

Contents

1 Radial Density Function	2
1.1 Calculation of Distances with Periodicity	2
1.2 Adding Noise For Atom Vibration	3
1.3 Cubane Example	3
1.3.1 Cubane Radial Density Functions	4
1.4 Experimental and Theoretical RDFs for Known Structures	6
1.4.1 Ga As	6
1.4.2 In As	7
1.4.3 Si Lattice	7
2 Smoothing Analysis	8
3 Matching Algorithm Evaluation	11
3.1 Noise Analysis	13
3.1.1 Peak Counts	14
3.1.2 Peak Locations	16
3.1.3 Noise Peak Heights	17
3.1.4 Sample Noisy Image	19
4 Recognition Using Eigenfaces	21
4.1 Mean Image	21
4.2 Variance Explained by Principal Components	23
4.3 Eigenfaces	24
4.4 Data in Eigenspace	27
4.4.1 Eigenspace Outliers	32
4.5 Experimental Image Recognition	37
4.5.1 3 Principal Components	37
4.5.2 10 Principal Components	48
4.5.3 128 Principal Components	59
4.6 Synthetic Experimental Image Recognition	70

5	Recognition Using Sparse Representations	73
5.1	Experimental Image Recognition	74
5.2	Synthetic Experimental Image Recognition	85
6	Code	86
7	Sources	86

1 Radial Density Function

1.1 Calculation of Distances with Periodicity

Suppose a large chemical structure has uncountably many atoms but the follow a periodic pattern of n atoms every p Angstroms. The atom locations within a period are given by a_1, a_2, \dots, a_n where $a_i \in \mathbb{R}^3$. The radial density function is the distribution of pairwise distances between these atoms.

The distances d between atoms a_i and a_j where $i \neq j$, atom a_i has been displaced by x , and atom a_j has been displaced by y per the periodicity is

$$\begin{aligned} d^2 &= \langle a_i + x - (a_j + y), a_i + x - (a_j + y) \rangle \\ &= \langle a_i - a_j, a_i - a_j \rangle + \langle x - y, x - y \rangle + 2\langle a_i - a_j, x - y \rangle \end{aligned}$$

where $x = (k_1 p, k_2 p, k_3 p)$ for $k_i \in \mathbb{Z}$ and $y = (l_1 p, l_2 p, l_3 p)$ for $l_i \in \mathbb{Z}$. Here $\langle x, y \rangle$ denotes the inner product between x and y .

Suppose D is a random variable that samples at random the distances, d , in the chemical structure. The radial density function is the probability density function of this random variable. This function can be estimated empirically via a histogram.

The histogram is then normalized by the volume a spherical shell.

$$\begin{aligned} &\frac{4}{3}\pi(r + \Delta r)^3 - \frac{4}{3}\pi r^3 \\ &= \frac{4}{3}(3r^2\Delta r + 3r(\Delta r)^2 + (\Delta r)^3) \\ &\approx 4\pi r^2 \Delta r \end{aligned}$$

where Δr tends to zero.

For a histogram with frequency, f , for bin $[d_i, d_{i+1}]$, we replace f with f/d_i^2 . And then normalize the histogram so that the sum over all bins is one.

1.2 Adding Noise For Atom Vibration

Due to the vibrations of the molecules, the radial density function will not be just the equilibrium positions. We can approximate this fluctuation in distances via a Gaussian filter or Weierstrass transform.

$$F(x) = \frac{1}{\sqrt{4\pi t}} \int_{-\infty}^{\infty} f(y) e^{-\frac{(x-y)^2}{4t}} dy$$

Given that the density function is only defined for a finite number of distances, we use a discrete version of the transform making sure to keep the sum of the weights equal to one.

$$F(d_k) = \frac{\sum_{d_i=d_0}^{d_n} f(d_i) \exp\left(-\frac{(d_k-d_i)^2}{4t}\right)}{\sum_{d_i=d_0}^{d_n} \exp\left(-\frac{(d_k-d_i)^2}{4t}\right)}$$

where d_0 is the minimum distance and d_n is the maximum distance.

1.3 Cubane Example

As an example of the above, below are the calculations for cubane (C_8H_8).

Here are the coordinates of the elements in cubane in Angstroms.

```
Element, x, y, z
C, 1.2455, 0.5367,-0.0729
C, 0.9239,-0.9952, 0.0237
C,-0.1226,-0.7041, 1.1548
C, 0.1989, 0.8277, 1.0582
```

C, 0.1226, 0.7042,-1.1548
C,-0.9239, 0.9952,-0.0237
C,-1.2454,-0.5367, 0.0729
C,-0.1989,-0.8277,-1.0582
H, 2.2431, 0.9666,-0.1313
H, 1.6638,-1.7924, 0.0426
H,-0.2209,-1.2683, 2.0797
H, 0.3583, 1.4907, 1.9059
H, 0.2208, 1.2681,-2.0799
H,-1.6640, 1.7922,-0.0427
H,-2.2430,-0.9665, 0.1313
H,-0.3583,-1.4906,-1.9058

1.3.1 Cubane Radial Density Functions

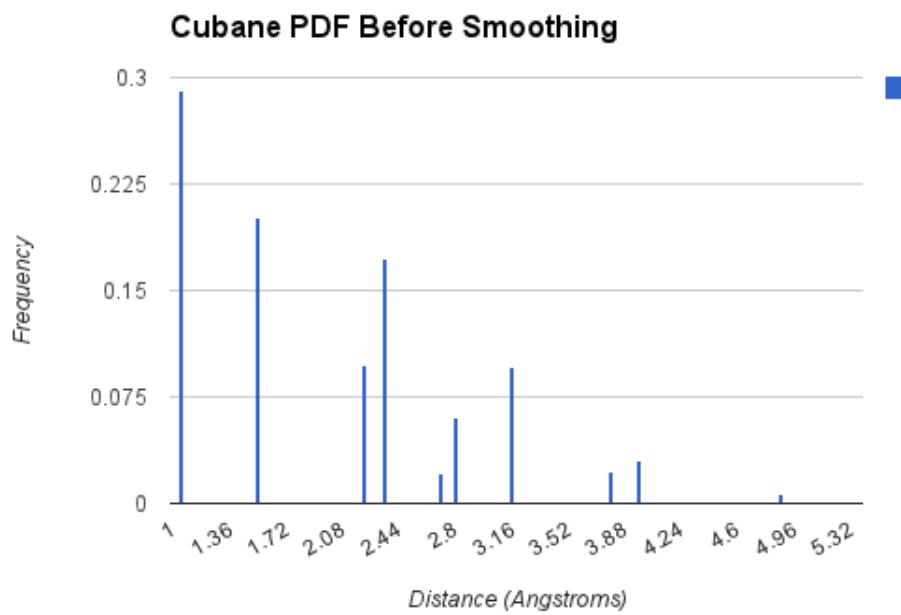


Figure 1: Before Smoothing

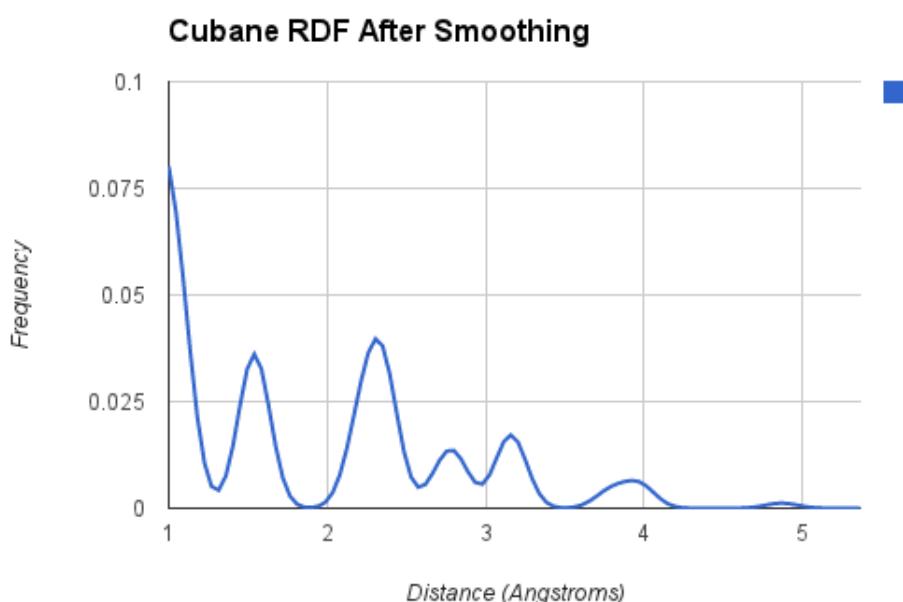


Figure 2: After Smoothing

1.4 Experimental and Theoretical RDFs for Known Structures

For some structures, we are able to theoretically calculate the RDF from atom locations and also have the experimental RDF from Xray scattering. These known matches provide some insight into understanding how the experiments and theory align. The RDF comparison are shown below.

Outside of these structures, there are not many other known matches. There are a few reasons for this. First, if a structures is already known at the atomic level then there is no need to run an xray diffraction experiment. Second, if a structure is periodic as in a lattice, the atomic structure can be determined by xray diffraction which is easier and cheaper than xray scattering.

1.4.1 Ga As

Experimental Data: Pair Distribution Functions Analysis, Valeri Petkov

Calculated Data: Maria Chan

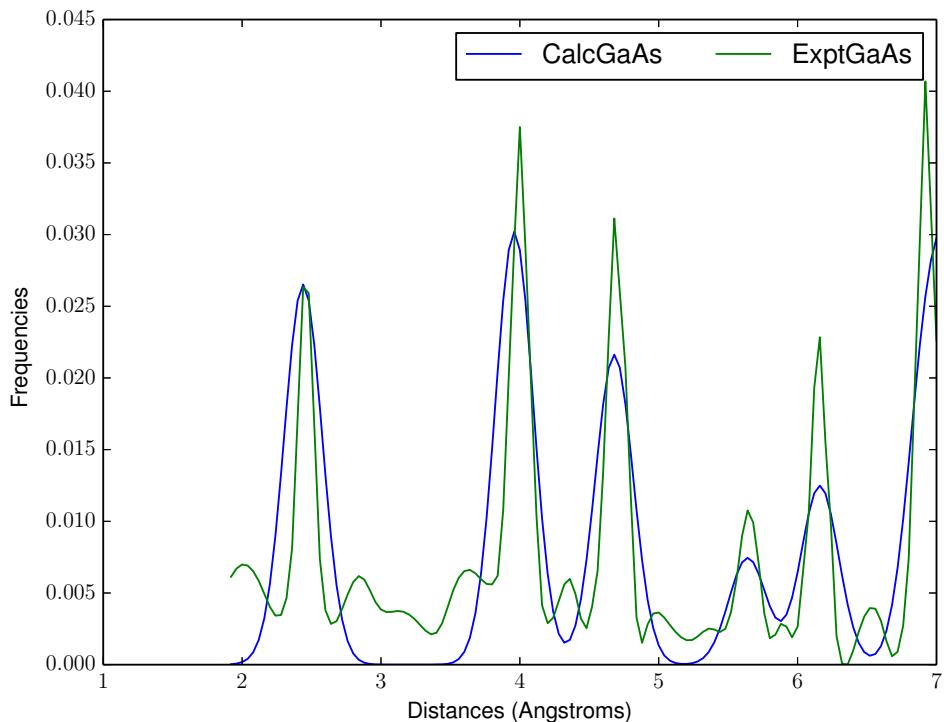


Figure 3: Ga As

1.4.2 In As

Experimental Data: Pair Distribution Functions Analysis, Valeri Petkov

Calculated Data: Maria Chan

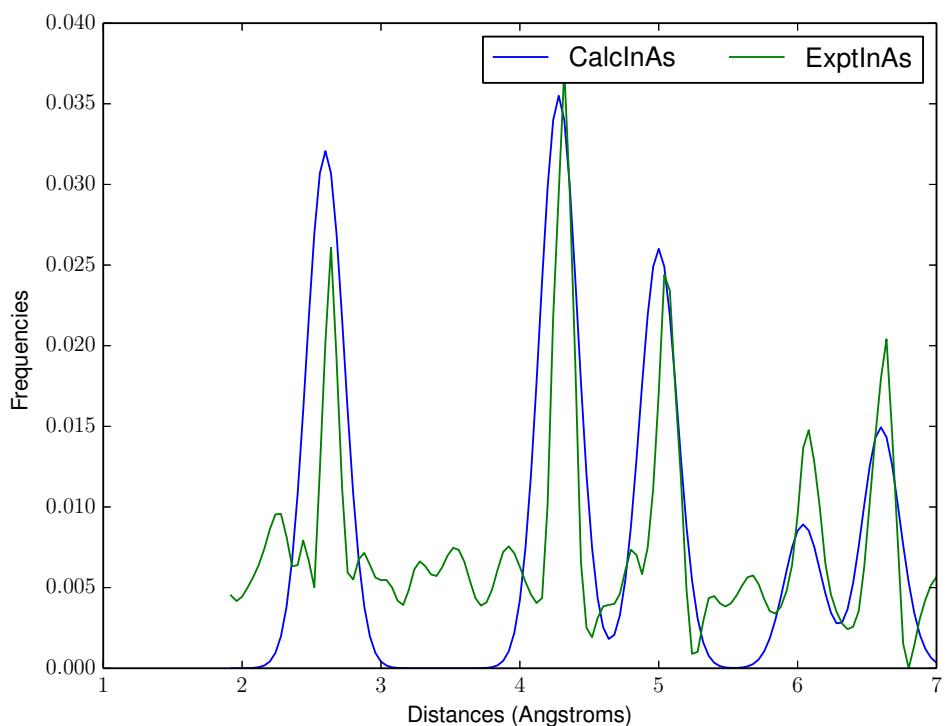


Figure 4: In As

1.4.3 Si Lattice

Experimental Data: J. AM. CHEM. SOC. VOL. 133, NO. 3, 2011, P: 503-512

Calculated Data: <http://materialsproject.org/materials/mp-149/>

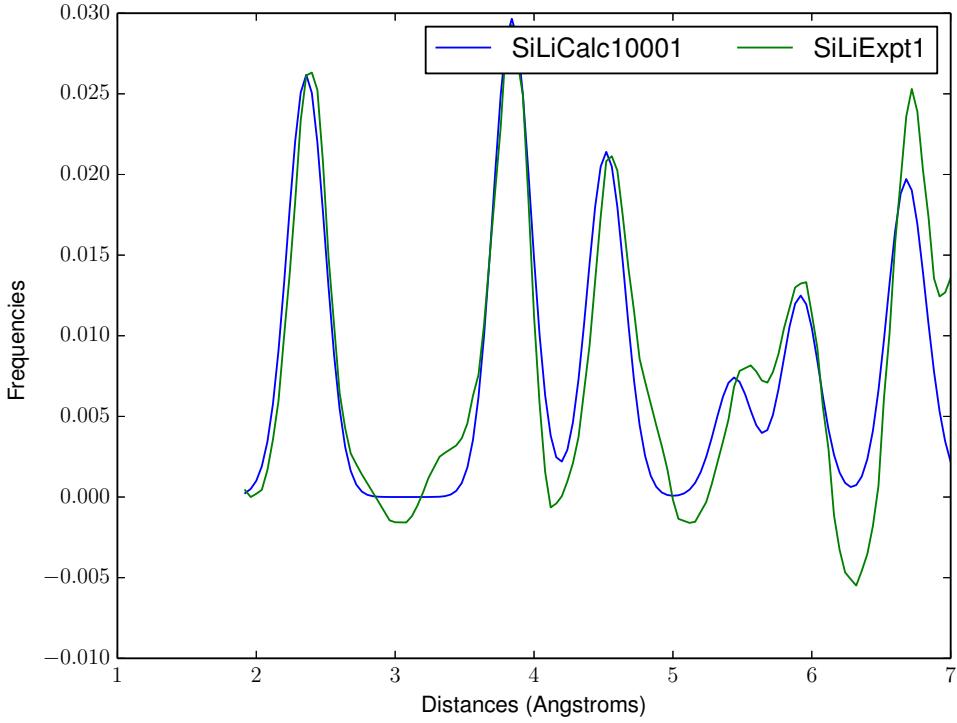


Figure 5: Si Lattice

2 Smoothing Analysis

Consider a function, $\text{SmoothAndNormalize}(i, t)$, that smoothes the image, i , with a smoothing coefficient of t and then normalizes the smoothed image so that the weights sum to one. To calibrate the smoothing coefficient, we focus on the SiLi calculated and experimental matches, SiLiCalc10001 and SiLiExpt1. We want to find the smoothing coefficient, t , that after smoothing and normalization the calculated image is the closest match to the experimental image.

$$\hat{t} = \arg \min_t \|X - \text{SmoothAndNormalize}(C, t)\|_2$$

where $\|\cdot\|_2$ is the L^2 norm, X is the experimental image, and C is the calculated image.

We found that $\hat{t} = 0.0092$.

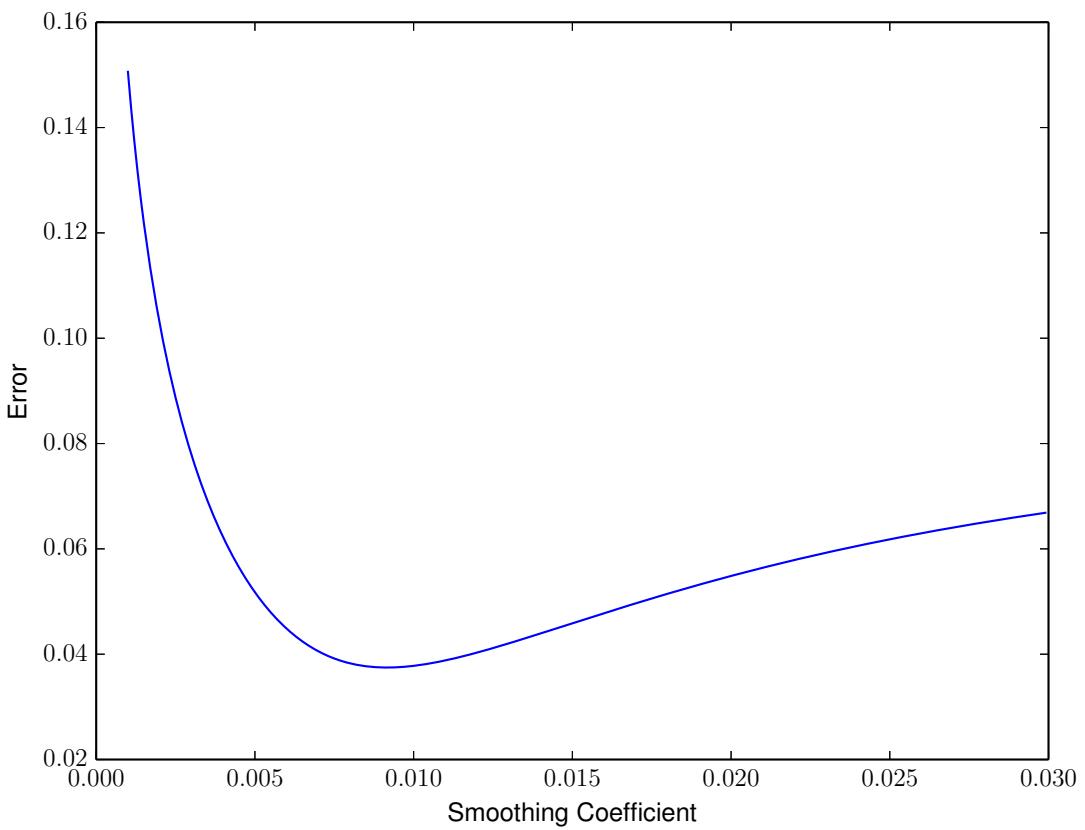


Figure 6: Smoothed - Expt Error vs Smoothing Coefficients

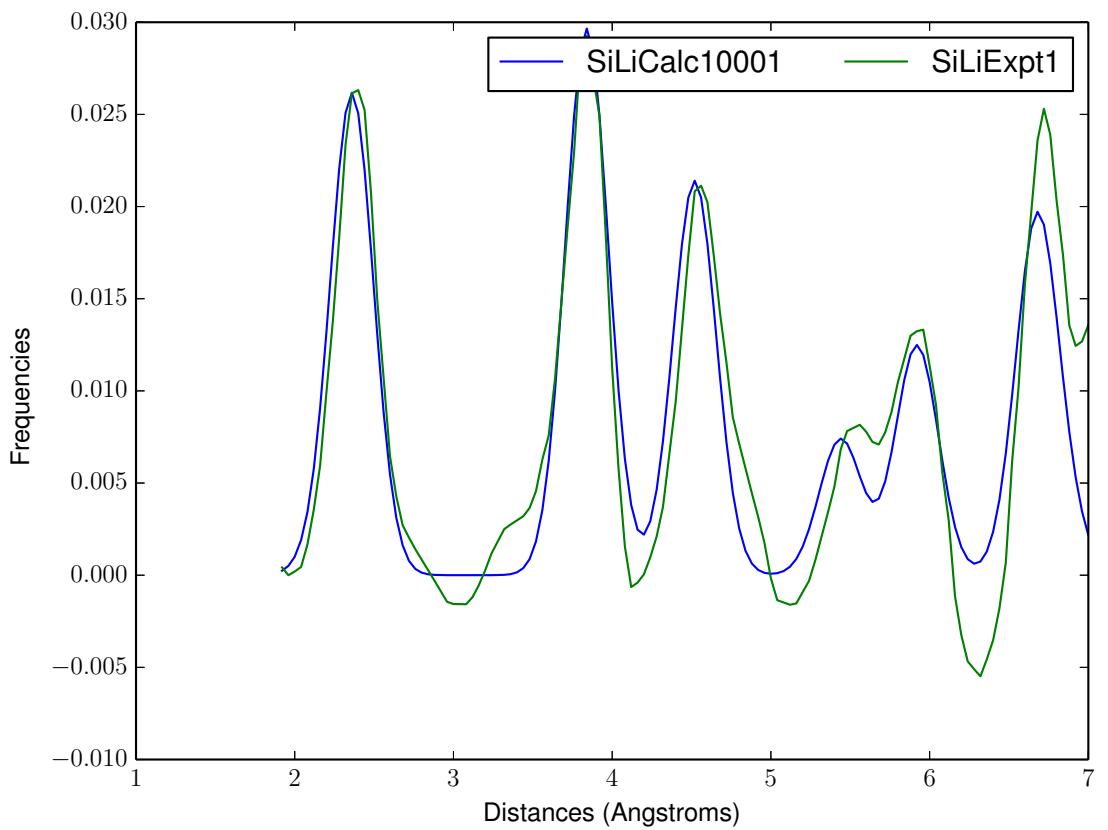


Figure 7: Smoothed SiLiCalc10001 vs SiLiExpt1

3 Matching Algorithm Evaluation

The goal of this project is to invent an algorithm that will find the best matching calculated image for a given experimental image. A more audacious goal is to find an algorithm that can decompose experimental image into a linear combination of a few calculated image.

Before running an experiment, it is possible to theoretically predict the feasible structures that could occur during the experiment. Given this matching algorithm, during the experiment, we could match the experimental results back to the theoretically predicted structures. This would give an 'x-ray vision' into the structures, being able to see that atom locations as the experiment progresses.

One way to evaluate the performance of a proposed matching algorithm is to use the set of known experimental and calculated matches. Taking the set of calculated images as a whole, the algorithm should be able to recover the calculated match given the experimental image.

The problem with this evaluation metric is that there are only three known calculated/experimental matches. This is not a large enough number of samples for good statistics. The reason there are not too many matching pairs is because the known structures at this time are periodic and can be observed experimentally through cheaper x-ray diffraction experiments. Non-periodic structures which are the focus of this project are studied precisely because their exact structures are not known.

An alternative approach to using the calculated and experimental matching pairs directly is to simulate experimental images by adding random noise to a calculated image. The goal then becomes to recover the original calculated image given the simulated experimental image.

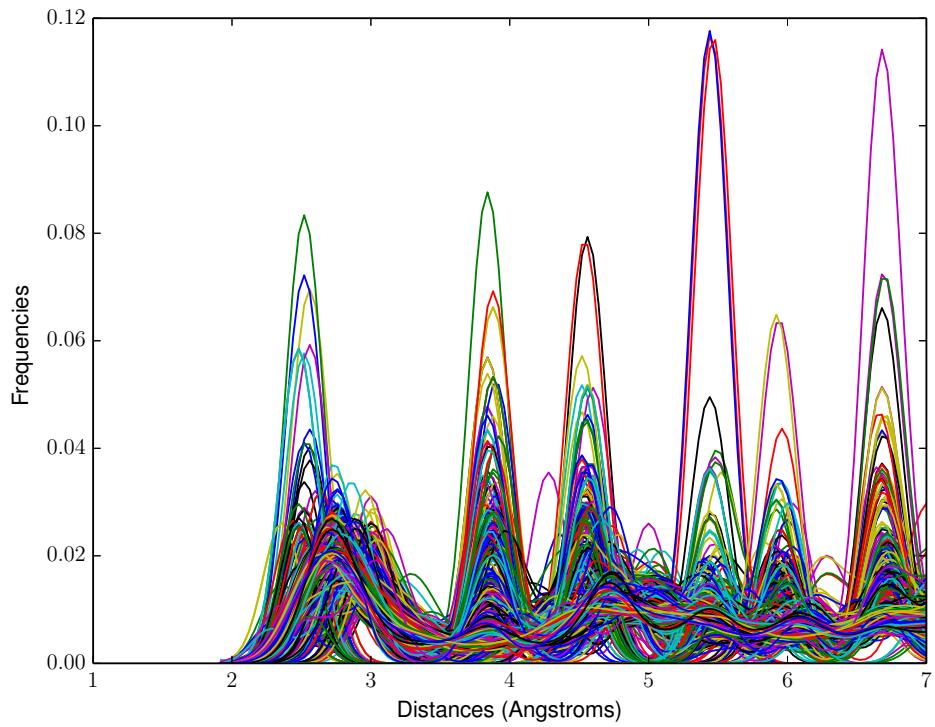


Figure 8: All Calculated Images

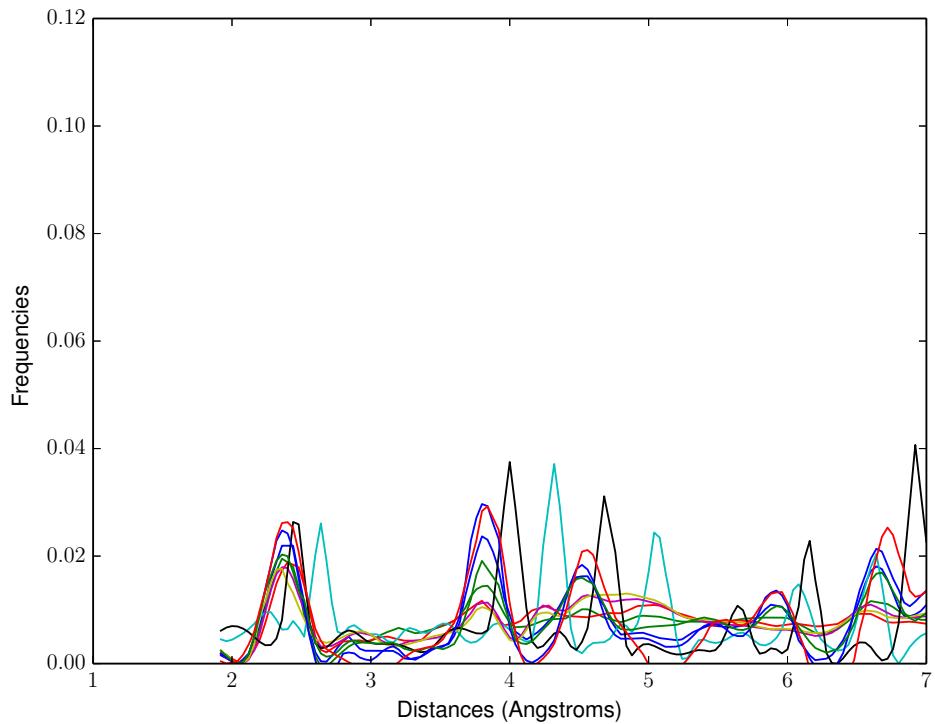


Figure 9: All Experimental Images

3.1 Noise Analysis

The experimental images present three different varieties of noise compared to the calculated images.

