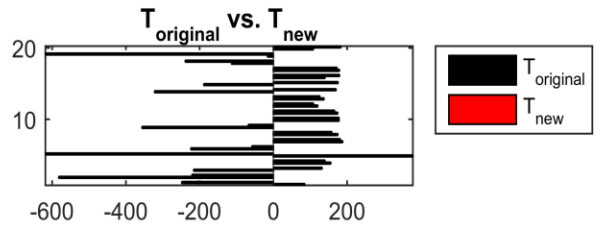

E original vs. E new, Phase1
1
0.5
0
500
1000
1500
H original
H new

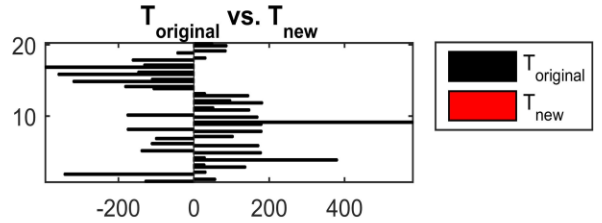

E original vs. E new, Phase2
1
0.5
0
500
1000
1500
H original
H new

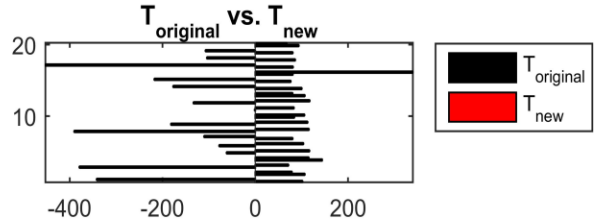

E original vs. E new, Phase3
1
0.5
0
500
1000
1500
H original
H new

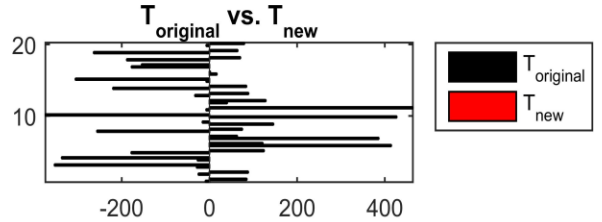

E original vs. E new, Phase4
1
0.5
0
500
1000
1500
H original
H new

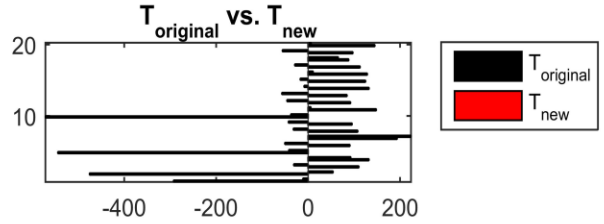

E original vs. E new, Phase5
1
0.5
0
500
1000
1500
H original
H new


T original vs. T new
20
10
-600
-400
-200
0
200
T original
T new


T original vs. T new
20
10
-200
0
200
400
T original
T new


T original vs. T new
20
10
-400
-200
0
200
T original
T new


T original vs. T new
20
10
-200
0
200
400
T original
T new


T original vs. T new
20
10
-400
-200
0
200
T original
T new

ShiftNMF Statistics

- R^2 = (SST – SSE)/SST, Cost = Least Squares Cost

Algorithm	Input Data	R^2 statistic	Final LS Cost	Number of Iterations
Original	1	1.0000	0.0112	2000
New		1.0000	0.0176	
Original	2	0.9937	2.02	3000
New		0.9909	2.99	
Original	3	0.9904	14.09	1000
New		0.9965	2.40	

ShiftNMF Statistics

- 150 iterations, 50 for each type of input phases **E**, randomized **P** and **T** matrices
  - Took difference in final R^2 values of each algorithm
  - Also counted number of runs where the algorithms converged to $R^2 > 0.99$
  - Null hypothesis: R^2 values and number of runs with $R^2 > 0.99$ are equal

Data Used	Value Observed	Type of Statistical Test	T-statistic	P-value
Mean of difference between R^2 values	0.040	2-sample t-test with unequal variances	0.1208	0.452
Difference in number of runs where $R^2 > 0.99$	8/150	2-sample proportion t-test	0.0419	0.4833

- Note: New ShiftNMF version works better for more complex/noisy inputs (more realistic) while original algorithm performs better with smoother data

ShiftNMF Reproducibility

- Ran 10 trials for each input data set, each with exact same **E**, **P**, and **T** inputs and initializations for ShiftNMF
- Maximum standard deviation of R^2 for any input data set → 2.3e-16
- Maximum R^2 difference between any two trials → less than 1e-15
- Reproduces same result given with same initialization close to machine error