- Tilt: Consider the baseline of the image to be the piecewise line connecting the valleys of the image. The experimental images seem to have a non-zero and slanted baseline compared to the calculated images which have a baseline at zero.
- Noise: Between the major peaks of the experimental images, there are also several minor peaks. These minor peaks are not found in the calculated images.
- Peak Heights Difference: The heights of the major peaks vary between the calculated and experimental images.

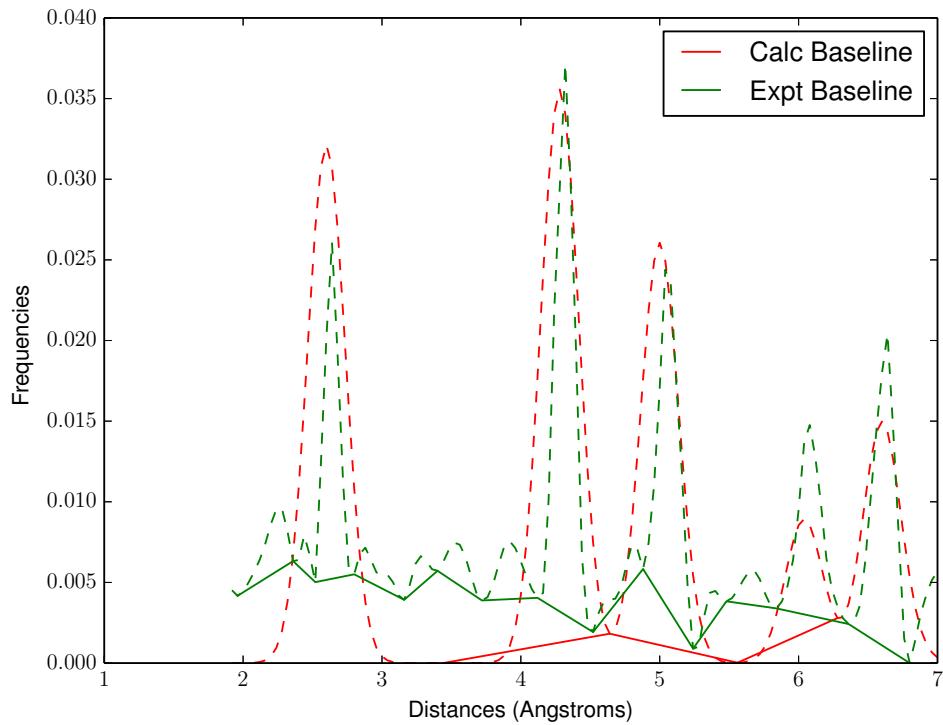


Figure 10: In As

To simulate these features, I first estimated the distributions of the number of peaks, peak locations, and the noise peak heights.

3.1.1 Peak Counts

To estimate the distribution of number of peaks per image, I took all of the experimental images and estimated the number of peaks. Then I visually inspected the histogram to estimate the distribution.

From the histogram, I concluded that the number of peaks is uniformly distributed between 7 and 15.

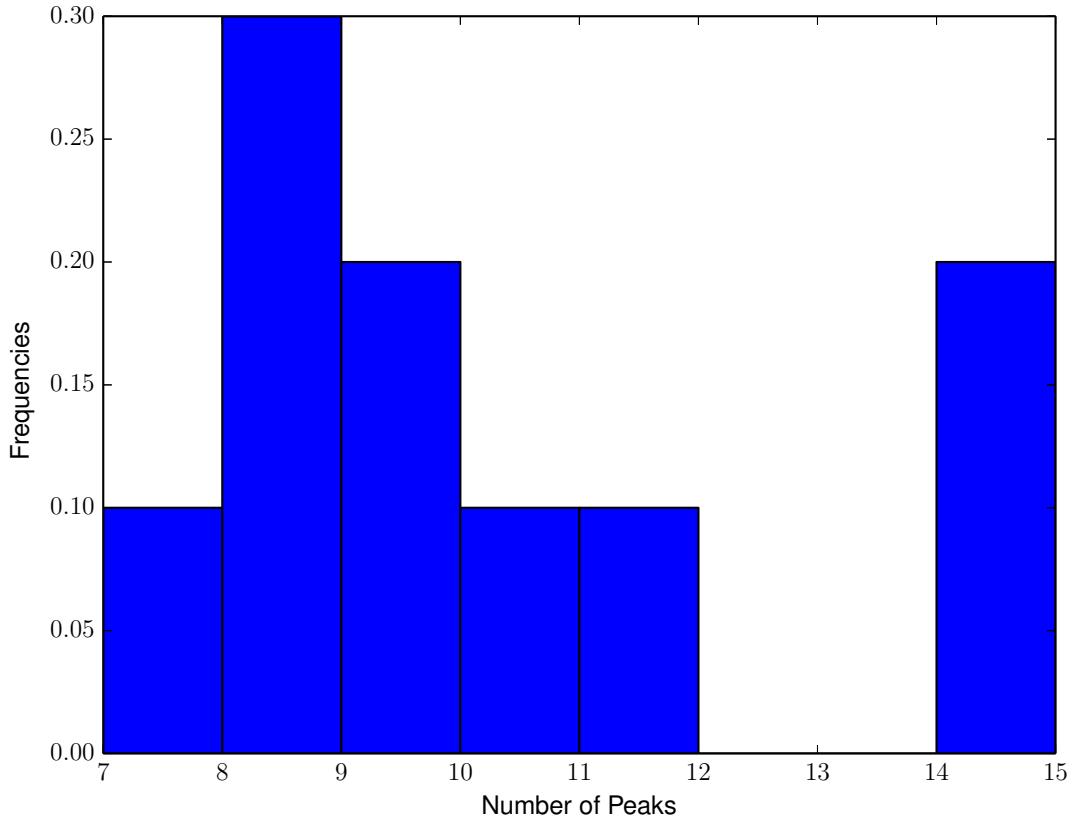


Figure 11: Peak Count Distribution

Note that the estimation is strongly dependent on the maximum length of the image. The raw InAs spectrum goes out to 10 angstroms and has much more than 15 peaks.

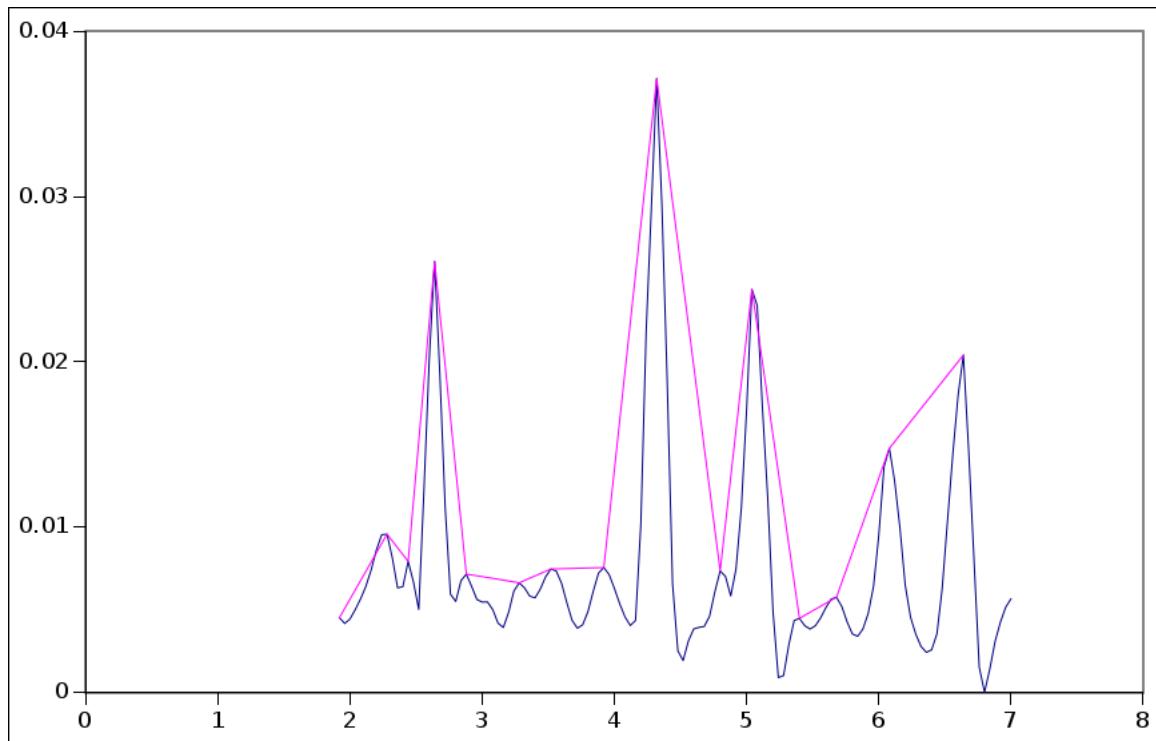


Figure 12: InAs Expt, Max 7 Angstroms

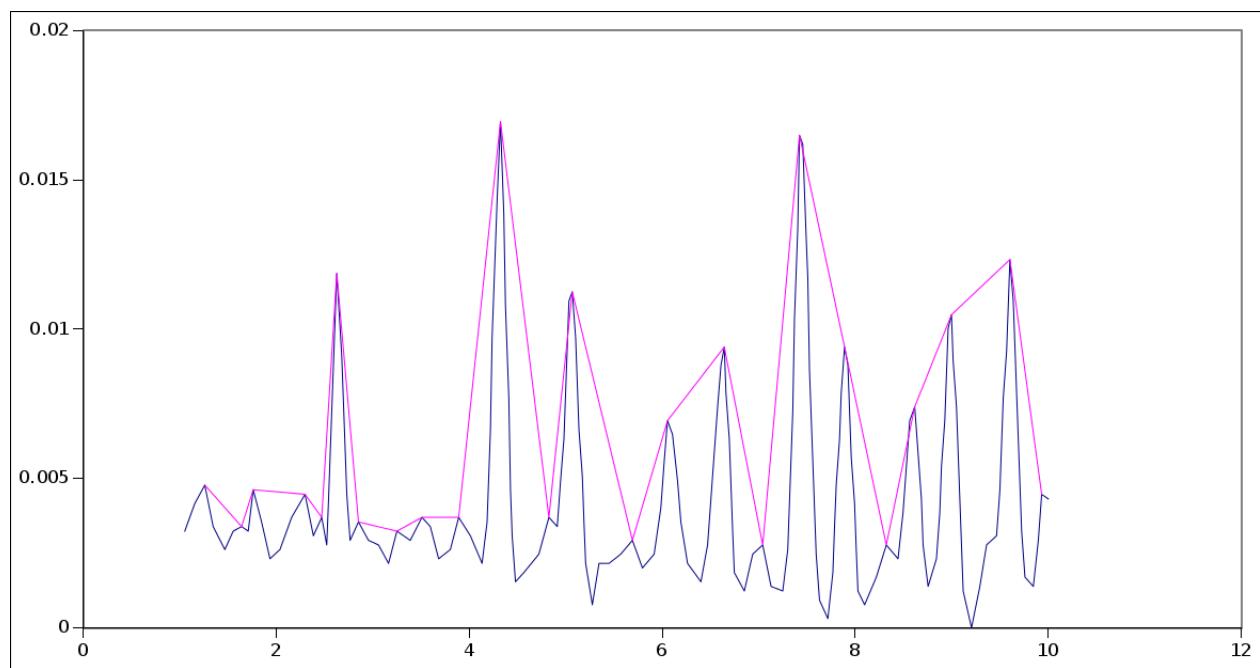


Figure 13: InAs Expt, Max 10 Angstroms

3.1.2 Peak Locations

To estimate the distribution of the location of the peaks, I first took all of the experimental images and calculated the distances of all of the peaks. Then I charted this as a histogram to visually inspect the distribution.

From the histogram, I concluded that the locations are uniformly distributed between 1.96 and 7.

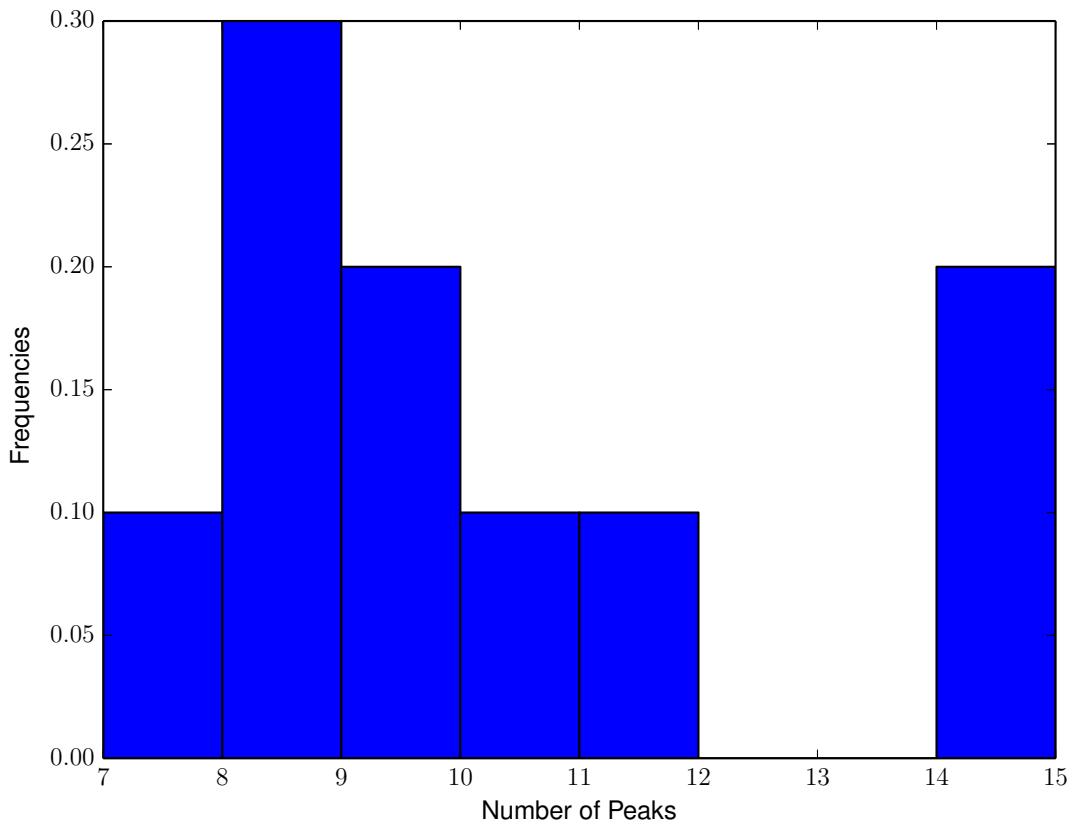


Figure 14: Peak Location Histogram

3.1.3 Noise Peak Heights

To add noise to the image, I random noise directly to the original frequencies. That is, $N = I + R$, where N is a vector of frequencies for the noisified image, I is for the original image, and R is for the random noise.

In light of this, to estimate how much noise to add, I first calculated $E = X - C$, where E is the error or noise to be added, X is the experimental image, and C is the calculated image. I considered the errors at different distances to be independent and thus considered all of the errors to be for any distance. Taking all of the errors together, I estimate the error distribution. From the histogram and cumulative distribution function, I concluded that the error distribution is most similar to a normal distribution. The estimate for the mean is 0.004 and standard deviation is 0.004.

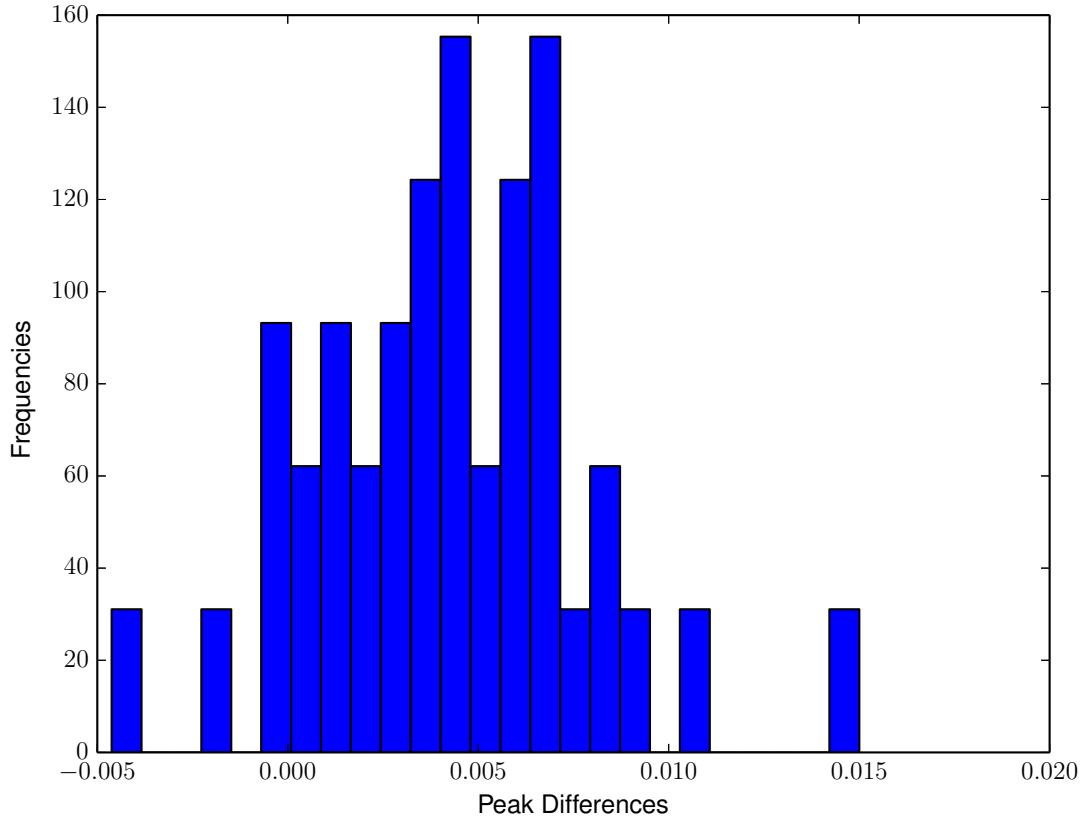


Figure 15: Noise Peak Heights Histogram

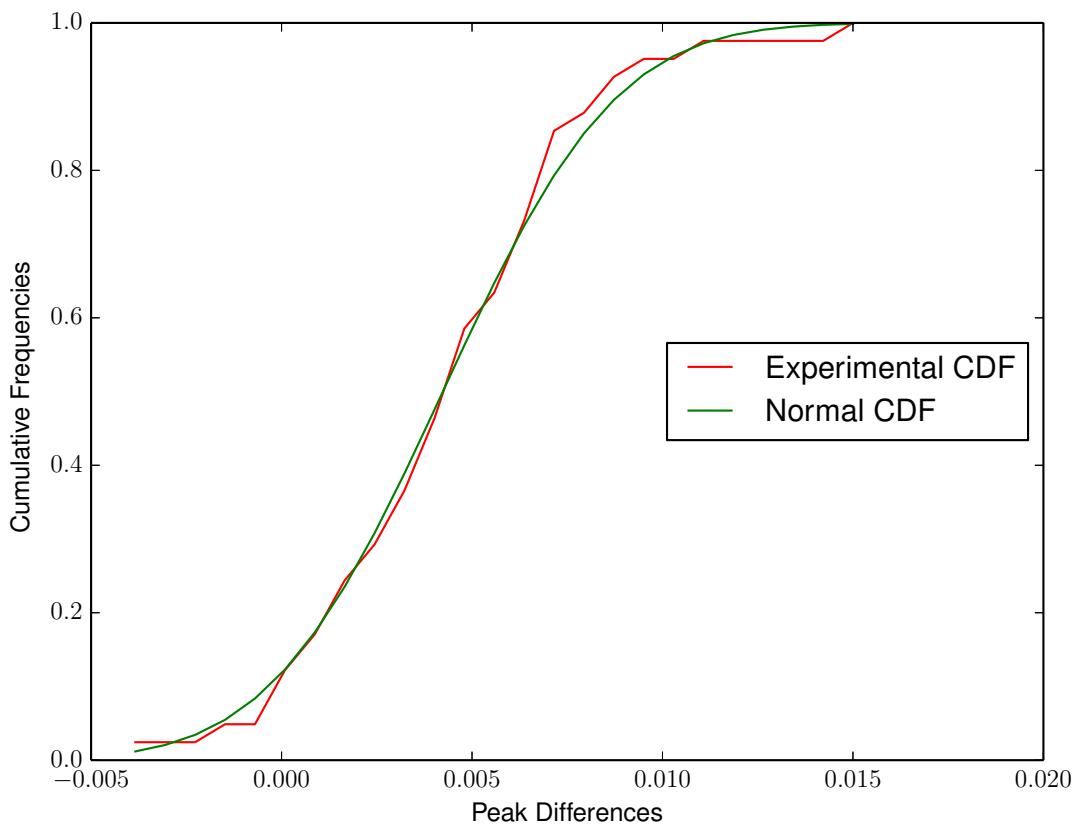


Figure 16: Noise Peak Heights Distribution

3.1.4 Sample Noisy Image

To generate the simulated experimental images, I first sample the number of peaks, the peak locations, and the peak heights from their respective distributions. Then this noise is added to the original image and the resulting image is re-normalized.

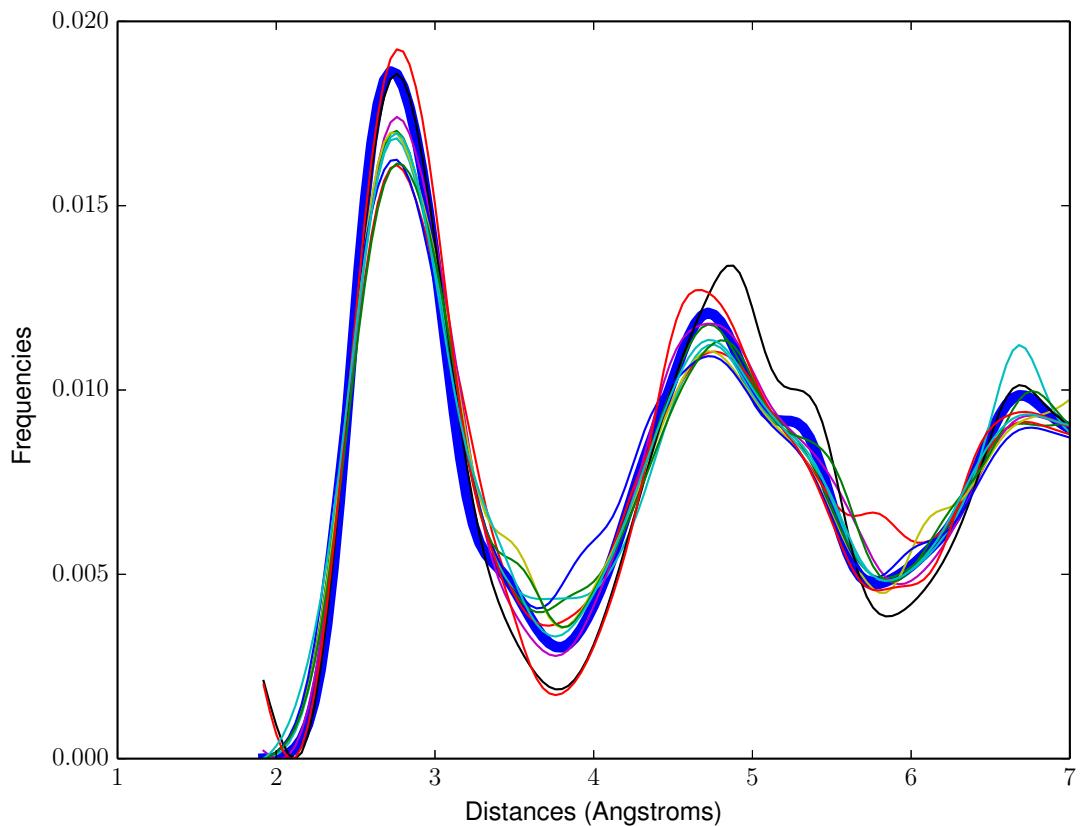


Figure 17: Simulated Experimental Images, 1x Standard Deviation

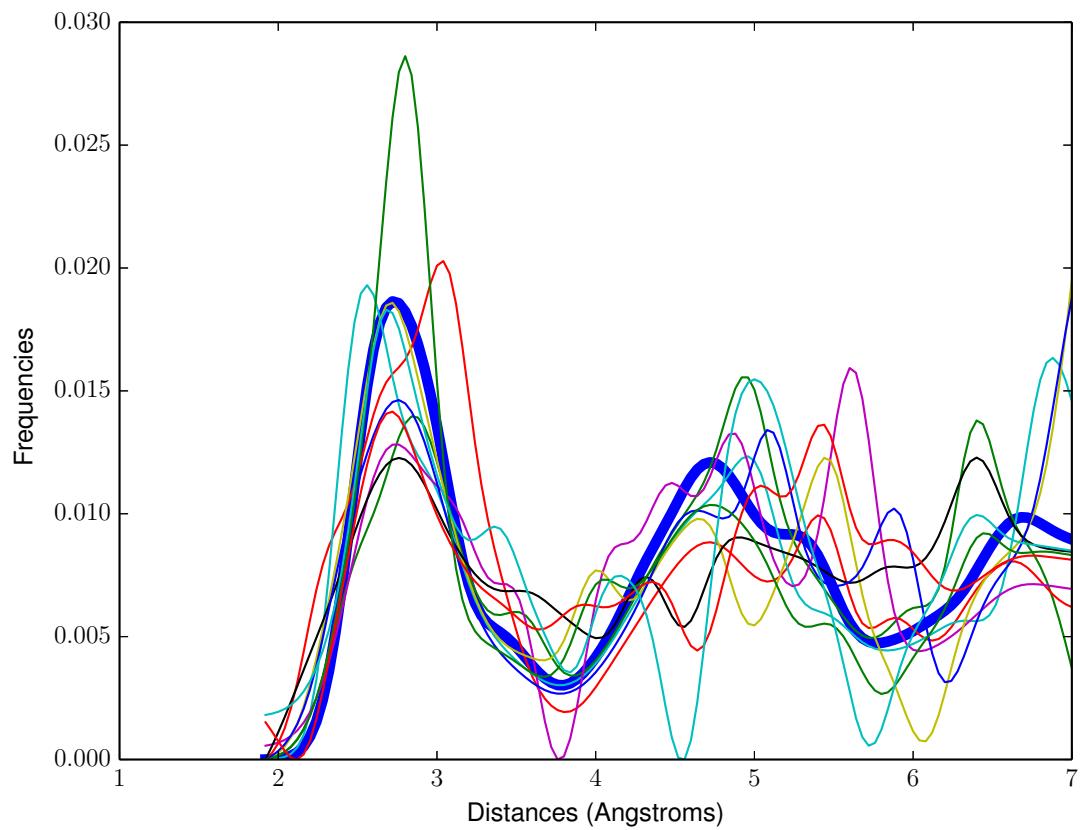


Figure 18: Simulated Experimental Images, 10x Standard Deviation

4 Recognition Using Eigenfaces

In this section, we apply the approach outlined in Turk and Pentland's "Eigenfaces for Recognition".

Let $C = [c_1 \dots c_n]^T$ be the calculated images and Ψ be the column-wise means of the C .

First compute the principal components of C by singular value decomposition.

$$\text{Cov}(C) \sim (C - \Psi)^T (C - \Psi) = W \Sigma^2 W^T$$

Here W are matrices that contain the loadings of C . Each column has different principal components and the rows span the dimensions of the images.

The first L loadings, T_L , are computed with the first L principal components. Let W_L be a matrix with the first L columns of W . Then $T_L = (C - \Psi)W_L$.

Suppose X is the target image that we are trying to find a best match for. First compute the loadings, S_L , for X as $S_L = (X - \Psi)W_L$. Then use the L^p norm to find the closest image in PCA space.

$$\hat{i} = \arg \min_i \|T_L(i, :) - S_L\|_p$$

4.1 Mean Image

Below shows the mean image, Ψ , over all of the calculated images.

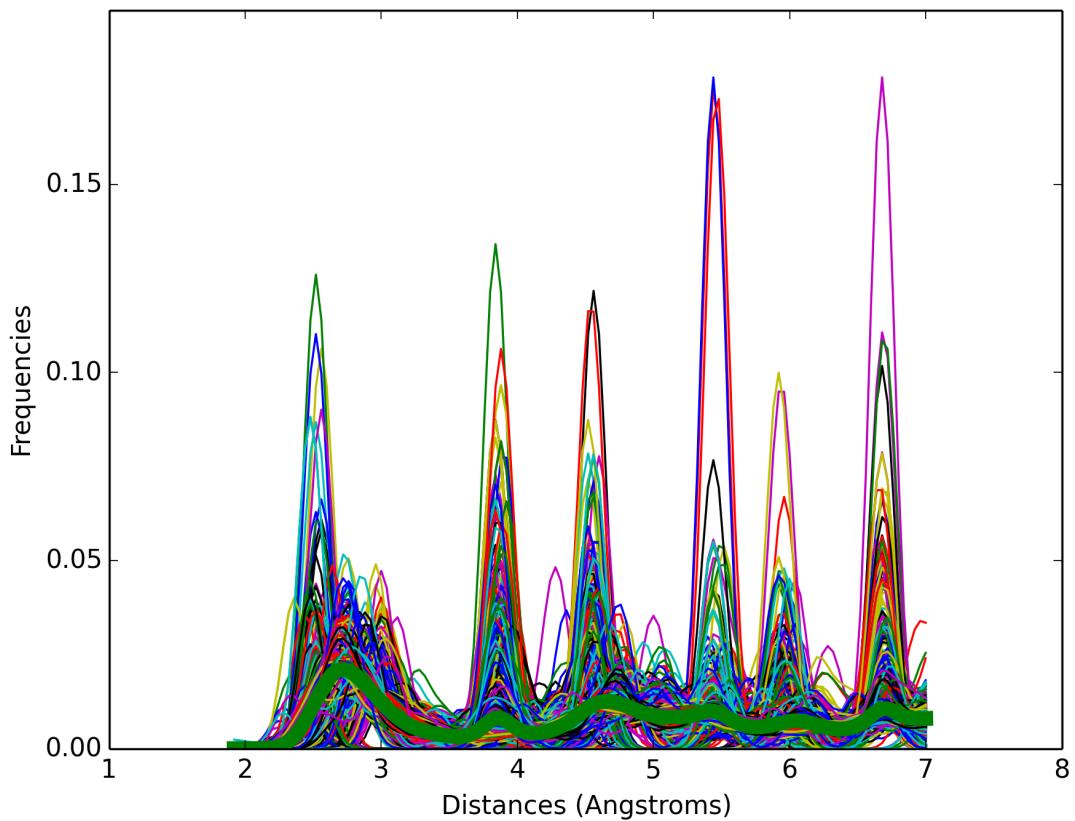


Figure 19: All Calculated Images with Mean

4.2 Variance Explained by Principal Components

Principal component analysis projects the data onto an orthogonal space. Thus in PCA space, we can consider the variance contributed by each of the principal components separately and can identify those principal components that contribute most to the variance of the data set. To identify how many principal components are needed to explain the majority of the variation in the data set, we can look at the cumulative variance.

$$C_k = \sum_{i=1}^k \sigma_i^2 / \sum_{i=1}^n \sigma_i^2$$

where C_k is the variance explained by k principal components and σ_i^2 is the variance of principal component i .

From the graph, we notice that around 15 principal components explain about 95% of the data.

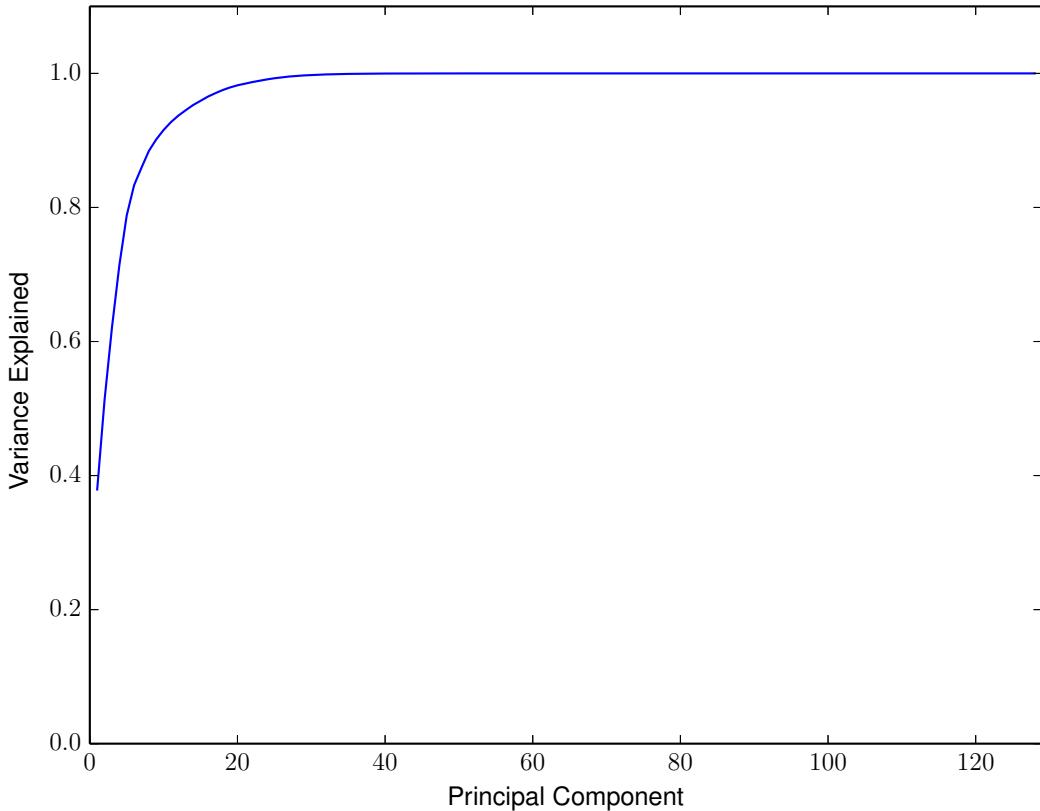


Figure 20: Cumulative Variance Explained by Principal Components

4.3 Eigenfaces

!!! Need to rerun all of these results with the changed smoothing coefficient!!! Here, we plot the eigenfaces to see if any key features of the images are revealed. Not much of anything is noteworthy.

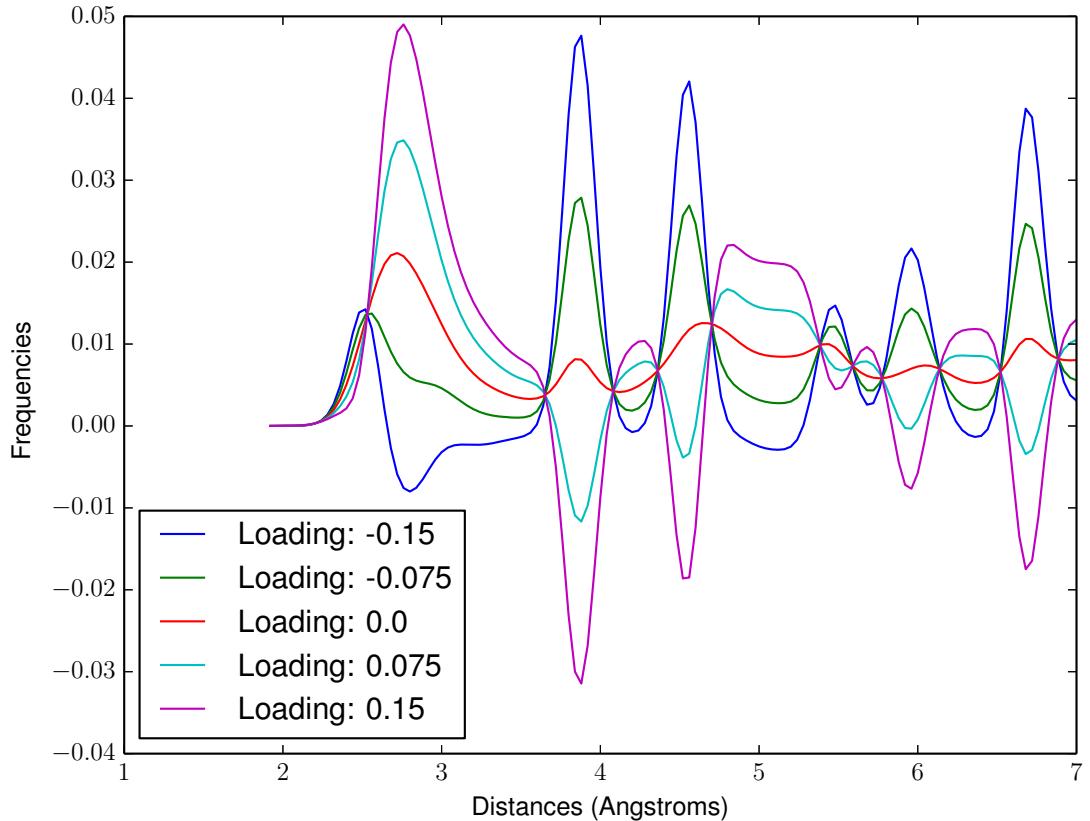


Figure 21: First Eigenface

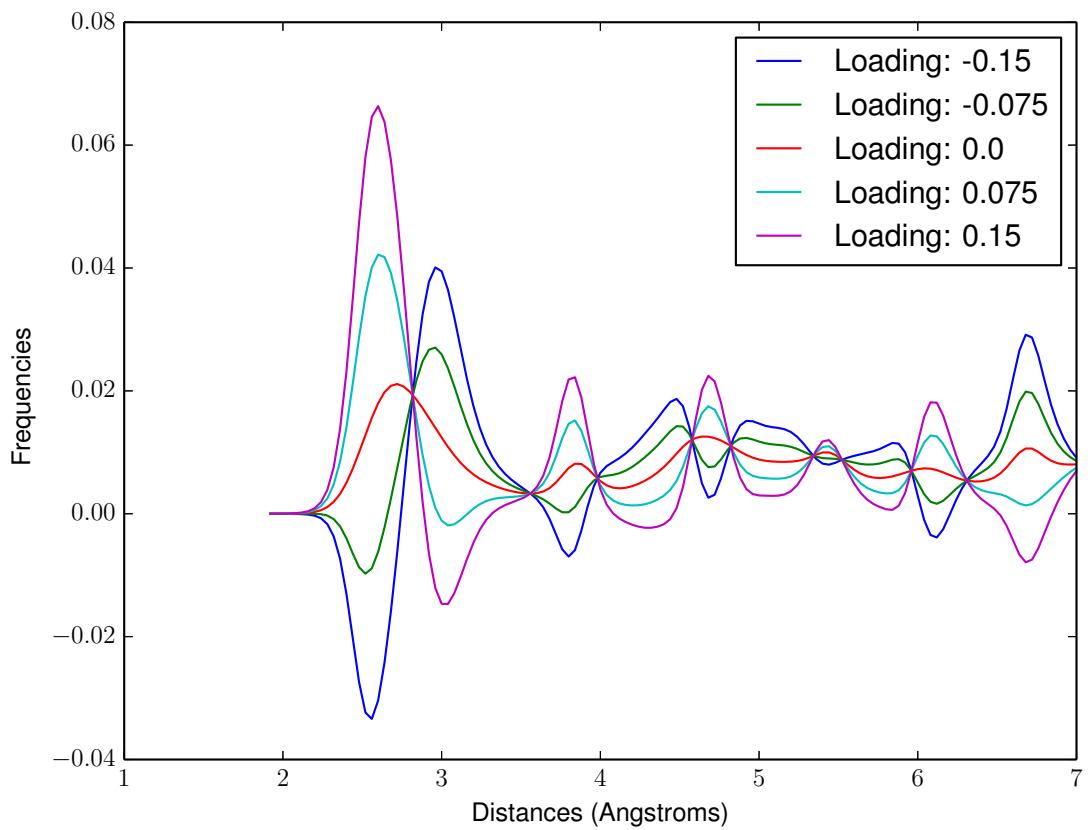


Figure 22: Second Eigenface

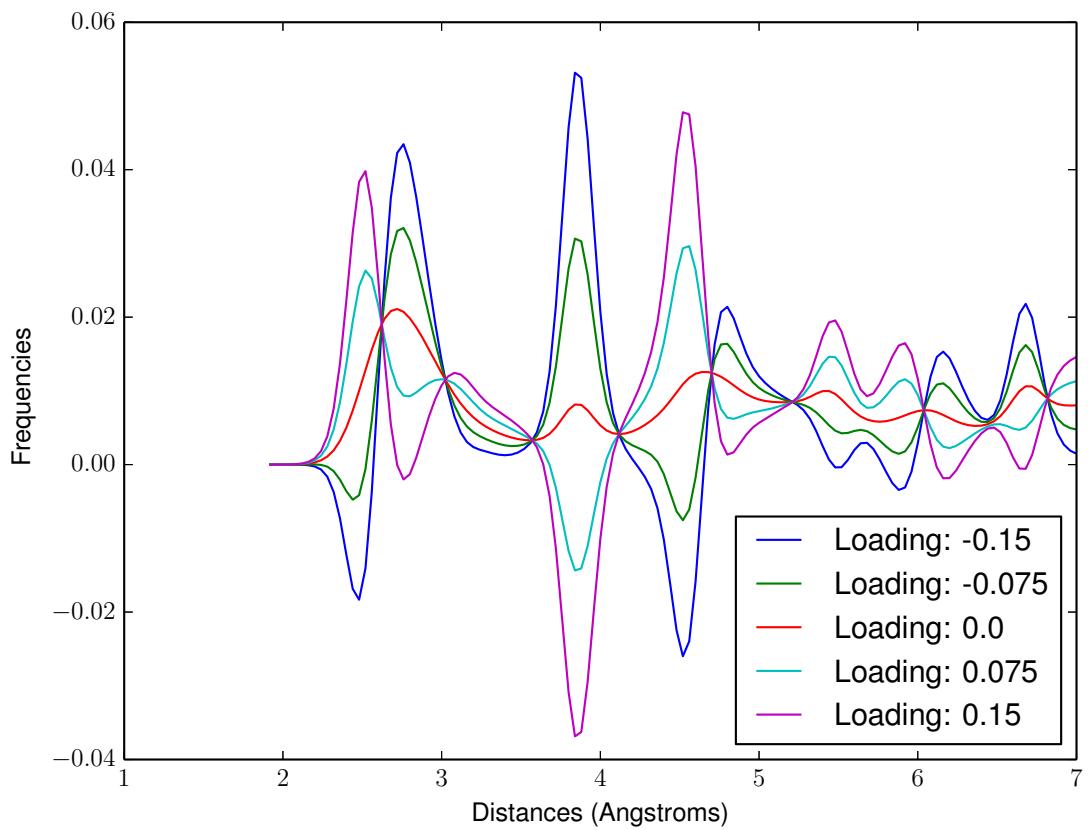


Figure 23: Third Eigenface

4.4 Data in Eigenspace

Below shows the data in principal component space for the most significant principal components.

One feature we observe is that the first principal component separates the experimental data very well and in fact appropriately sorts the silicon lithium structures by that amount of lithium. Higher principal components are useful for distinguishing the experimental images.

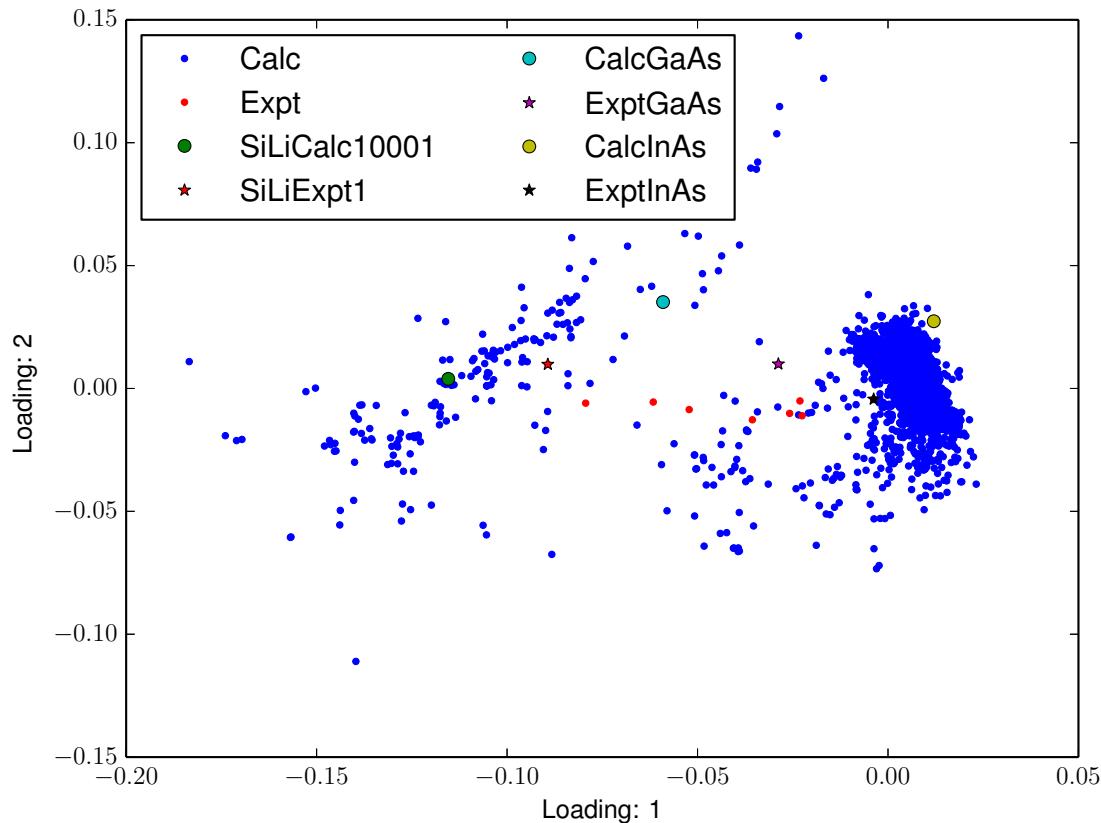


Figure 24: Loading 1 vs Loading 2

Label	Loading 1	Loading 2
SiLiExpt1	-0.0893	0.00982
SiLiExpt2	-0.0794	-0.006
SiLiExpt3	-0.0616	-0.0055
SiLiExpt4	-0.0522	-0.0086
SiLiExpt5	-0.0356	-0.0128
ExptGaAs	-0.0288	0.00994
SiLiExpt7	-0.0258	-0.0101
SiLiExpt6	-0.0231	-0.0051
SiLiExpt8	-0.0226	-0.0111
ExptInAs	-0.0038	-0.0044

Table 1: Experimental Data Sorted by Loading 1

Label	Loading 1	Loading 2
SiLiExpt5	-0.0356	-0.0128
SiLiExpt8	-0.0226	-0.0111
SiLiExpt7	-0.0258	-0.0101
SiLiExpt4	-0.0522	-0.0086
SiLiExpt2	-0.0794	-0.006
SiLiExpt3	-0.0616	-0.0055
SiLiExpt6	-0.0231	-0.0051
ExptInAs	-0.0038	-0.0044
SiLiExpt1	-0.0893	0.00982
ExptGaAs	-0.0288	0.00994