Future Work

- My version of ShiftNMF runs twice as slow as previous authors' code
  - Must increase efficiency of algorithm
- Must replace current NMF steps of GRENDEL with ShiftNMF
- Align ShiftNMF with Graph Cut
  - We wish to change Graph Cut's objective function
  - ShiftNMF allows Graph Cut to not be run iteratively
  - Testing proper order of spectral clustering, ShiftNMF, and Graph Cut
  - Ensure Gibbs' Phase Rule is applied to ShiftNMF
- Create looping mechanism to ensure convergence
  - Stop ShiftNMF and restart if convergence is to an incorrect local minimum
  - Must weight accuracy with trade-off in extra CPU time

Timeline/Milestones (OLD)

- Fully understand, replicate previous code/results – mid/late October
- Phase 1 – Constraint Programming
  - Add connectivity constraints, expert prior knowledge for given samples - November
  - Add constraints for peak shifting - January
  - Potential addition of other physical laws, Mixed Integer Programming - February
- Phase 2 – Active Learning (Time permits)
  - Have algorithm to predict next best point to sample – March
  - Optimize the sampling algorithm for one material – mid April
  - Optimize algorithm for all material data given – late April

Timeline/Milestones (Final Revision)

- Fully understand, replicate previous code/results – mid/late October
- Stage 1 – Connectivity Constraint
  - Write Cannot Link algorithm – November
  - Validate and optimize parameters – December
- Stage 2 – Peakshifting Constraint
  - Locate and understand algorithm, ShiftNMF – January
  - Write ShiftNMF algorithm – February
  - Validation – March
- Stage 3 – Optimization of GRENDEL
  - Develop method to integrate ShiftNMF with Graph Cut - April
  - Collect final results, decrease run time of algorithm - May

Deliverables

- Final code/algorithm
- Results for given materials
  - Phase diagrams
  - Spectral graphs
  - Constituent phase compositions
- End of the year report and presentation

Bibliography

- LeBras R., Damoulas T., Gregoire J.M., Sabharwal A., Gomes C.P., and van Dover R.B., 2011. Constraint reasoning and kernel clustering for pattern decomposition with scaling. AAAI. CP'11: pp.508-522.
- Kusne A.G., Keller D., Anderson A., Zaban A., and Takeuchi I., 2015. High-throughput determination of structural phase diagram and constituent phases using GRENDEL. Nanotechnology. 26(44): pp. 444002.
- Ermon S., LeBras R., Suram S., Gregoire J.M., Gomes C.P., Selman B., and van Dover R.B., 2015. Pattern decomposition with complex combinatorial constraints: application to materials discovery. AAAI Conference on Artificial Intelligence. Available at http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/10020
- Hastie T., Tibshirani R., and Friedman J., 2013. *The Elements of Statistical Learning - Data Mining, Interference, and Prediction*. ed. 2 (Berlin: Springer).
- Settles B., 2012. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning #18 (Morgan & Claypool).
- Kan D., Suchoski R. Fujino S., Takeuchi I., 2009. Combinatorial investigation of structural and ferroelectric properties of A- and B- site co-doped BiFeO3 thin films, Integrated Ferroelectrics. 111: pp. 116-124.
- Takeuchi I., 2016. Data Driven Approaches to Combinatorial Materials Science. Materials Research Society Spring Meeting (presentation).
- Zare A., Gader P., Bchir O., and Frigui H., *Piecewise Convex Multiple-Model Endemember Detection and Spectral Unmixing,* IEEE Transactions on Geoscience and Remote Sensing **51** (2013), no. 5: 2853-2862.
- Boykov Y. and Kologorov V., *An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision*, IEEE Transactions on PAMI, **26** (2004), no. 9: 1124-1137.
- Morup M., Madsen K. H., and Hansen L. K., *Shifted Non-negative Matrix Factorization*, IEEE International Workshop on Machine Learning for Signal Processing, (2007): pp. 139-144.
- Xue Y., Bai J., Le Bras R., Rappazzo B., Bernstein R., Bjork J., Longpre L., Suram S., van Dover R., Gregoire J., and Gomes C., *Phase-Mapper: An AI Platform to Accelerate High Throughput Material Discovery*, CoRR, **1610** (2016).
- Suram S., Xue Y., Bai J., Le Bras R., Rappazzo B., Bernstein R., Bjorck J., Zhou L., van Dover R., Gomes C., and Gregoire J., *Automated Phase Mapping with AgileFD and its Application to Light Absorber Discovery in the V-Mn-Nb Oxide System*, arXiv:1610.02005 (2016).
- Information about White House Genome Initiative courtesy of https://www.whitehouse.gov/mgi