Table 2: Experimental Data Sorted by Loading 2

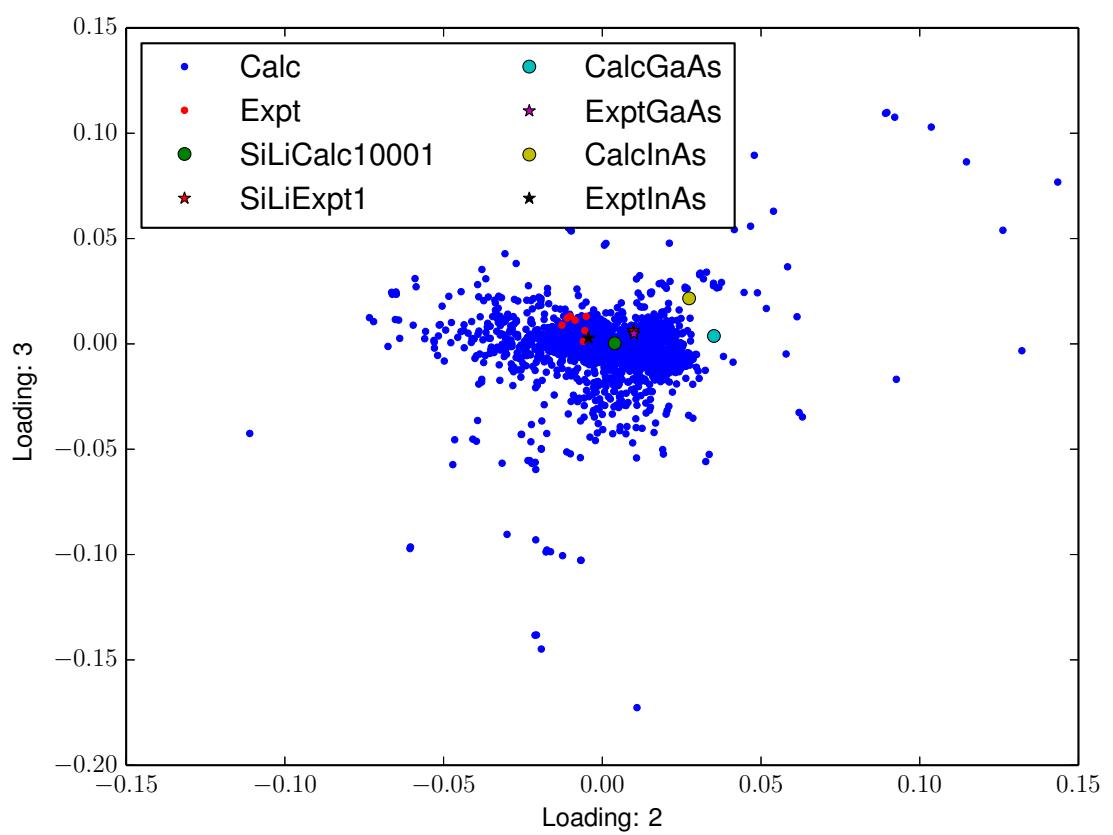


Figure 25: Loading 2 vs Loading 3

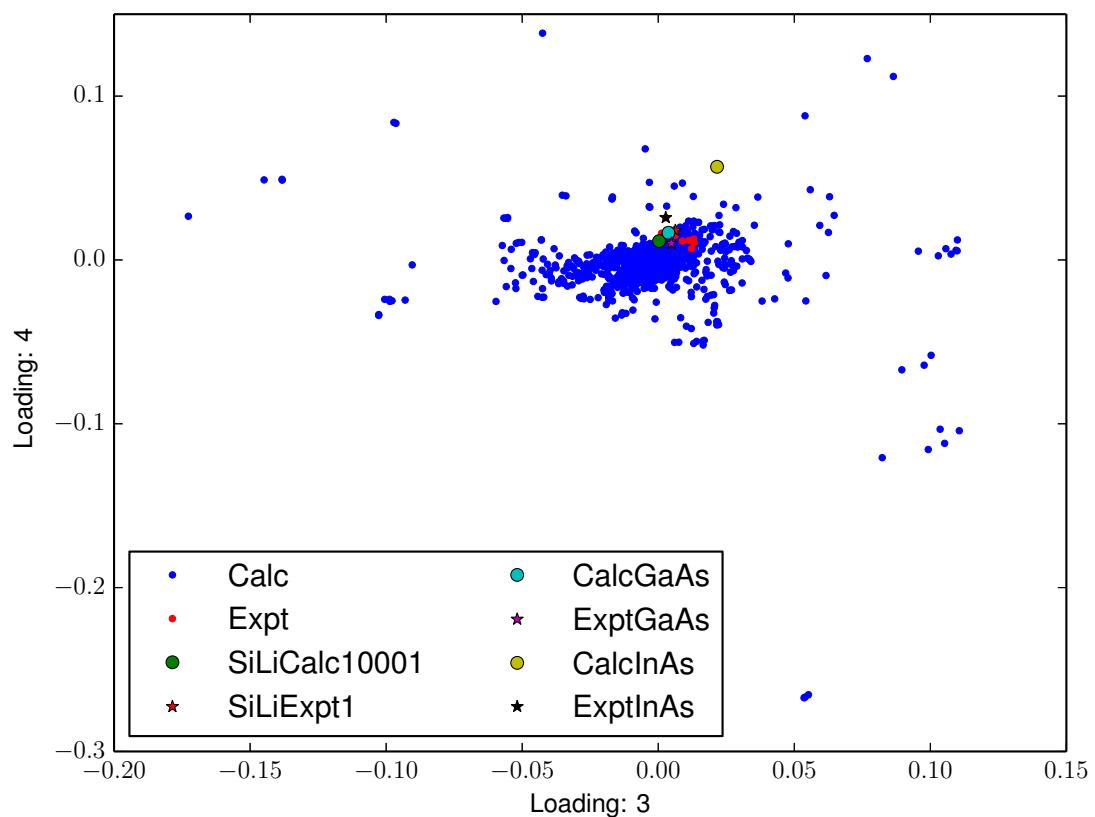


Figure 26: Loading 3 vs Loading 4

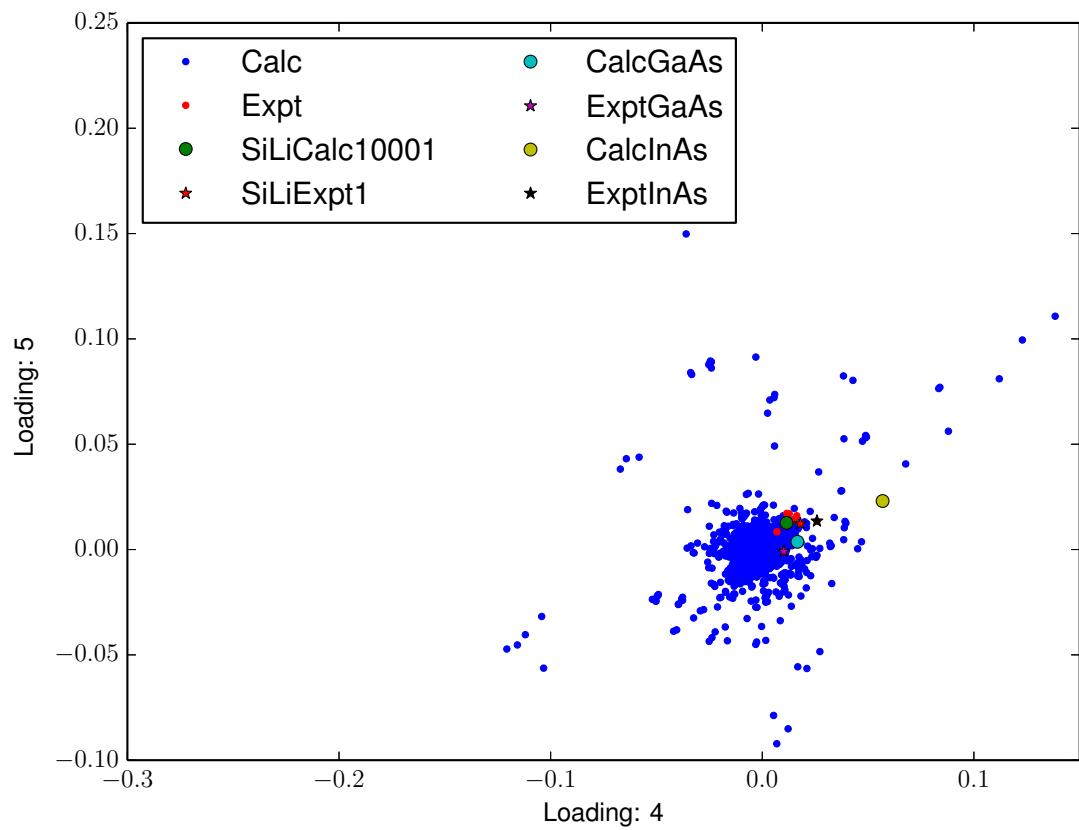


Figure 27: Loading 4 vs Loading 5

4.4.1 Eigenspace Outliers

Given that the higher order principal components did not separate the experimental images, we looked at the images with high loading values.

From the plots below, we notice that the higher order principal components capture the very high peaks in images. Indeed for some images that have very few peaks, after normalization those peaks will become much higher than those images that have many peaks.

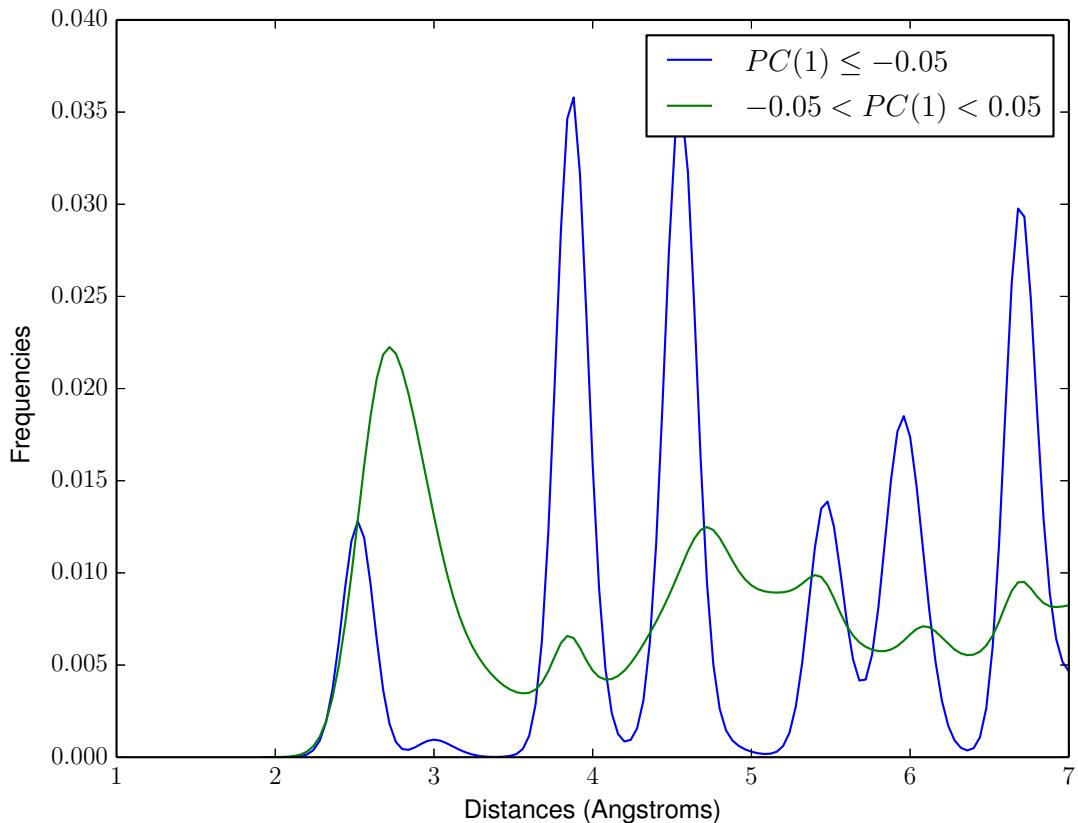


Figure 28: First Principal Component Outliers

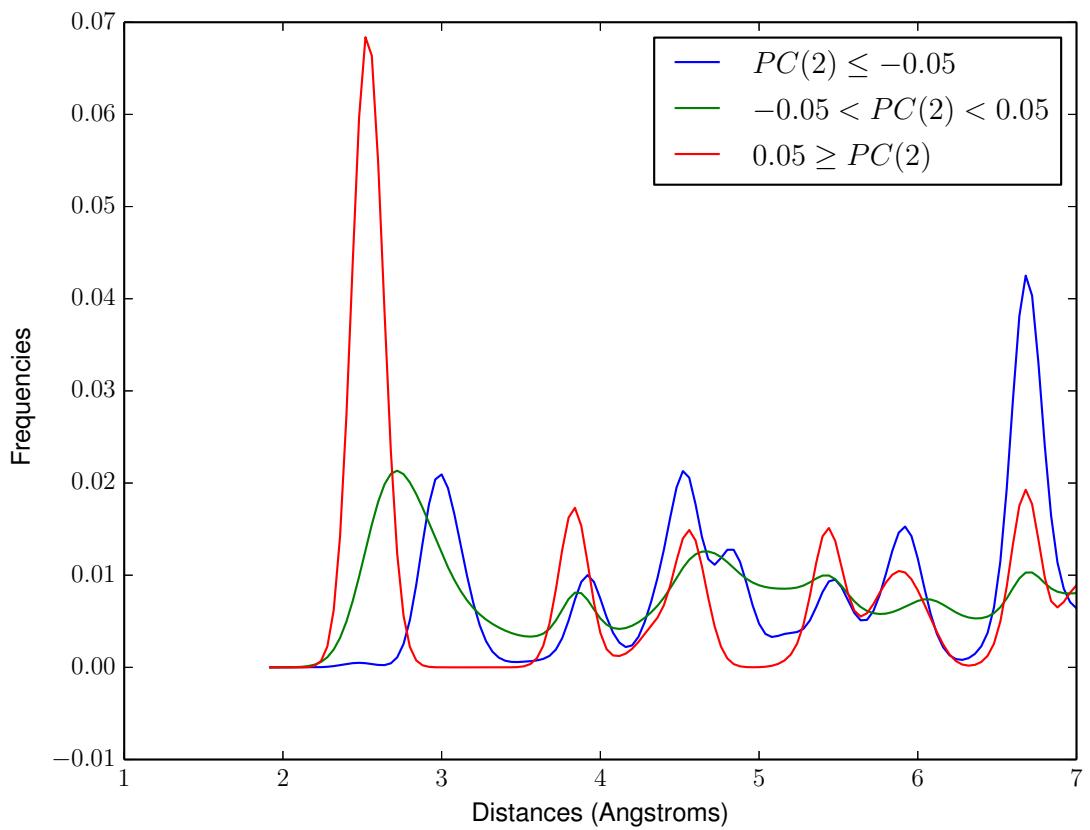


Figure 29: Second Principal Component Outliers

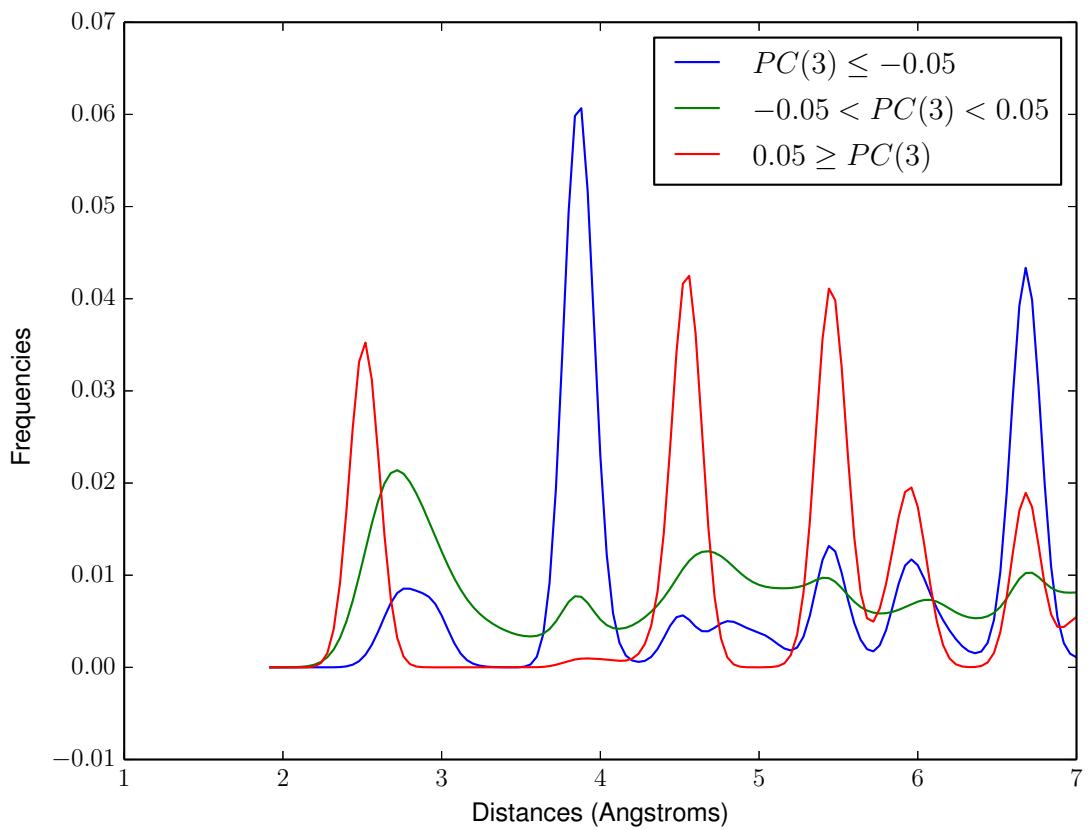


Figure 30: Third Principal Component Outliers

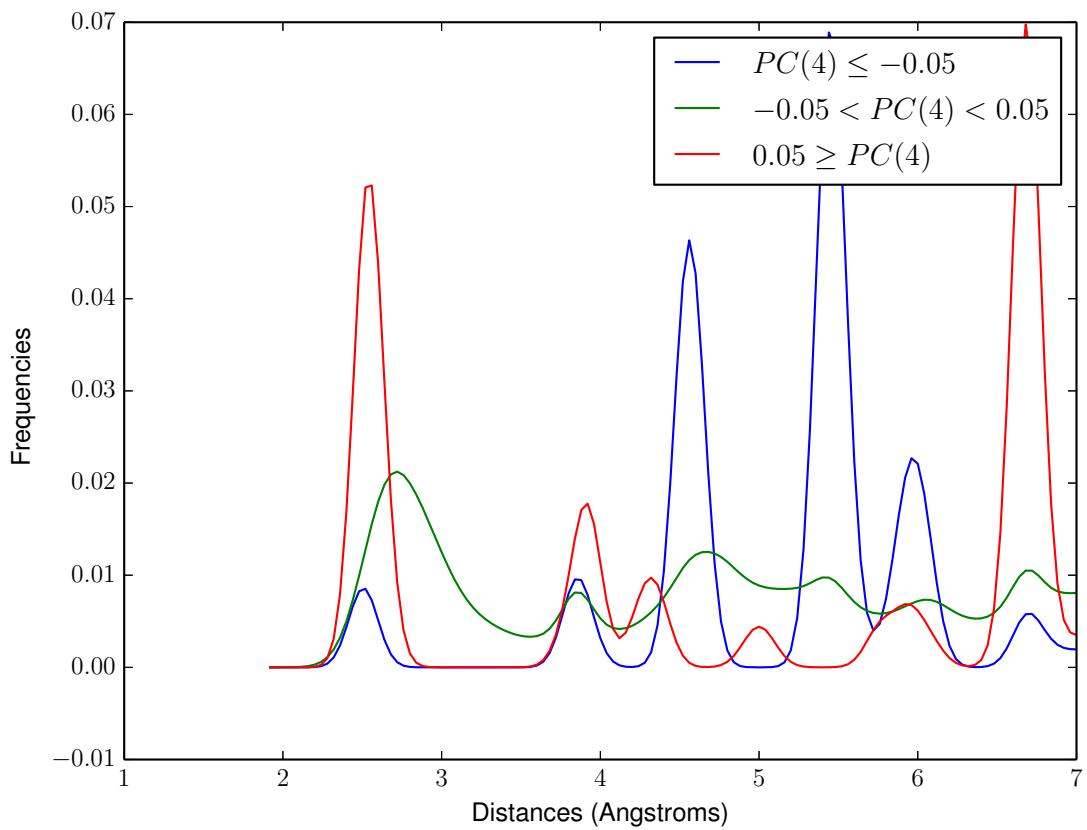


Figure 31: Fourth Principal Component Outliers

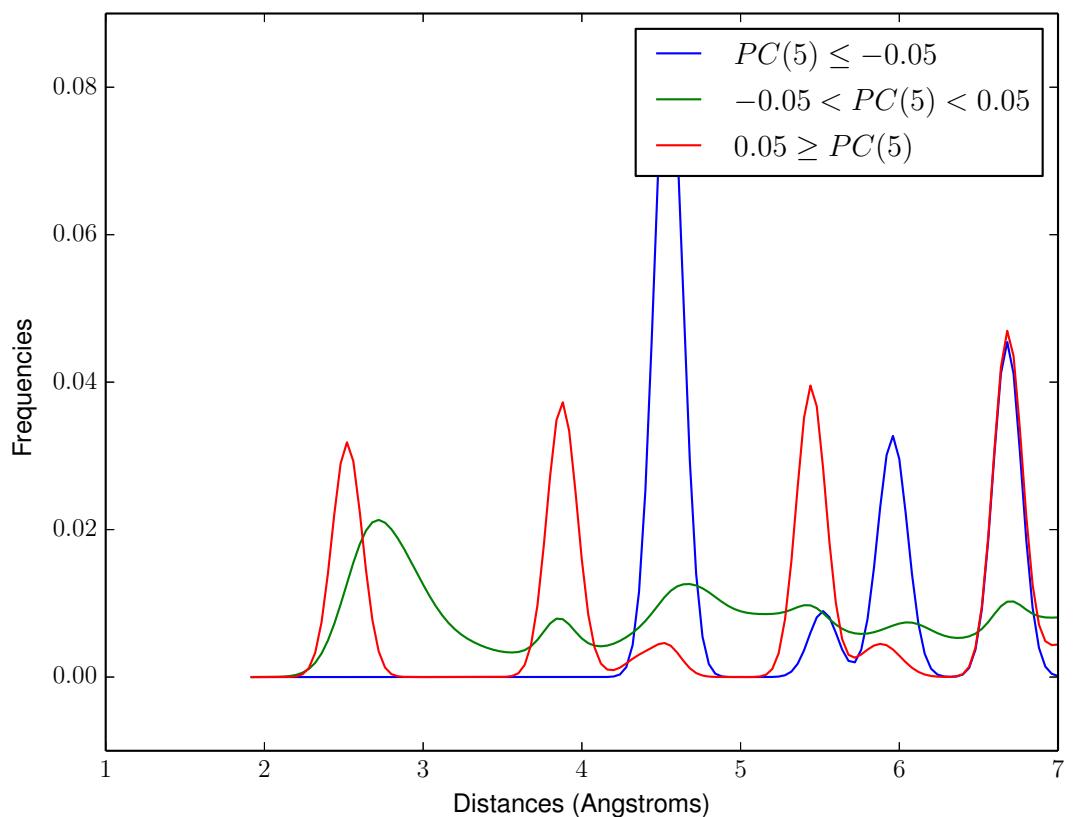


Figure 32: Fifth Principal Component Outliers

4.5 Experimental Image Recognition

Here we compute the best matches in PCA space for the experimental images for the first 3, 10, and all principal components.

4.5.1 3 Principal Components

Image	Best Match	2	3	4	5
ExptGaAs	SiLiCalc11436	SiLiCalc11634	SiLiCalc11967	SiLiCalc12738	SiLiCalc10225
ExptInAs	SiLiCalc10643	SiLiCalc10560	SiLiCalc10693	SiLiCalc10617	SiLiCalc10621
SiLiExpt1	SiLiCalc10208	SiLiCalc10315	SiLiCalc10317	SiLiCalc10188	SiLiCalc10187
SiLiExpt2	SiLiCalc10317	SiLiCalc10287	SiLiCalc10320	SiLiCalc10283	SiLiCalc10273
SiLiExpt3	SiLiCalc10287	SiLiCalc10239	SiLiCalc10259	SiLiCalc10317	SiLiCalc10232
SiLiExpt4	SiLiCalc10229	SiLiCalc10225	SiLiCalc10232	SiLiCalc10239	SiLiCalc10259
SiLiExpt5	SiLiCalc10225	SiLiCalc10256	SiLiCalc10232	SiLiCalc10229	SiLiCalc10231
SiLiExpt6	SiLiCalc10322	SiLiCalc10225	SiLiCalc10247	SiLiCalc10229	SiLiCalc10256
SiLiExpt7	SiLiCalc10225	SiLiCalc10322	SiLiCalc10256	SiLiCalc10247	SiLiCalc10229
SiLiExpt8	SiLiCalc10225	SiLiCalc10322	SiLiCalc10247	SiLiCalc10337	SiLiCalc10256

Table 3: Recognition with 3 Principal Components

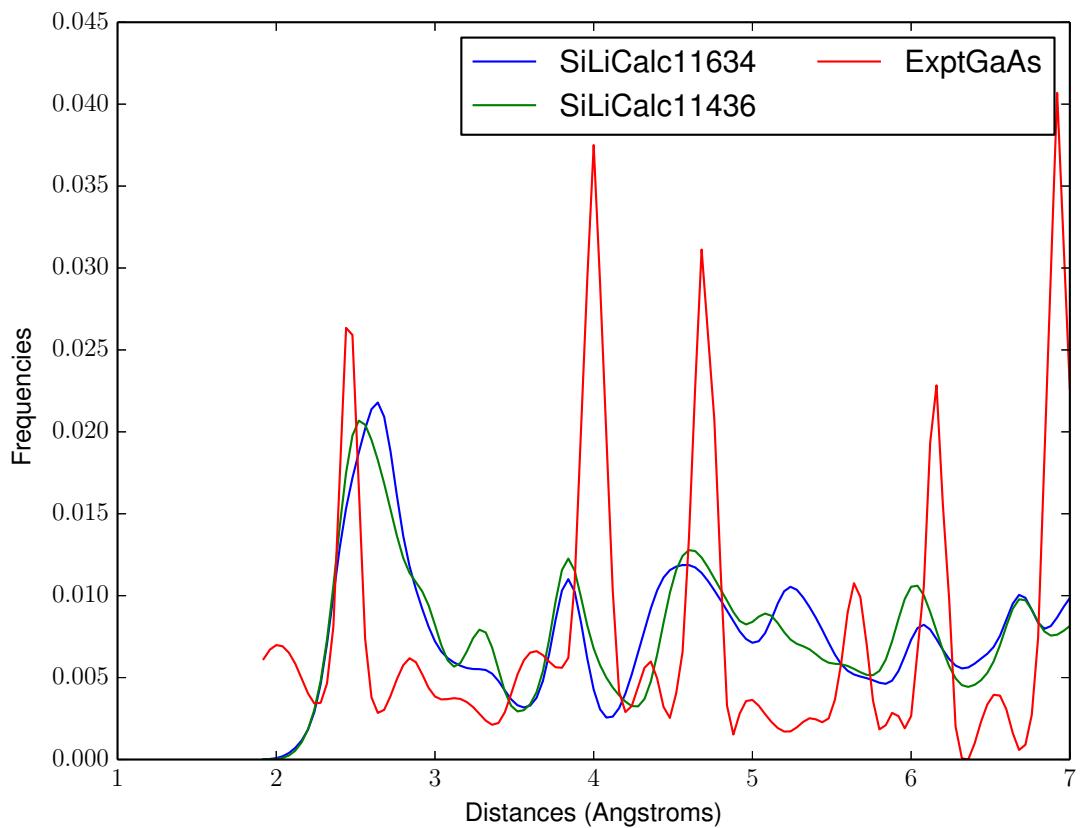


Figure 33: PCA Matches: ExptGaAs, SiLiCalc11436, SiLiCalc11634

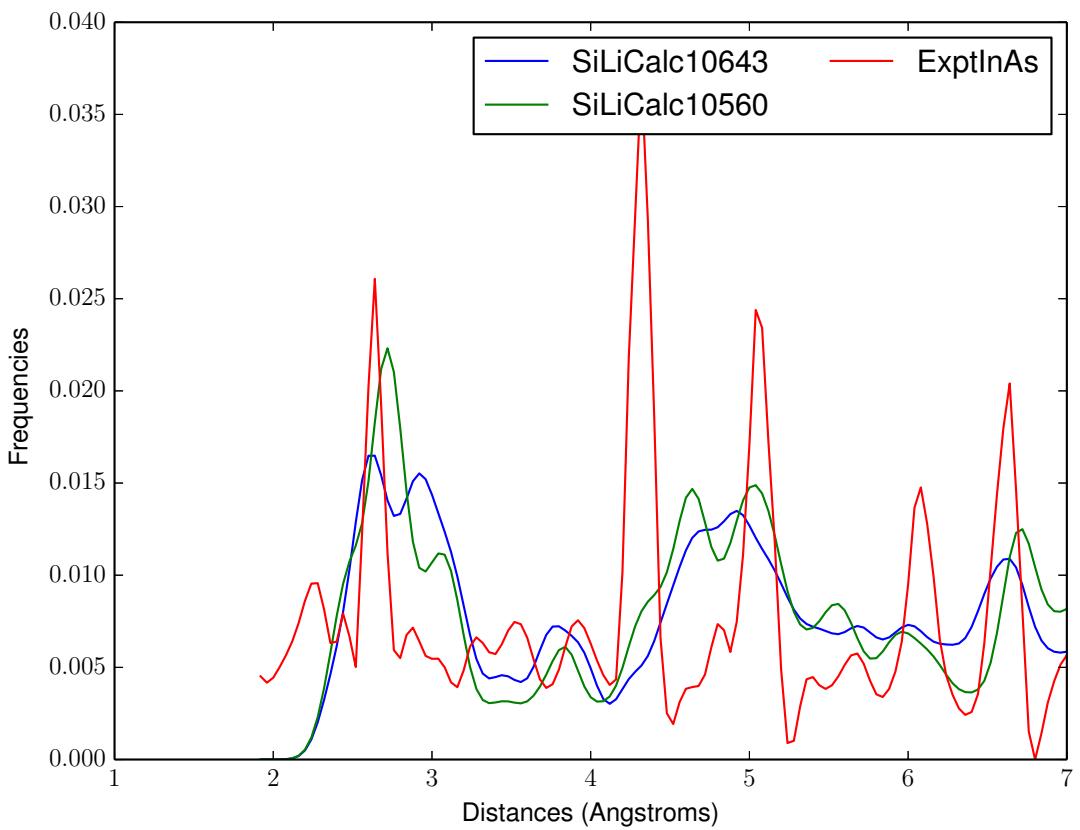


Figure 34: PCA Matches: ExptInAs, SiLiCalc10643, SiLiCalc10560

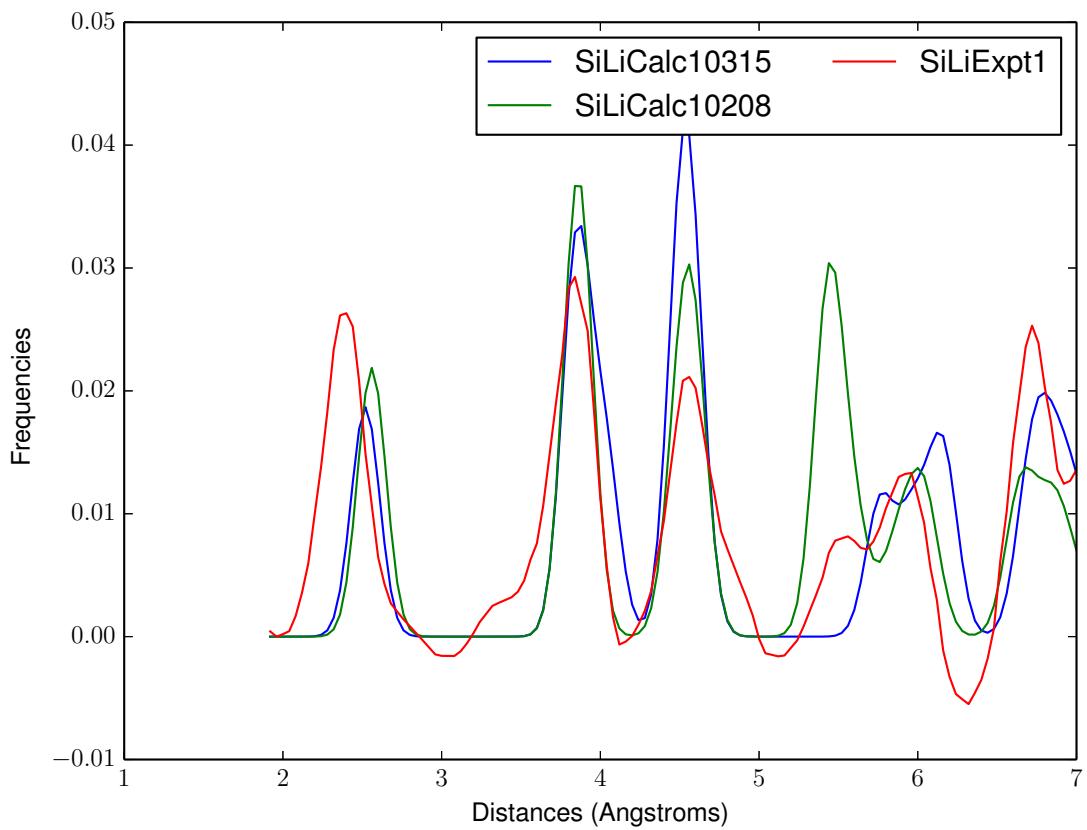


Figure 35: PCA Matches: SiLiExpt1, SiLiCalc10208, SiLiCalc10315

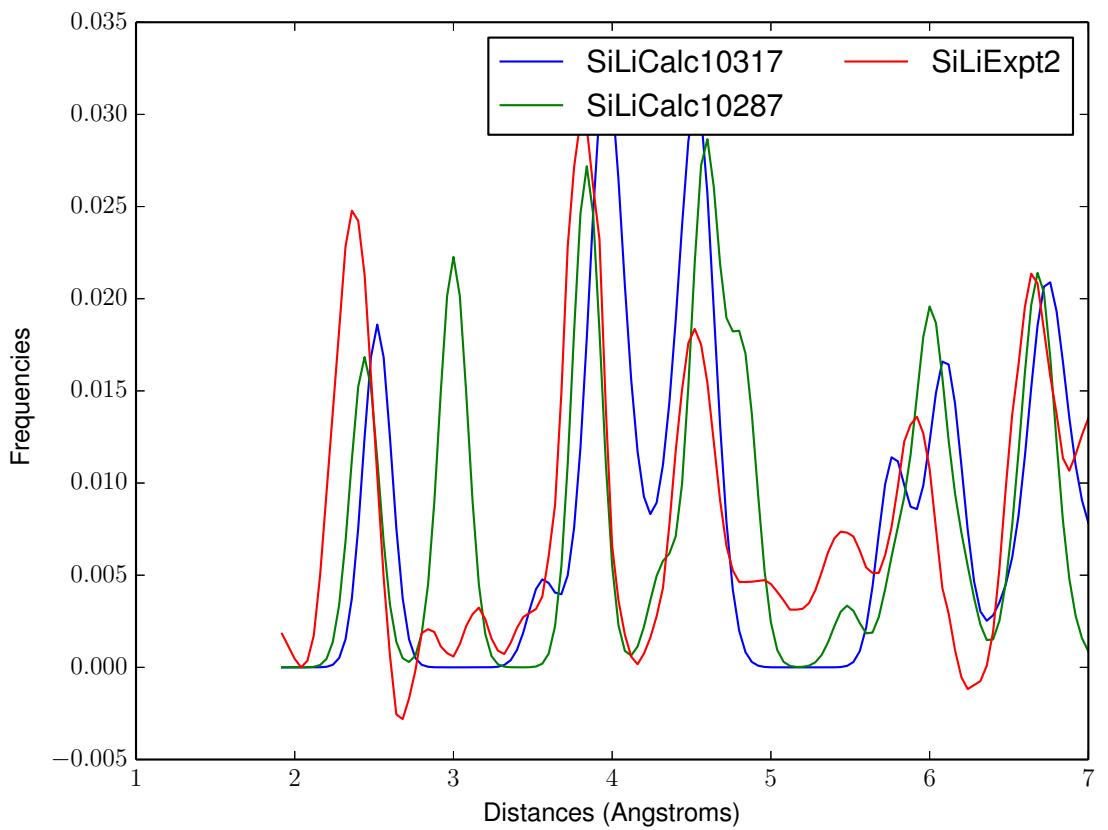


Figure 36: PCA Matches: SiLiExpt2, SiLiCalc10317, SiLiCalc10287

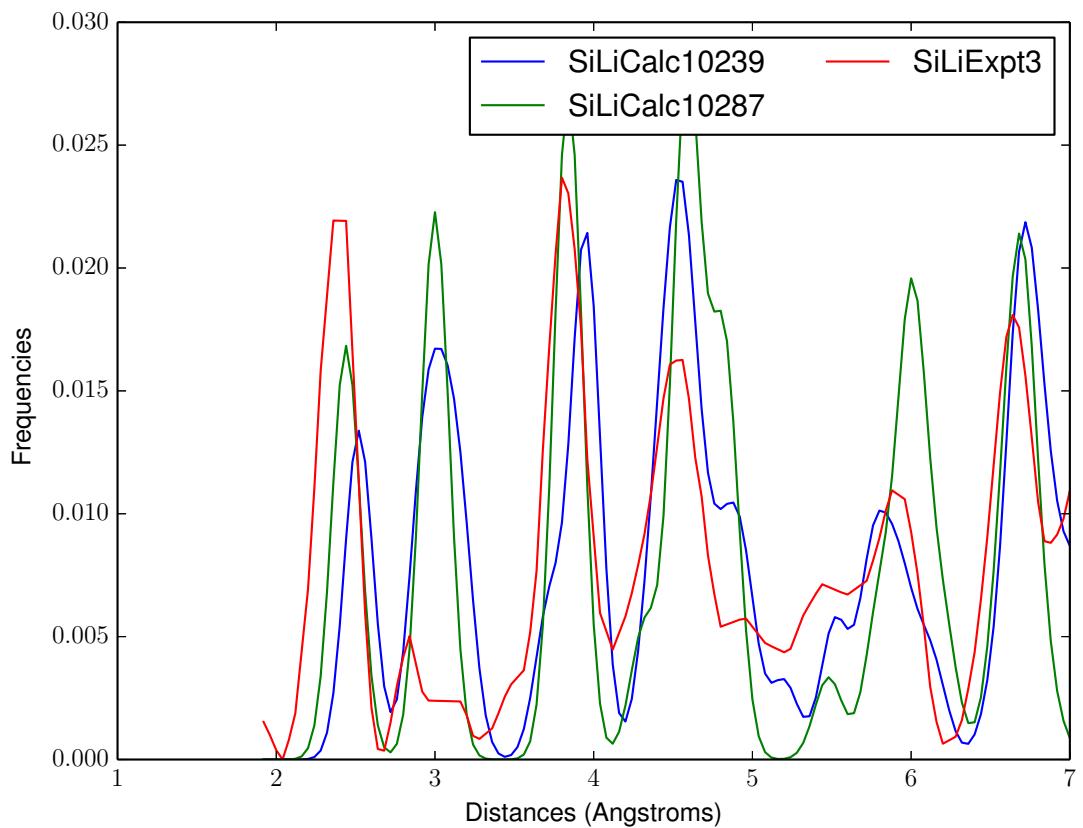


Figure 37: PCA Matches: SiLiExpt3, SiLiCalc10287, SiLiCalc10239

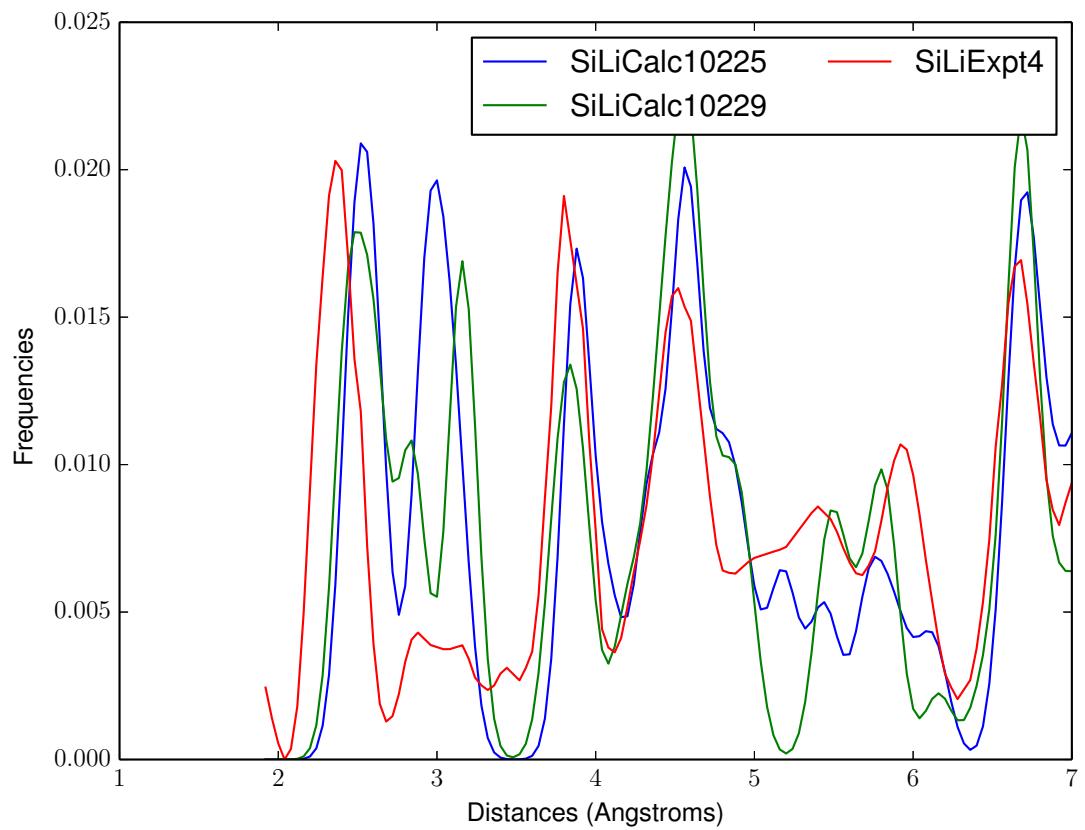


Figure 38: PCA Matches: SiLiExpt4, SiLiCalc10229, SiLiCalc10225

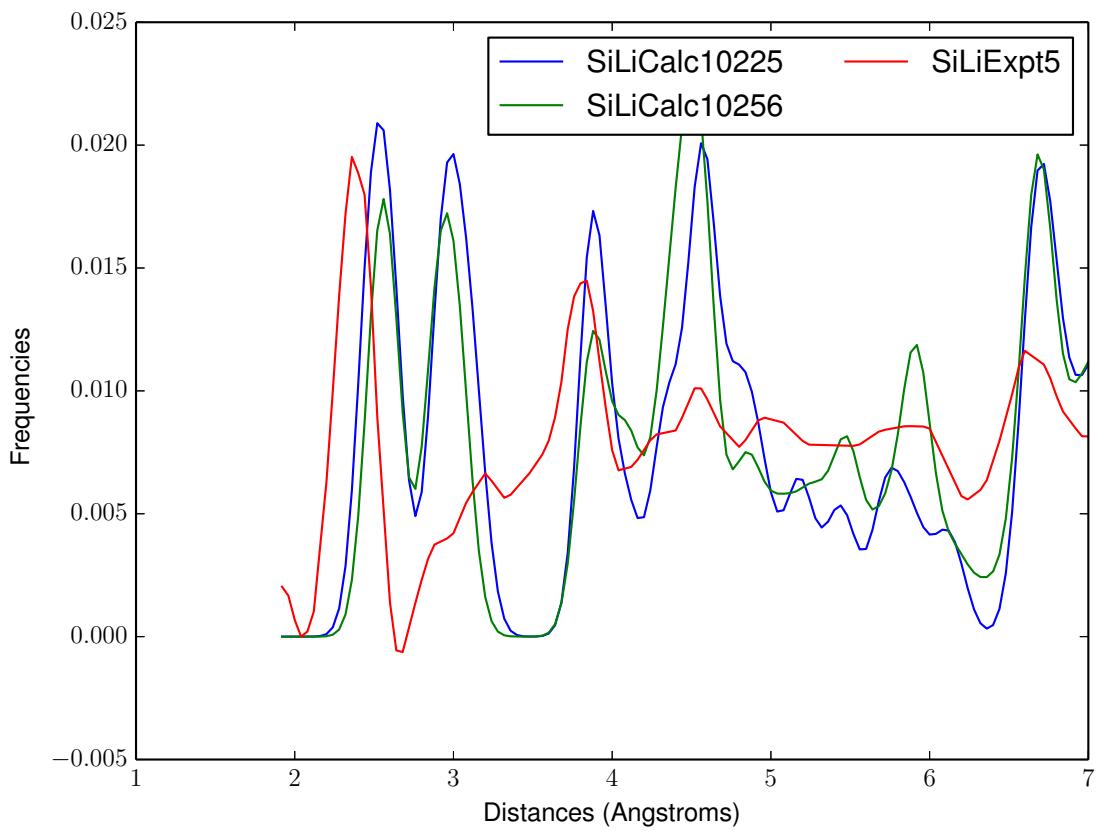


Figure 39: PCA Matches: SiLiExpt5, SiLiCalc10225, SiLiCalc10256

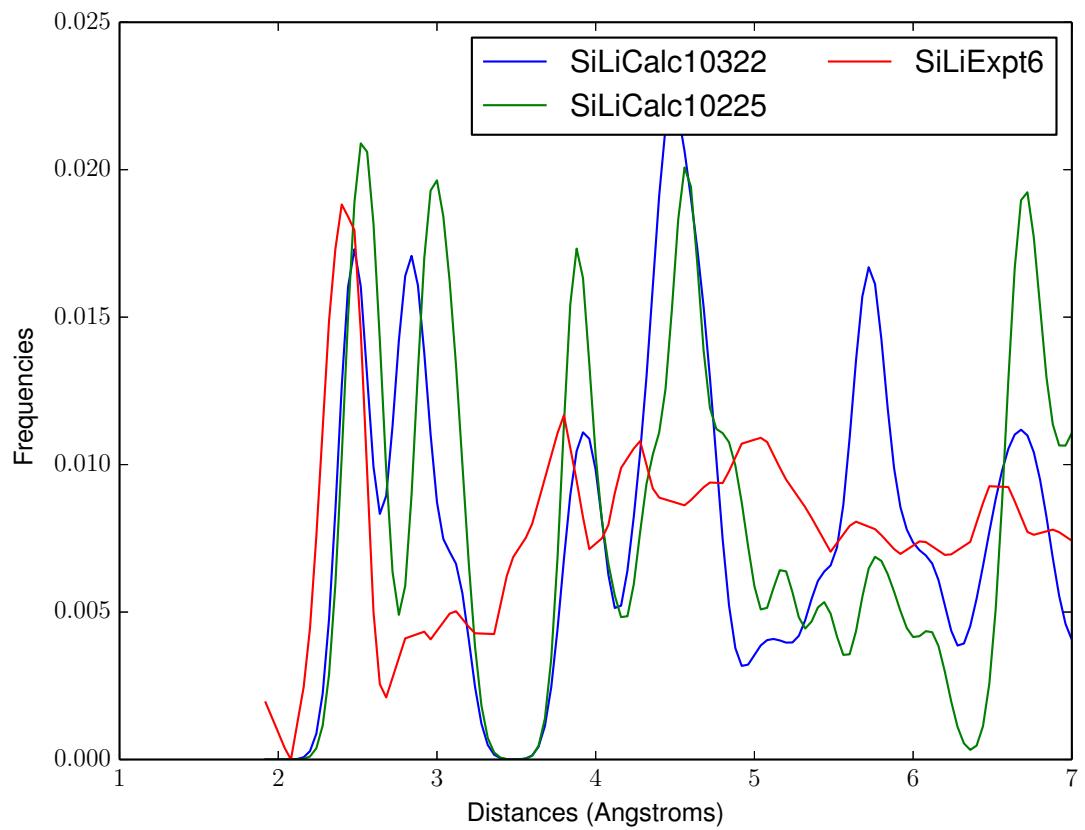


Figure 40: PCA Matches: SiLiExpt6, SiLiCalc10322, SiLiCalc10225

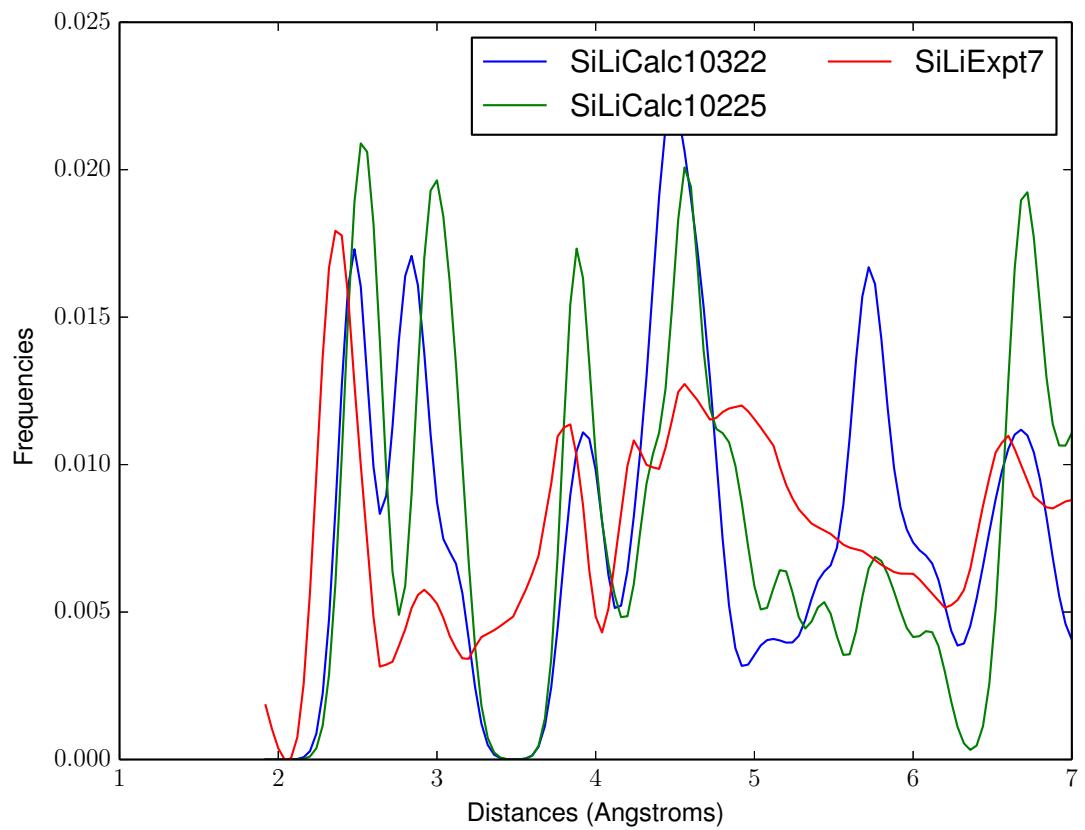


Figure 41: PCA Matches: SiLiExpt7, SiLiCalc10225, SiLiCalc10322

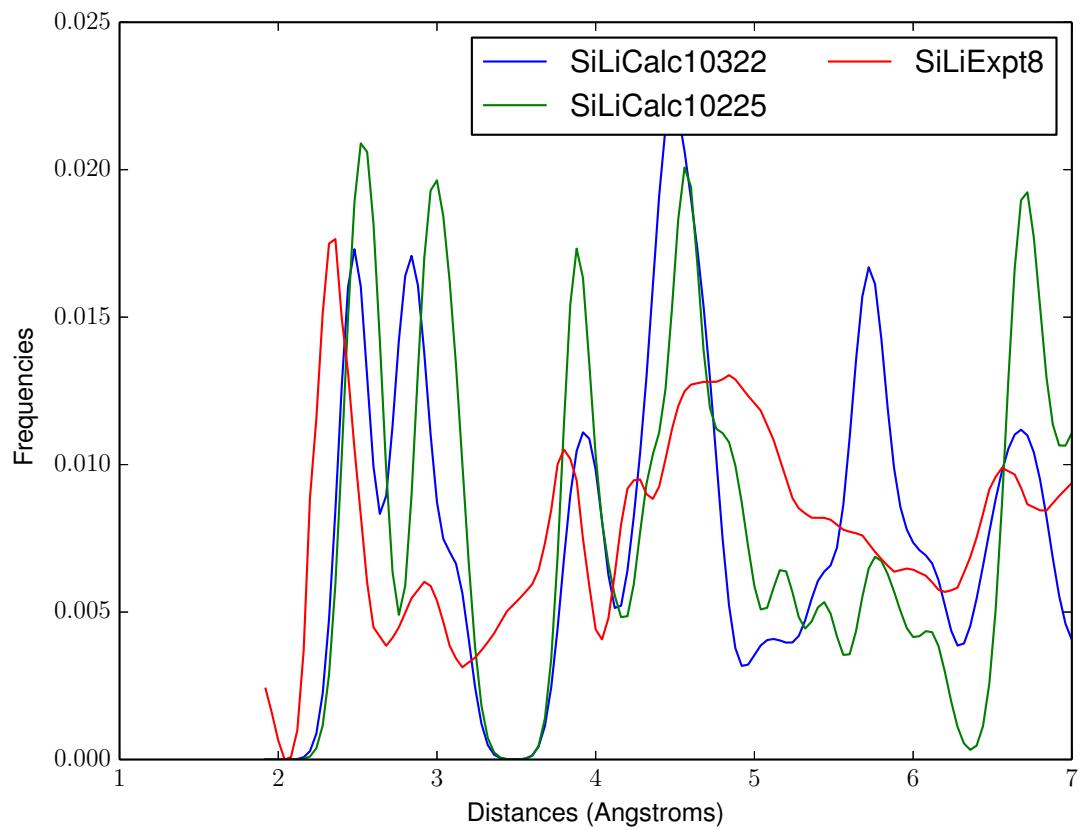


Figure 42: PCA Matches: SiLiExpt8, SiLiCalc10225, SiLiCalc10322

4.5.2 10 Principal Components

Image	Best Match	2	3	4	5
ExptGaAs	CalcGaAs	SiLiCalc10329	SiLiCalc11337	SiLiCalc11436	SiLiCalc10571
ExptInAs	SiLiCalc10646	SiLiCalc10805	SiLiCalc10792	SiLiCalc10836	SiLiCalc10767
SiLiExpt1	SiLiCalc10213	SiLiCalc10215	SiLiCalc10001	SiLiCalc10003	SiLiCalc10313
SiLiExpt2	SiLiCalc10001	SiLiCalc10003	SiLiCalc10209	SiLiCalc10317	SiLiCalc10313
SiLiExpt3	SiLiCalc10257	SiLiCalc10317	SiLiCalc10259	SiLiCalc10258	SiLiCalc10256
SiLiExpt4	SiLiCalc10257	SiLiCalc10258	SiLiCalc10256	SiLiCalc10229	SiLiCalc10232
SiLiExpt5	SiLiCalc10445	SiLiCalc10616	SiLiCalc11436	SiLiCalc10329	SiLiCalc11337
SiLiExpt6	SiLiCalc10445	SiLiCalc10616	SiLiCalc11436	SiLiCalc10693	SiLiCalc11337
SiLiExpt7	SiLiCalc10445	SiLiCalc10693	SiLiCalc11337	SiLiCalc10616	SiLiCalc10482
SiLiExpt8	SiLiCalc10445	SiLiCalc10693	SiLiCalc10329	SiLiCalc11337	SiLiCalc10482

Table 4: Recognition with 10 Principal Components

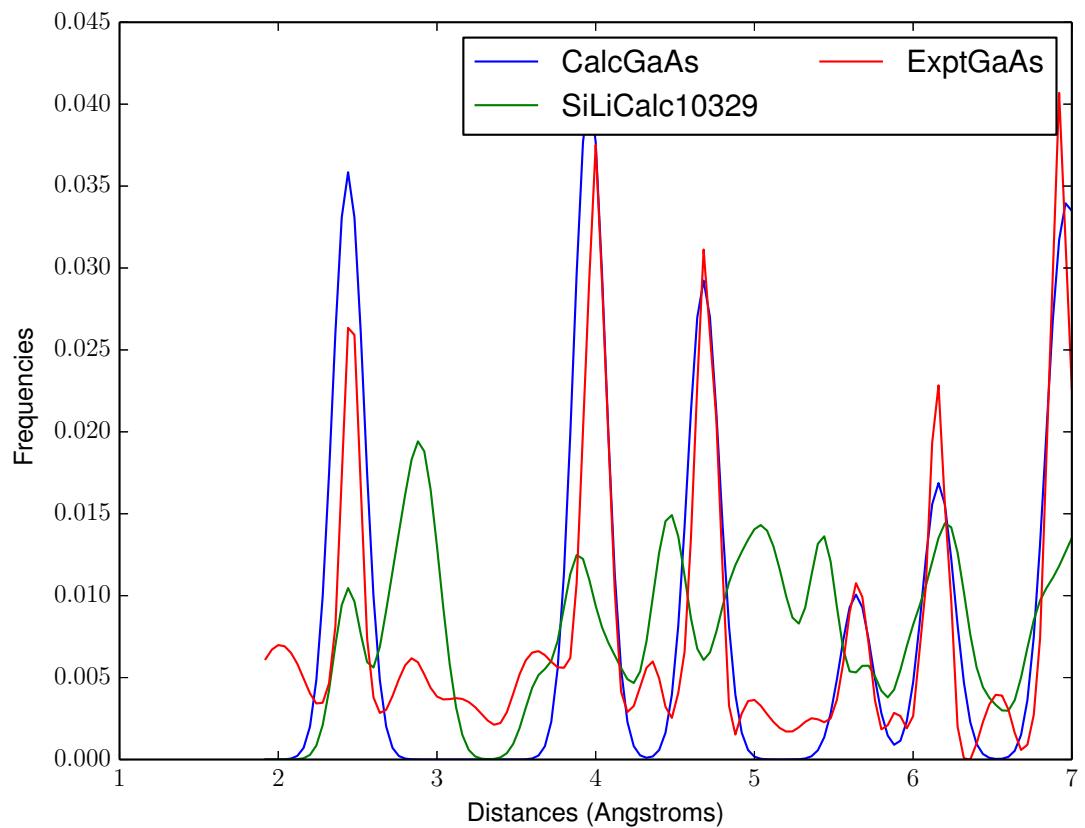


Figure 43: PCA Matches: ExptGaAs, CalcGaAs, SiLiCalc10329

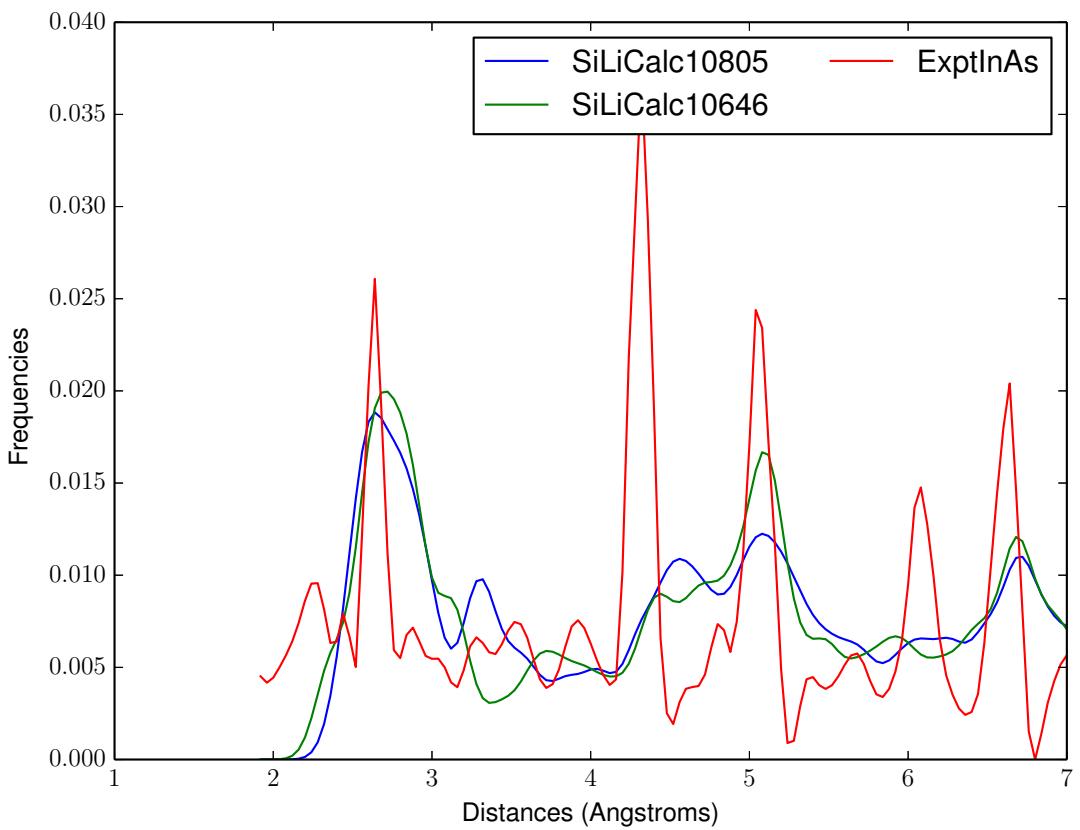


Figure 44: PCA Matches: ExptInAs, SiLiCalc10646, SiLiCalc10805

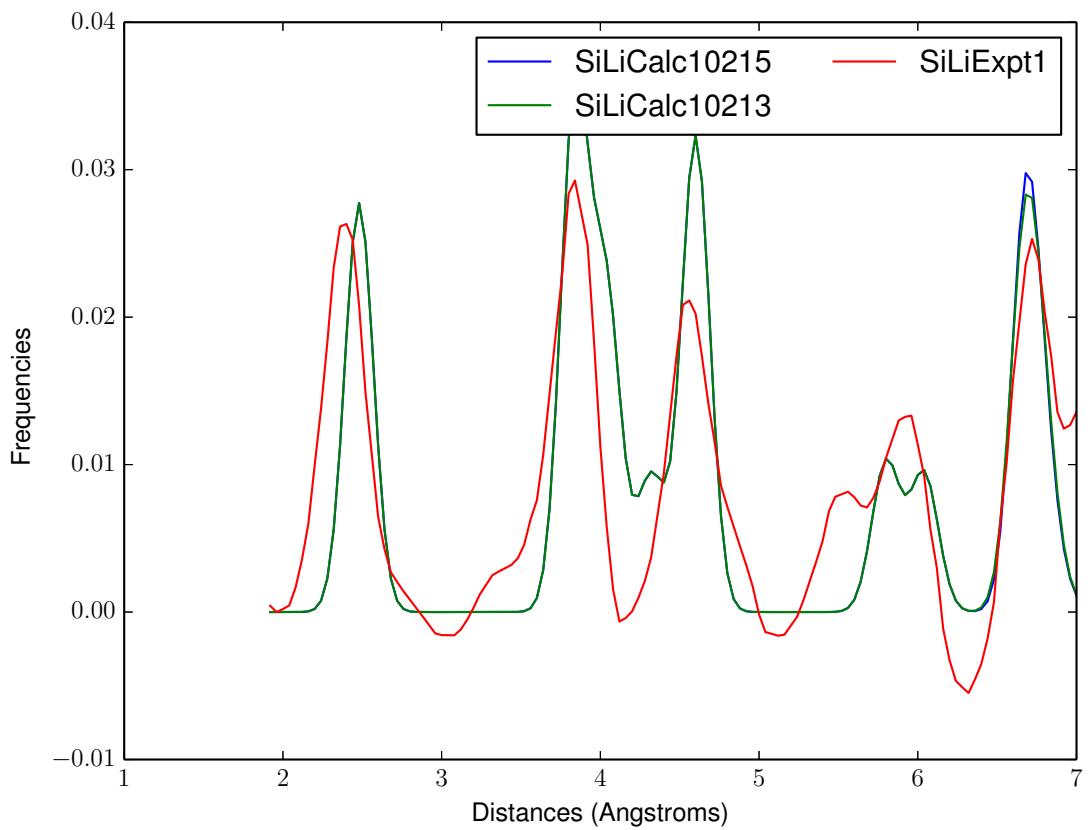


Figure 45: PCA Matches: SiLiExpt1, SiLiCalc10213, SiLiCalc10215

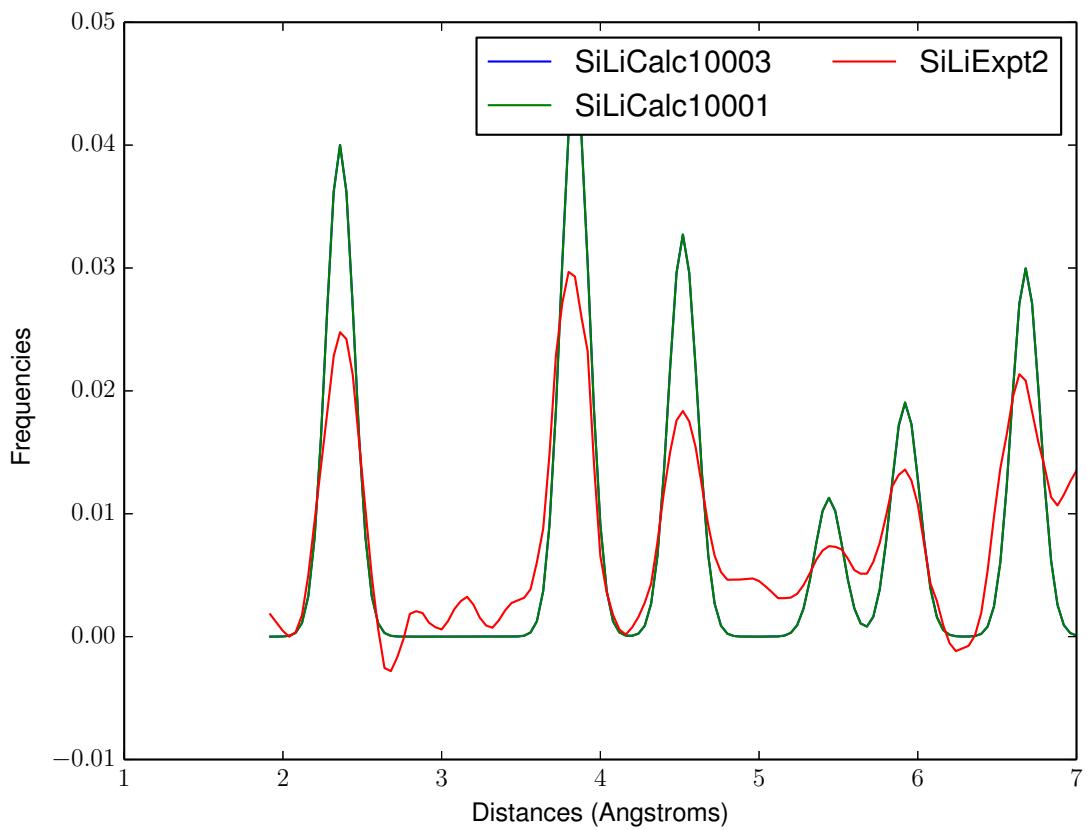


Figure 46: PCA Matches: SiLiExpt2, SiLiCalc10001, SiLiCalc10003

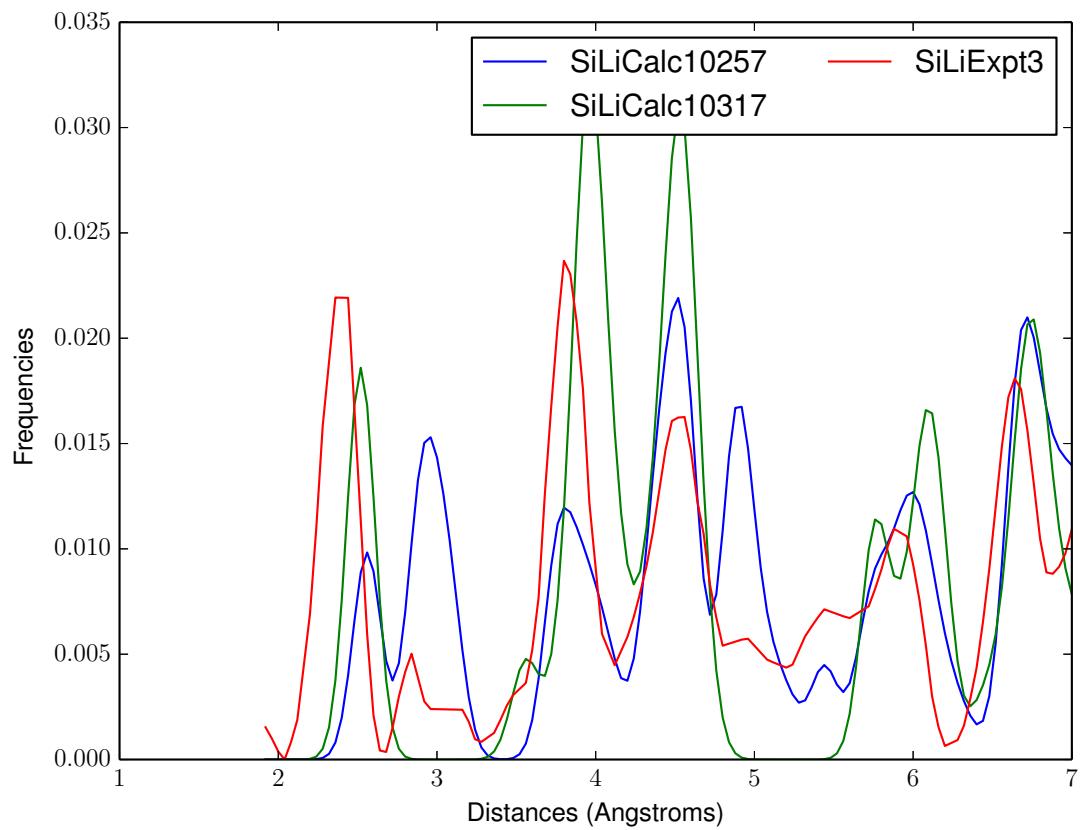


Figure 47: PCA Matches: SiLiExpt3, SiLiCalc10257, SiLiCalc10317

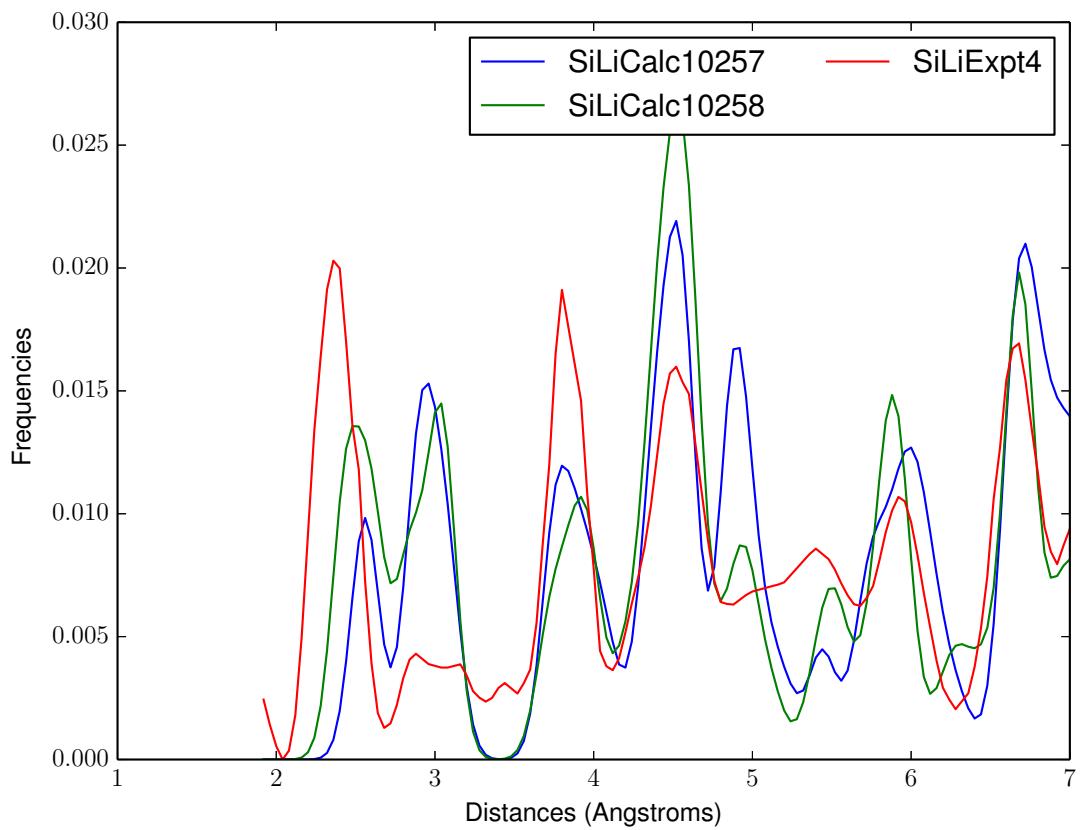


Figure 48: PCA Matches: SiLiExpt4, SiLiCalc10257, SiLiCalc10258

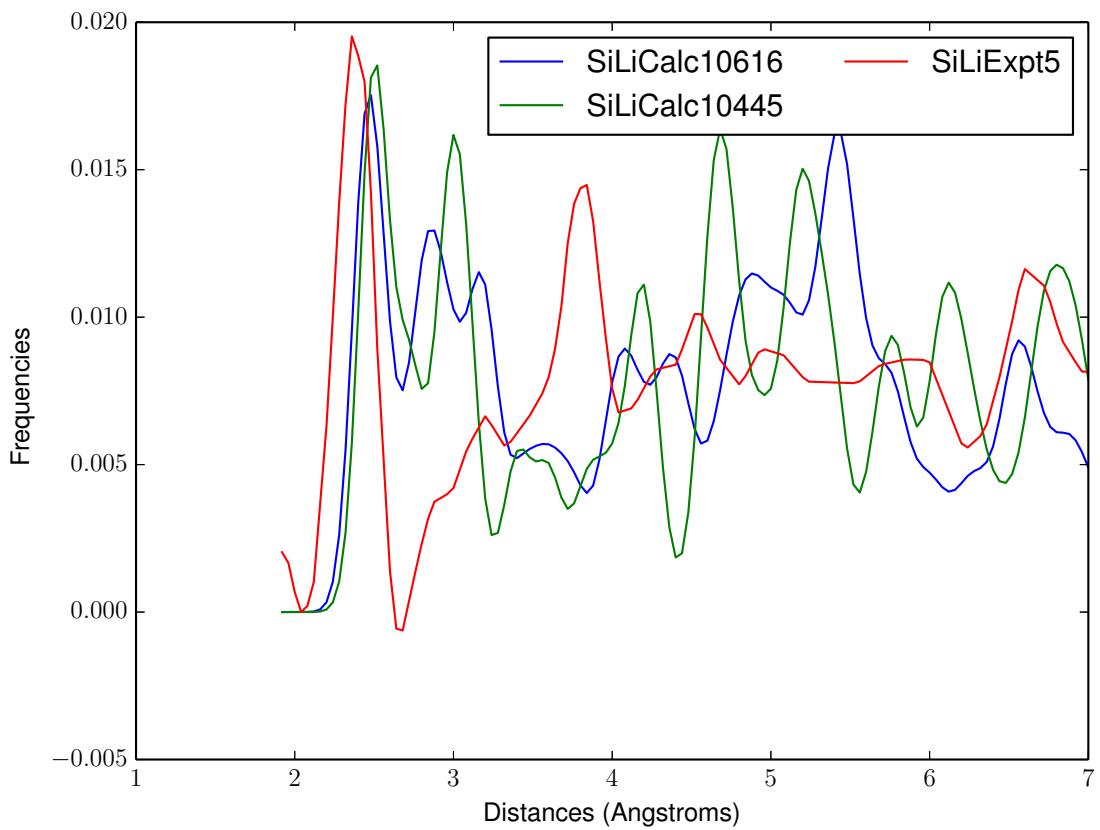


Figure 49: PCA Matches: SiLiExpt5, SiLiCalc10445, SiLiCalc10616

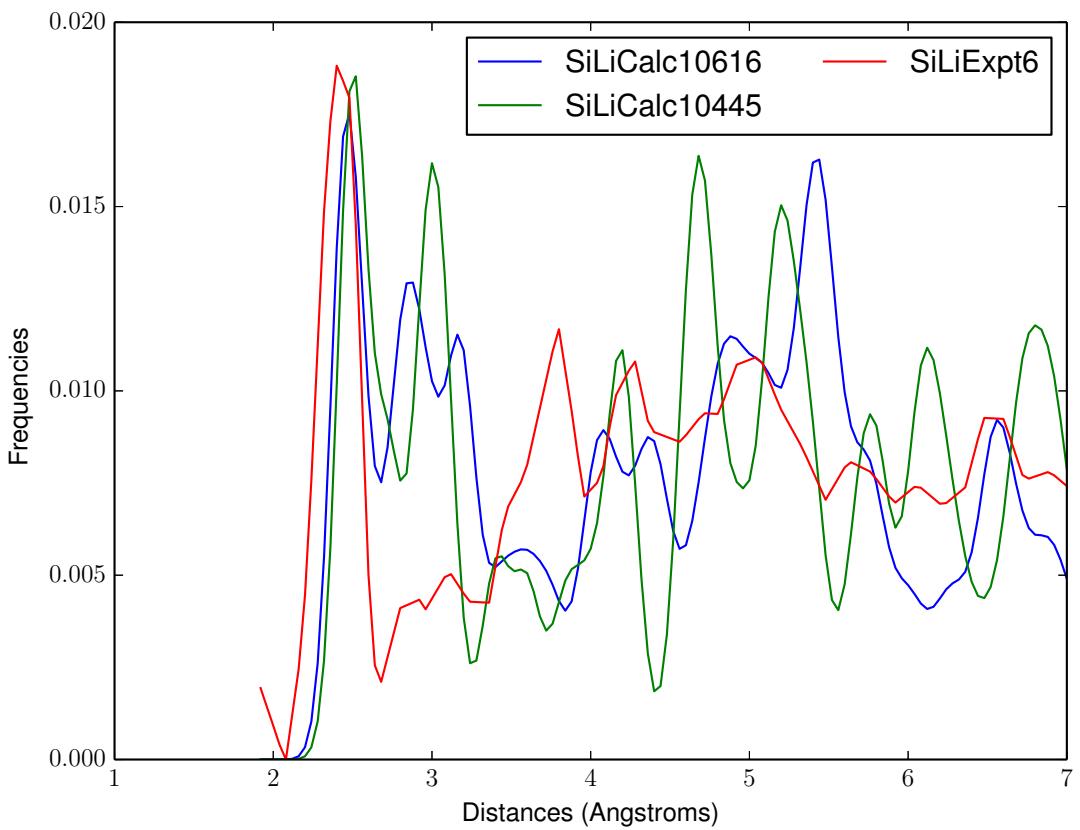


Figure 50: PCA Matches: SiLiExpt6, SiLiCalc10445, SiLiCalc10616

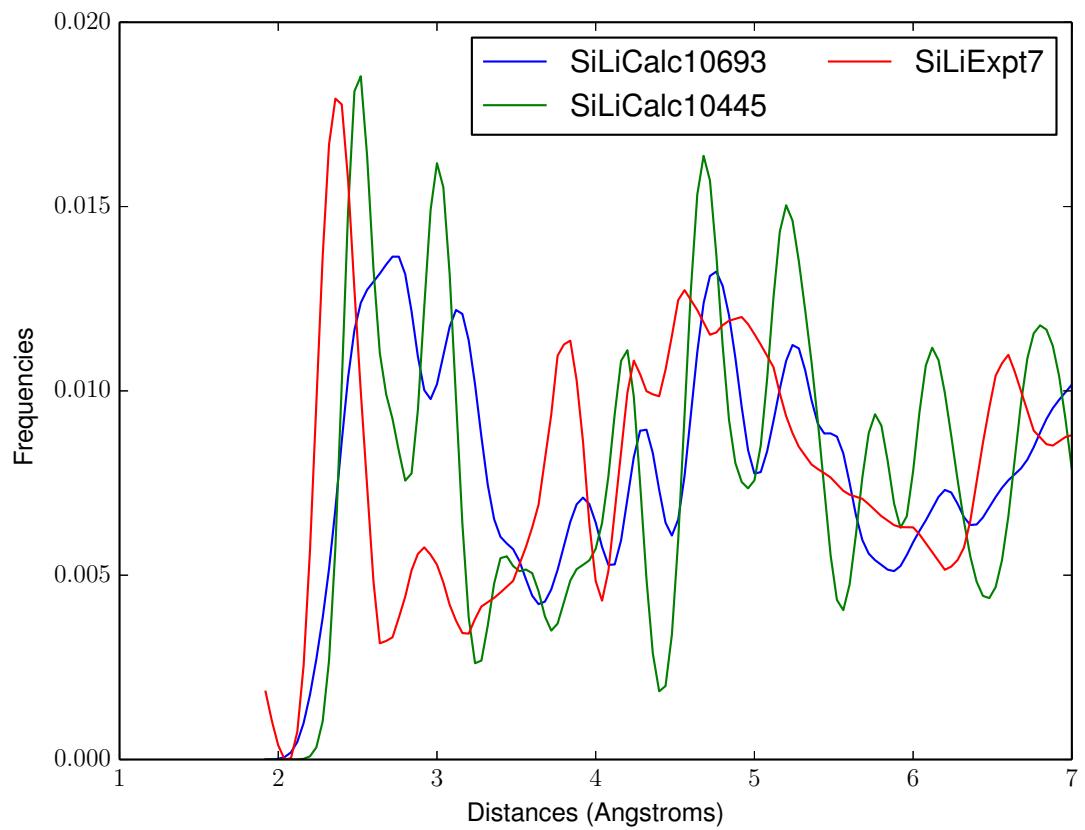


Figure 51: PCA Matches: SiLiExpt7, SiLiCalc10445, SiLiCalc10693

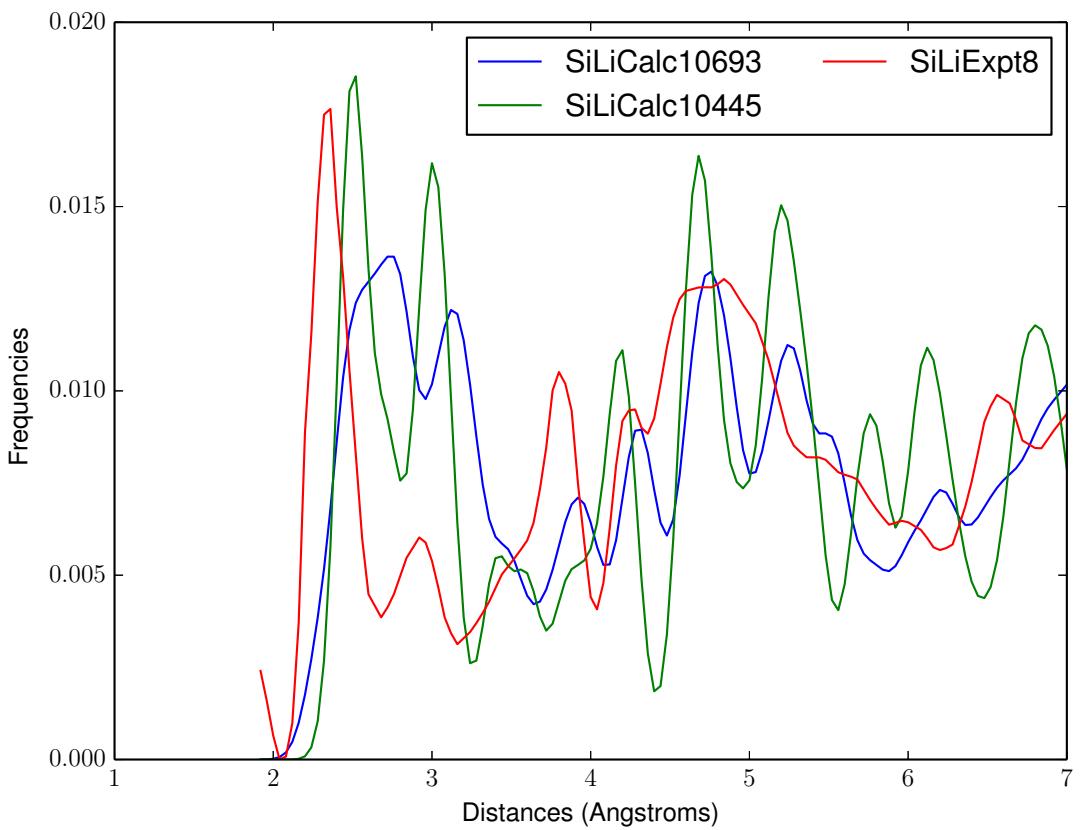


Figure 52: PCA Matches: SiLiExpt8, SiLiCalc10445, SiLiCalc10693

4.5.3 128 Principal Components

Image	Best Match	2	3	4	5
ExptGaAs	CalcGaAs	SiLiCalc10445	SiLiCalc11436	SiLiCalc10693	SiLiCalc11337
ExptInAs	SiLiCalc10429	SiLiCalc10602	SiLiCalc10838	SiLiCalc10901	SiLiCalc10607
SiLiExpt1	SiLiCalc10194	SiLiCalc10001	SiLiCalc10003	SiLiCalc10136	SiLiCalc10147
SiLiExpt2	SiLiCalc10001	SiLiCalc10003	SiLiCalc10194	SiLiCalc10136	SiLiCalc10147
SiLiExpt3	SiLiCalc10258	SiLiCalc10229	SiLiCalc10245	SiLiCalc11436	SiLiCalc10259
SiLiExpt4	SiLiCalc10258	SiLiCalc11436	SiLiCalc10229	SiLiCalc11337	SiLiCalc11634
SiLiExpt5	SiLiCalc10616	SiLiCalc11337	SiLiCalc10693	SiLiCalc11436	SiLiCalc11336
SiLiExpt6	SiLiCalc10616	SiLiCalc10693	SiLiCalc11337	SiLiCalc11436	SiLiCalc11336
SiLiExpt7	SiLiCalc10693	SiLiCalc11337	SiLiCalc10482	SiLiCalc10616	SiLiCalc10651
SiLiExpt8	SiLiCalc10693	SiLiCalc10651	SiLiCalc11337	SiLiCalc10482	SiLiCalc10616

Table 5: Recognition with 128 Principal Components

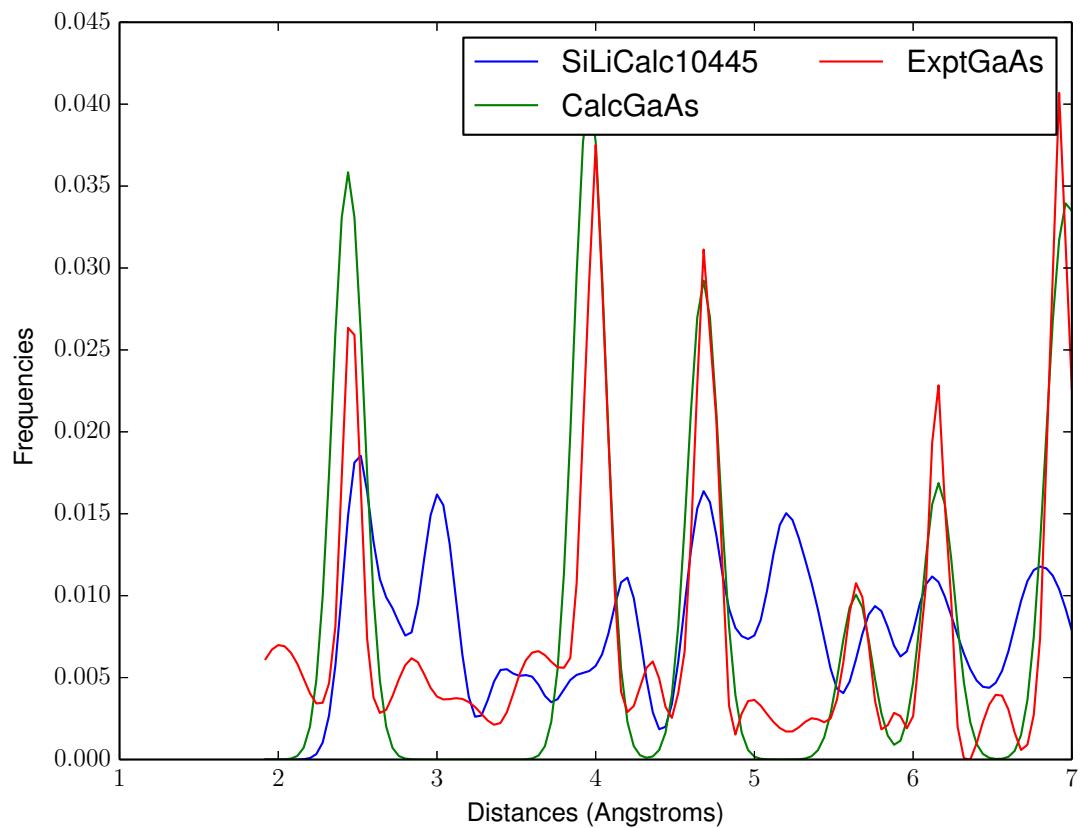


Figure 53: PCA Matches: ExptGaAs, CalcGaAs, SiLiCalc10445

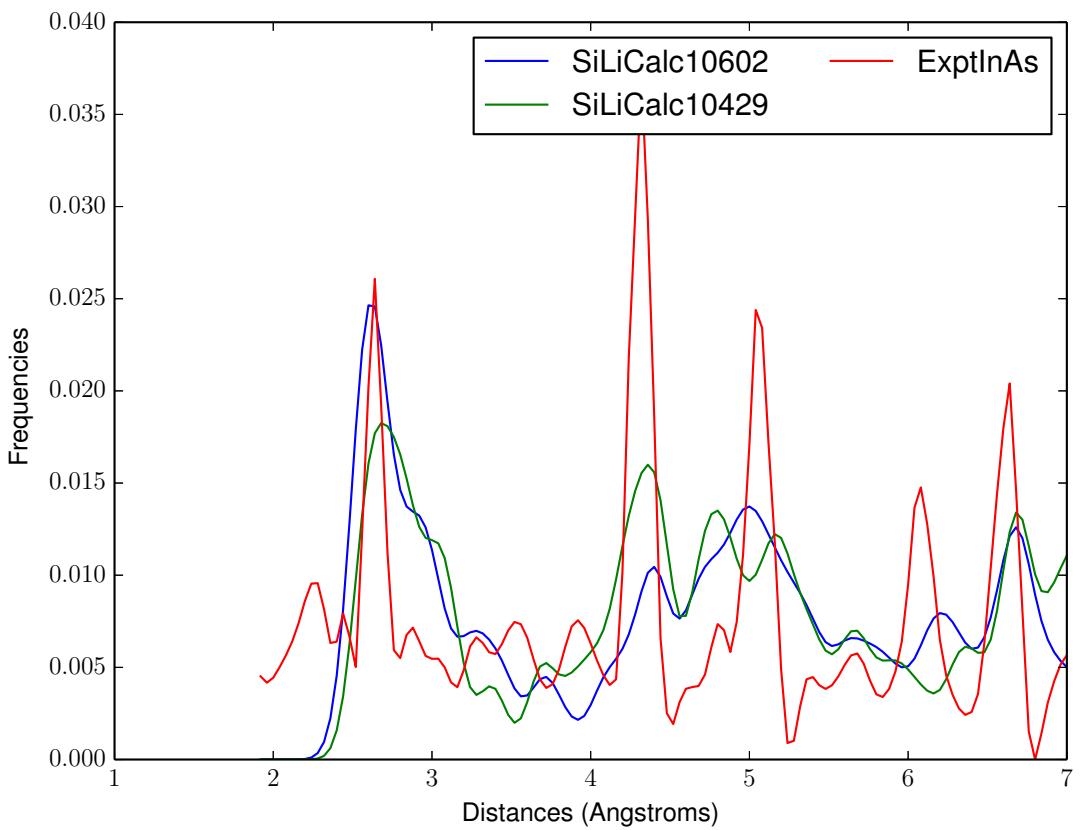


Figure 54: PCA Matches: ExptInAs, SiLiCalc10429, SiLiCalc10602

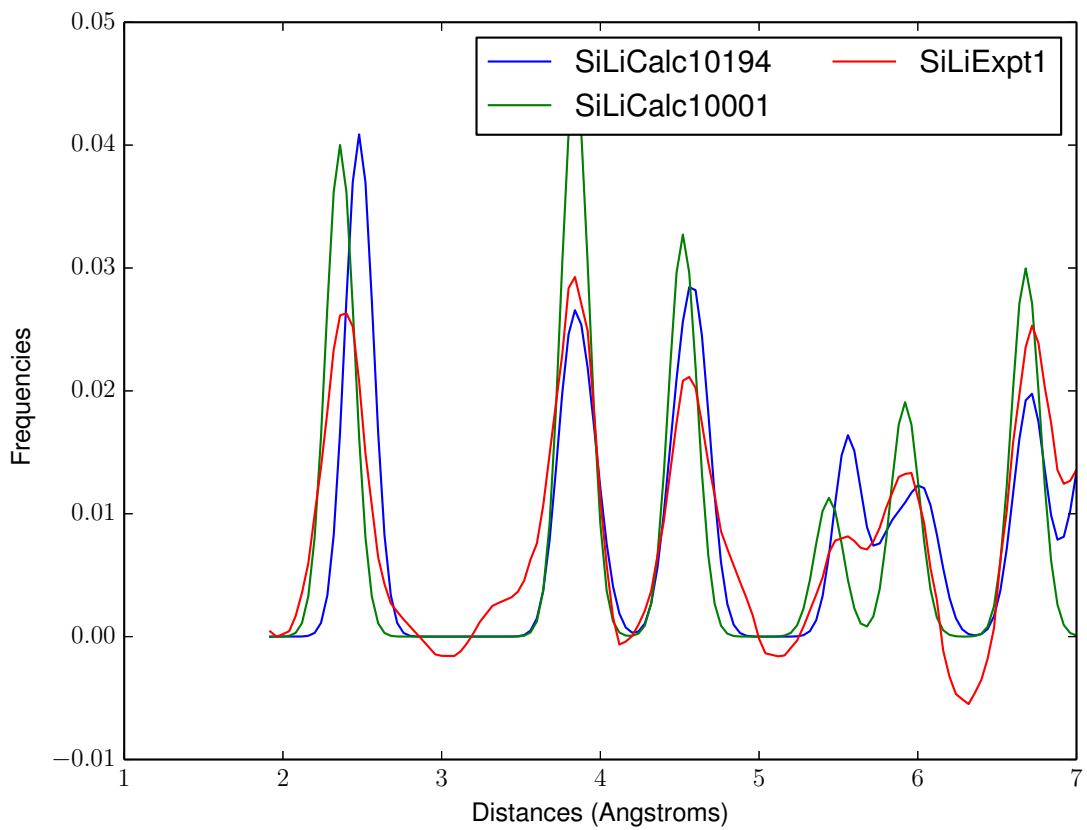


Figure 55: PCA Matches: SiLiExpt1, SiLiCalc10194, SiLiCalc10001

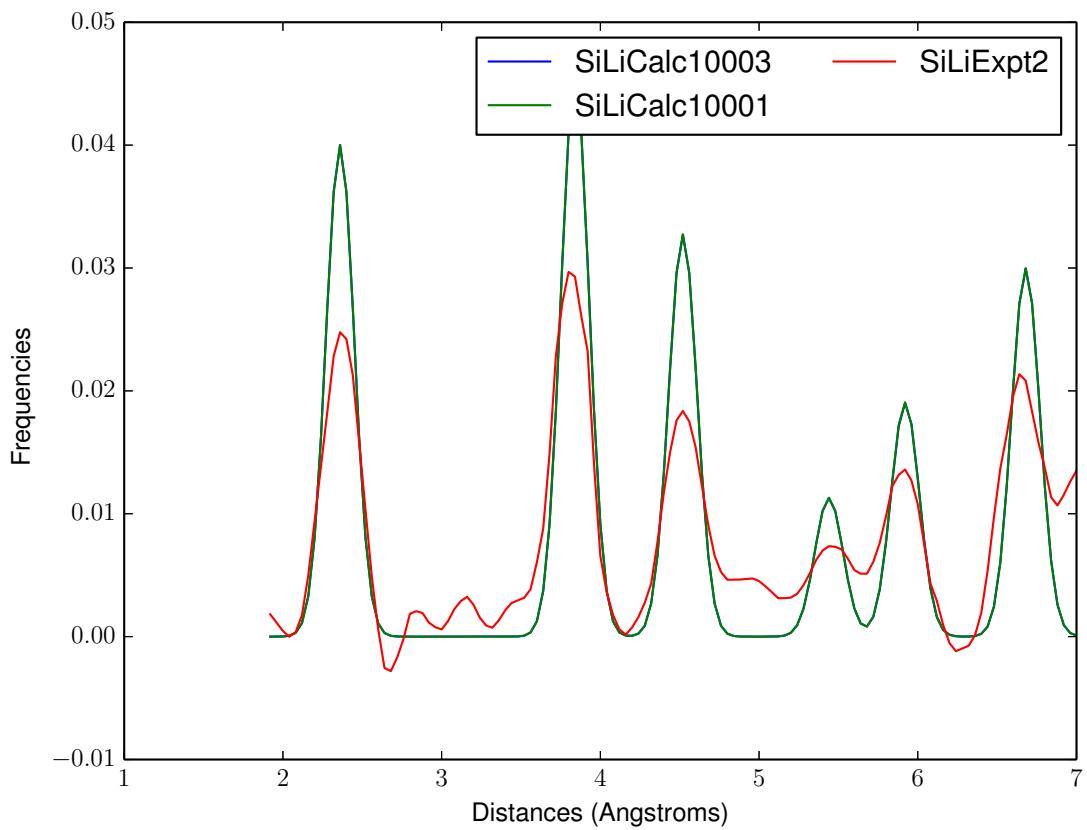


Figure 56: PCA Matches: SiLiExpt2, SiLiCalc10001, SiLiCalc10003

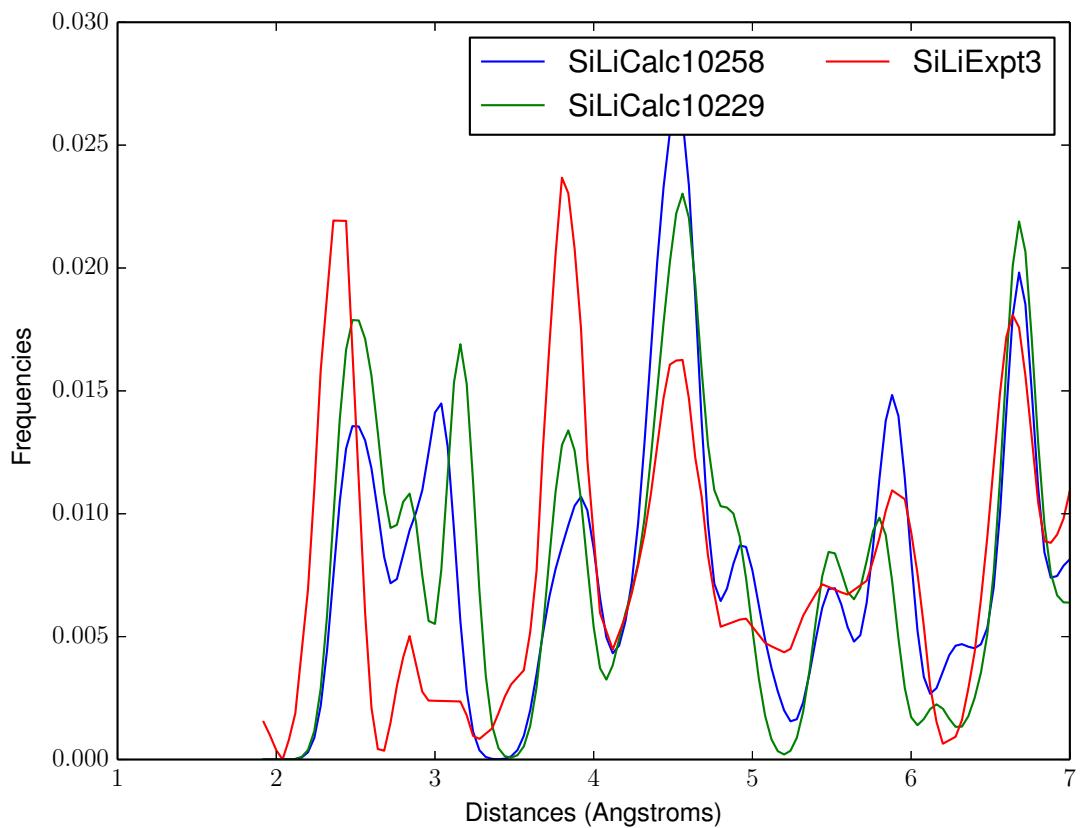


Figure 57: PCA Matches: SiLiExpt3, SiLiCalc10258, SiLiCalc10229

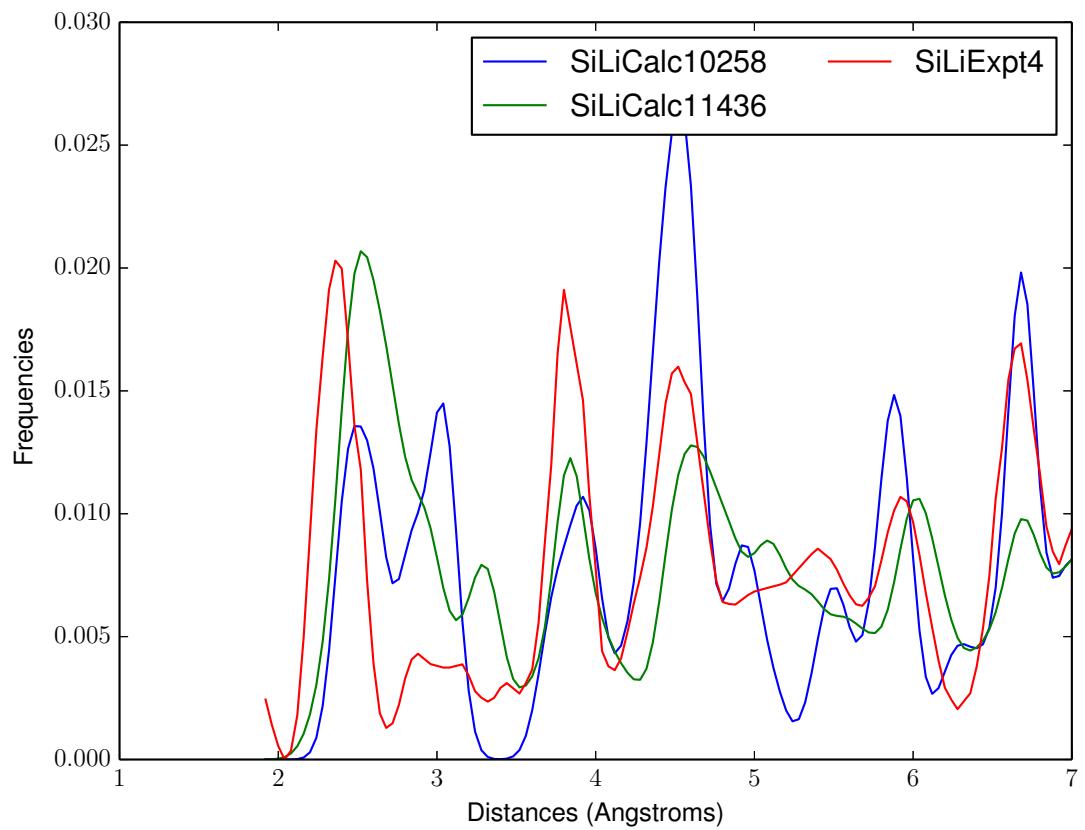


Figure 58: PCA Matches: SiLiExpt4, SiLiCalc10258, SiLiCalc11436

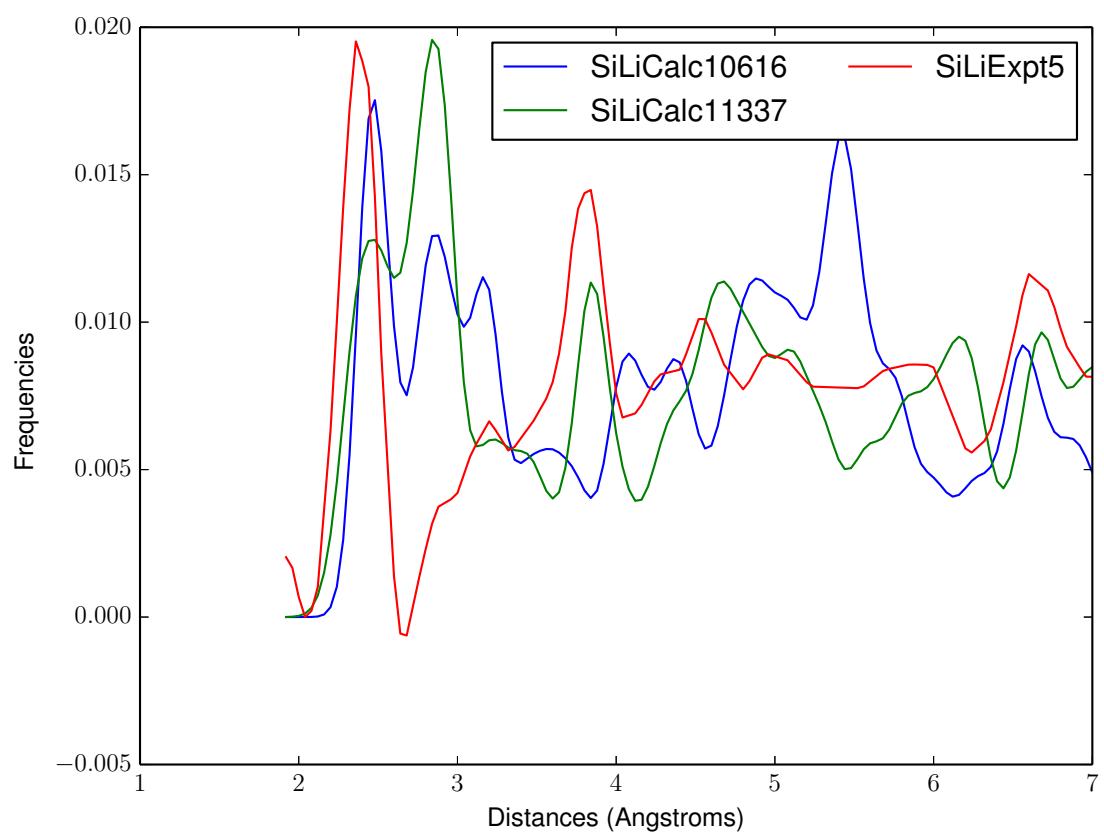


Figure 59: PCA Matches: SiLiExpt5, SiLiCalc10616, SiLiCalc11337

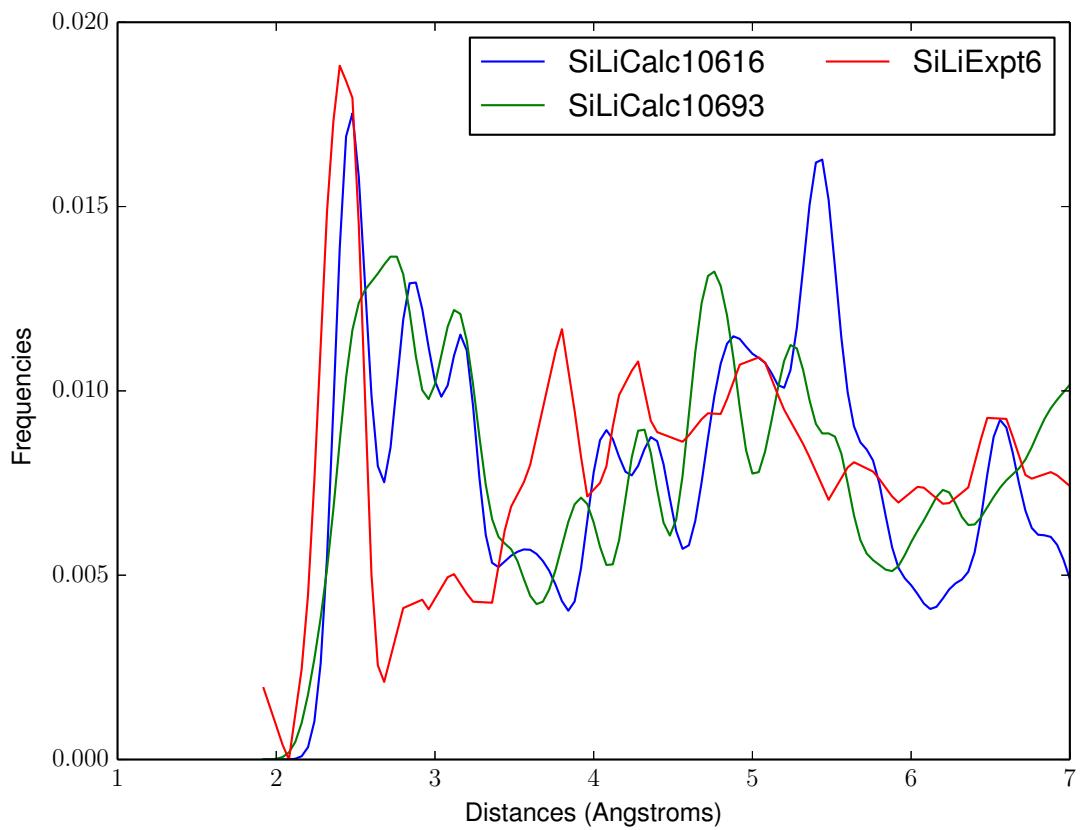


Figure 60: PCA Matches: SiLiExpt6, SiLiCalc10616, SiLiCalc10693

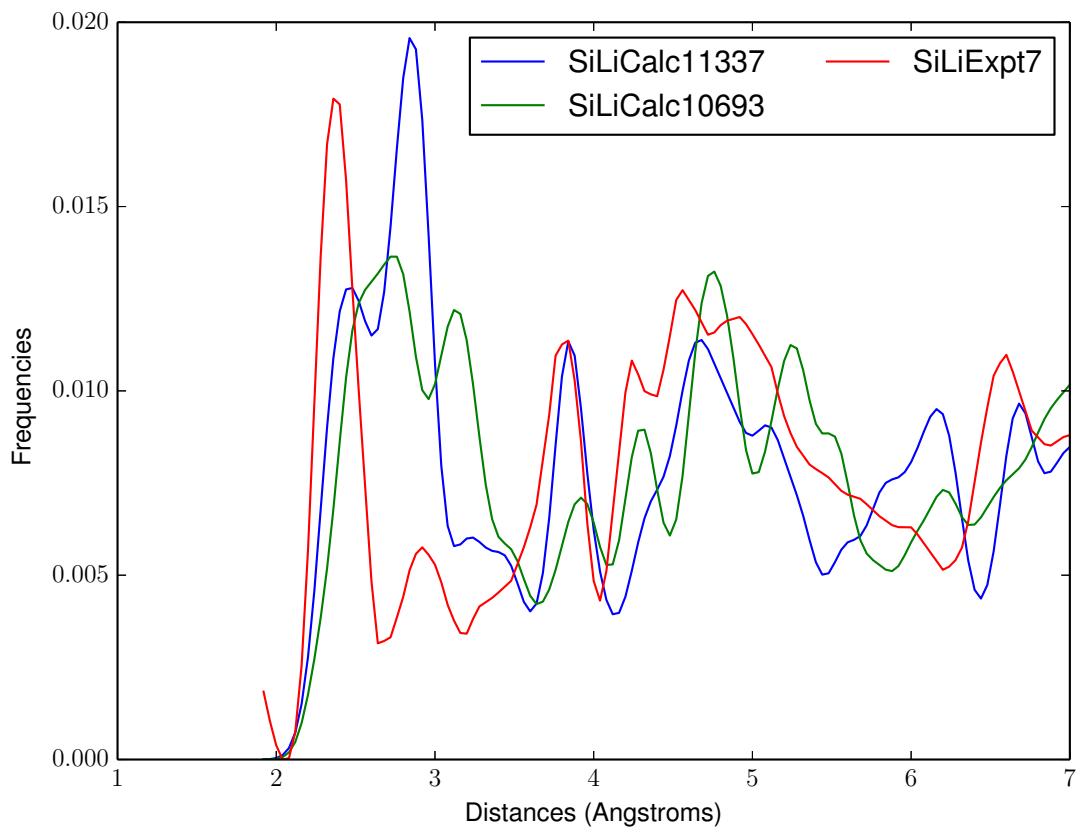


Figure 61: PCA Matches: SiLiExpt7, SiLiCalc10693, SiLiCalc11337

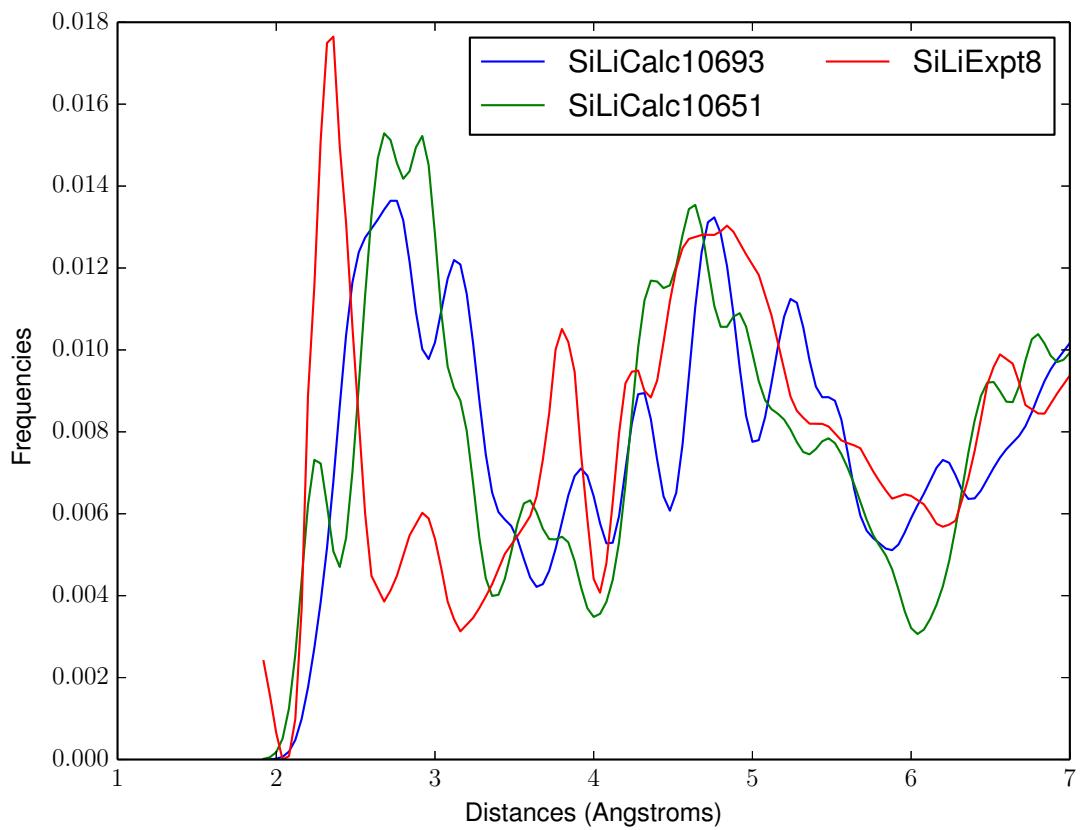


Figure 62: PCA Matches: SiLiExpt8, SiLiCalc10693, SiLiCalc10651

4.6 Synthetic Experimental Image Recognition

Here we run the alternative analysis approach outlined in section 3. The exact method is described below.

```
For i = 1 to nSamples
    calculatedImage = randomlySelectFrom(calculatedImages)
    experimentalImage = addNoise(calculatedImage)
    bestMatchImage = findBestMatch(experimentalImage)

    If bestMatchImage == calculatedImage Then
        increment successCounter

Accuracy = successCounter / nSamples
```

Below we consider findBestMatch functions that find the minimum L1 and L2 norm in PCA space for ?all? principal components.

Here we check how the accuracy varies as we increase the number of principal components. We see that the accuracy does not change much after 15 principal components. This aligns with our conclusion from the cumulative variance chart that 15 principal components are sufficient to describe much of the variance in the data set.

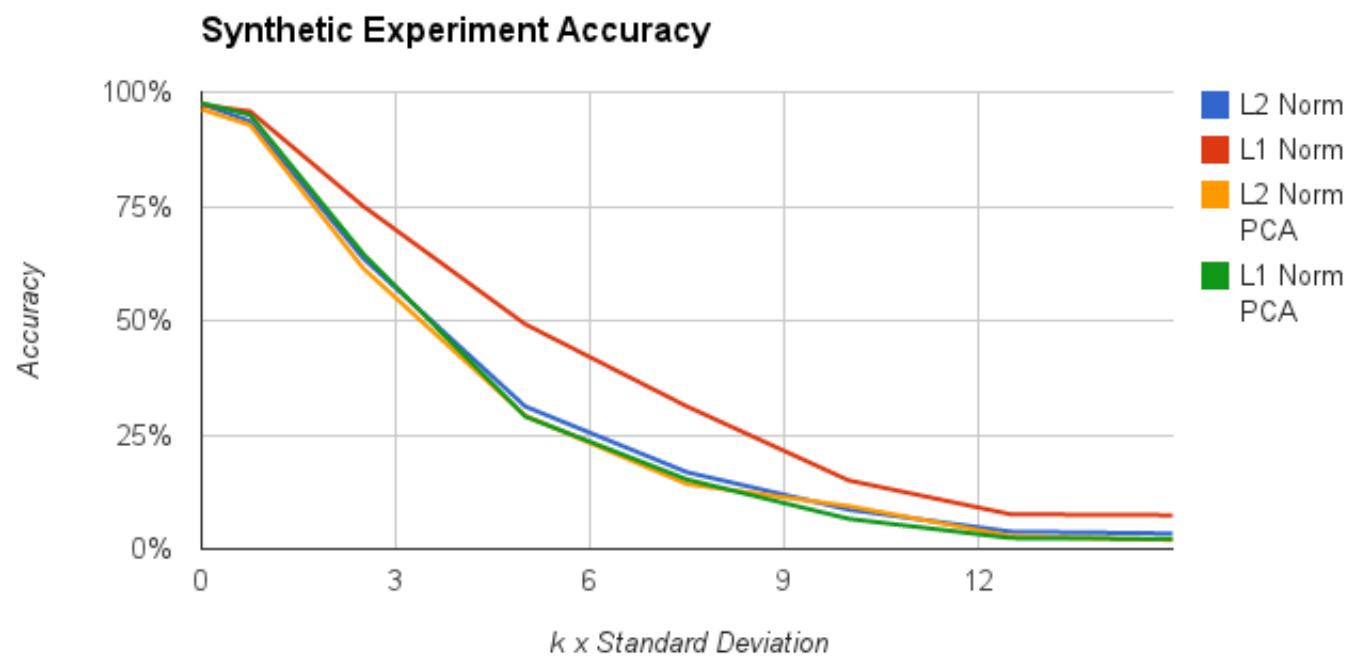


Figure 63: Synthetic Experimental Images Accuracy

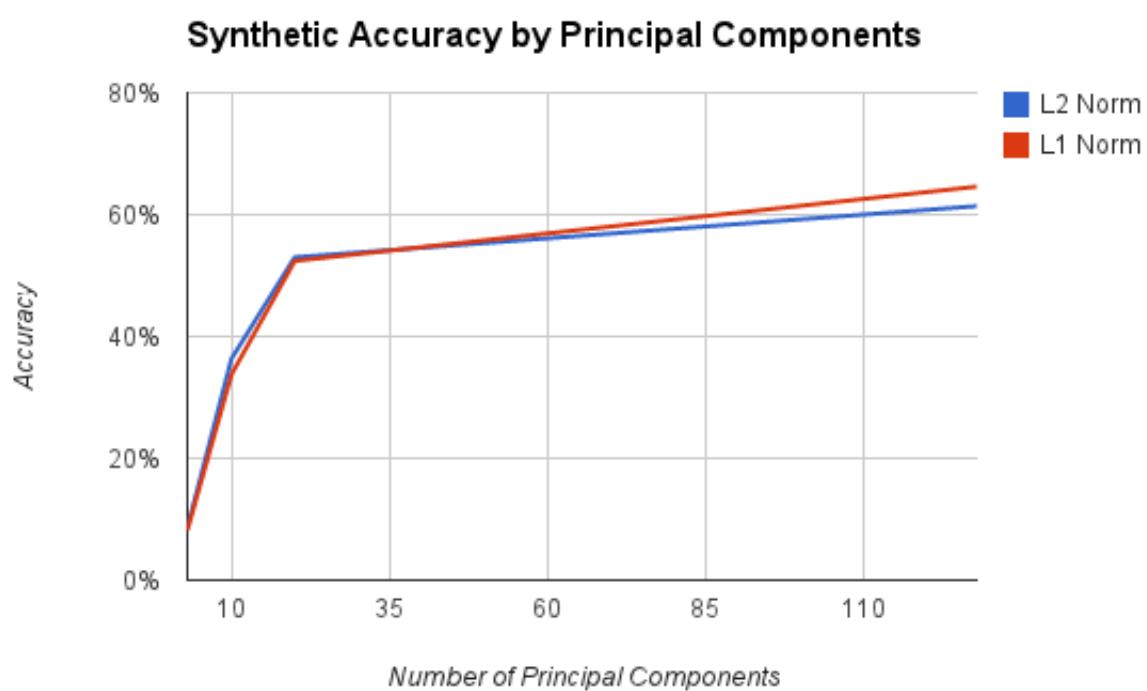


Figure 64: Accuracy vs Number of Principal Components

5 Recognition Using Sparse Representations

Following the lead of Wright, et al. in "Robust Face Recognition via Sparse Representation", we apply the method of recognition via sparse representation to the xray images. The idea is that we try to express the experimental image as a sparse linear combination of calculated images. If the solution is sufficiently sparse, then the sparse solution can be found by minimizing the L_1 norm of the coefficients. The best match is then are the images that have non-zero coefficients. One application of this that is outlined in Wright et al's paper is to choose the best match as the image that produces the lowest error after being multiplied by its coefficient.

$$\begin{pmatrix} c_{11} & c_{12} & c_{13} & \dots & c_{1n} \\ \dots & \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & c_{m3} & \dots & c_{mn} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \dots \\ y_n \end{pmatrix} = \begin{pmatrix} x_1 \\ \dots \\ x_m \end{pmatrix} + \begin{pmatrix} e_1 \\ \dots \\ e_m \end{pmatrix}$$

$c_i : i^{th}$ Calculated Image

$C = [c_1 c_2 \dots c_n]$

x : Target Image

We wish to find the solution that is sparsest with bounded error. This can be found via a second order cone programming problem.

$$\hat{y} = \arg \min_y \|y\|_1 \quad | \quad \|Cy - x\|_2 \leq \epsilon$$

5.1 Experimental Image Recognition

!!! Need to rerun the results with the new smoothing coefficient. !!! Here we show the best matches for the experimental images using the sparse representation method. This method is able to recover the correct matches for the known experimental and calculated pairs. The matches for most of the experimental images seem reasonable as well. For SiLi experiments 6 through 8 though, the matches are quite bad.

Experiment	Match
ExptGaAs	CalcGaAs
ExptInAs	CalcInAs
SiLiExpt1	SiLiCalc10001
SiLiExpt2	SiLiCalc10001
SiLiExpt3	SiLiCalc10001
SiLiExpt4	SiLiCalc10003
SiLiExpt5	SiLiCalc10003
SiLiExpt6	SiLiCalc10616
SiLiExpt7	SiLiCalc10382
SiLiExpt8	SiLiCalc10382

Table 6: Experimental Image Recognition

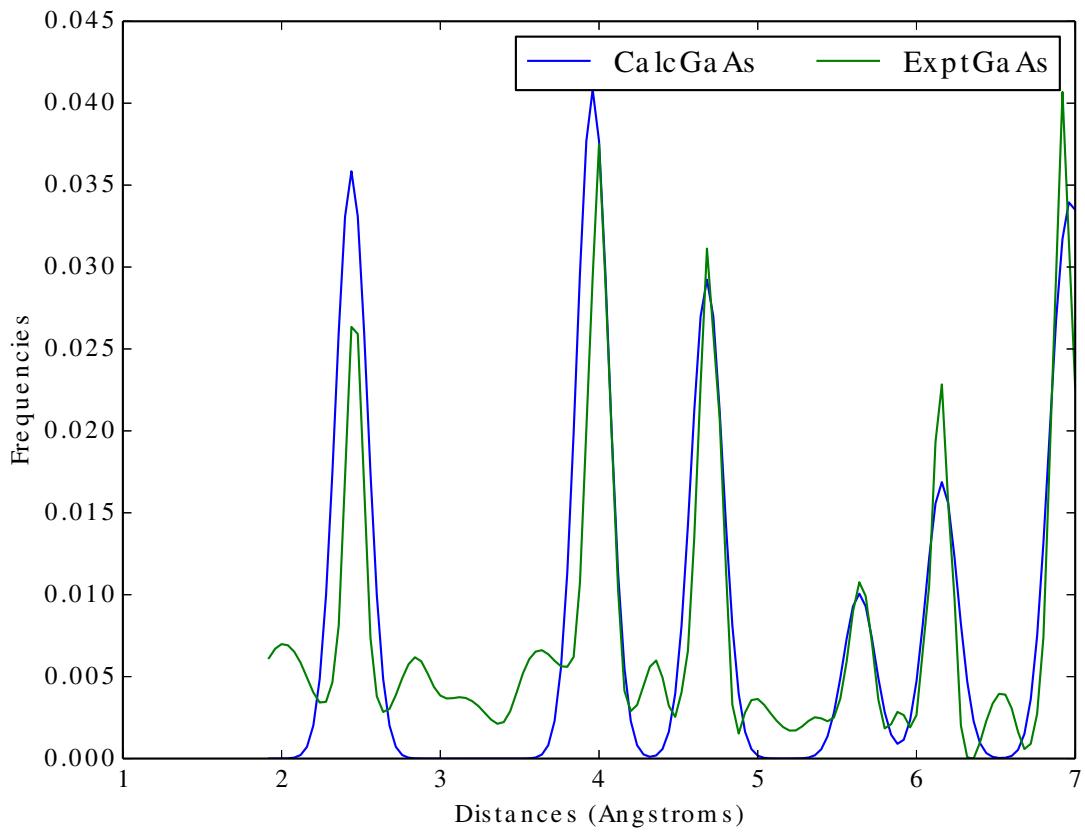


Figure 65: ExptGaAs, CalcGaAs

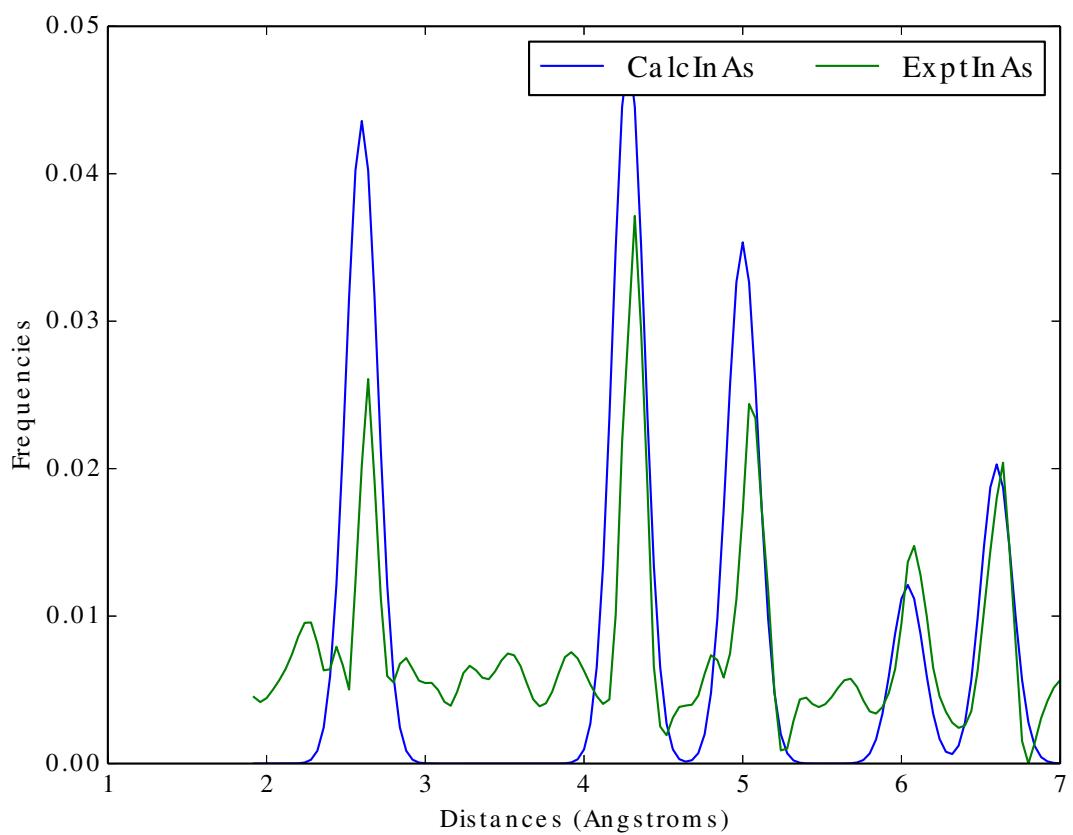


Figure 66: ExptInAs, CalcInAs

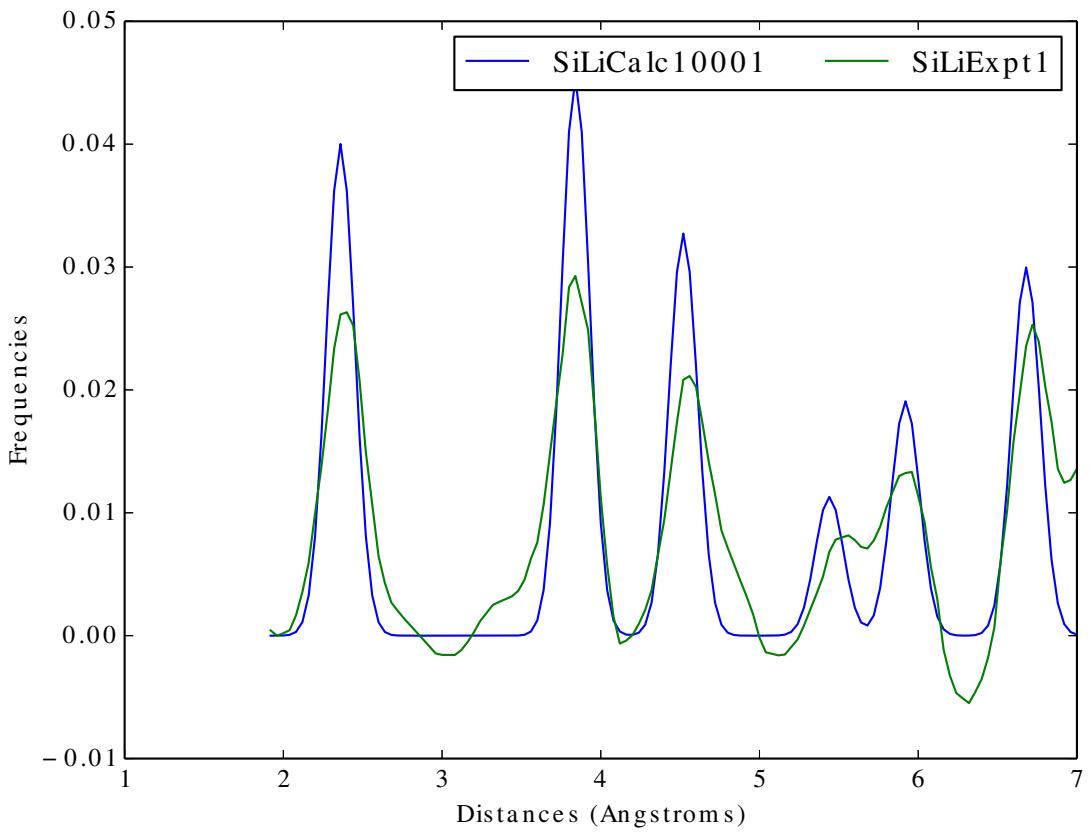


Figure 67: SiLiExpt1, SiLiCalc10001

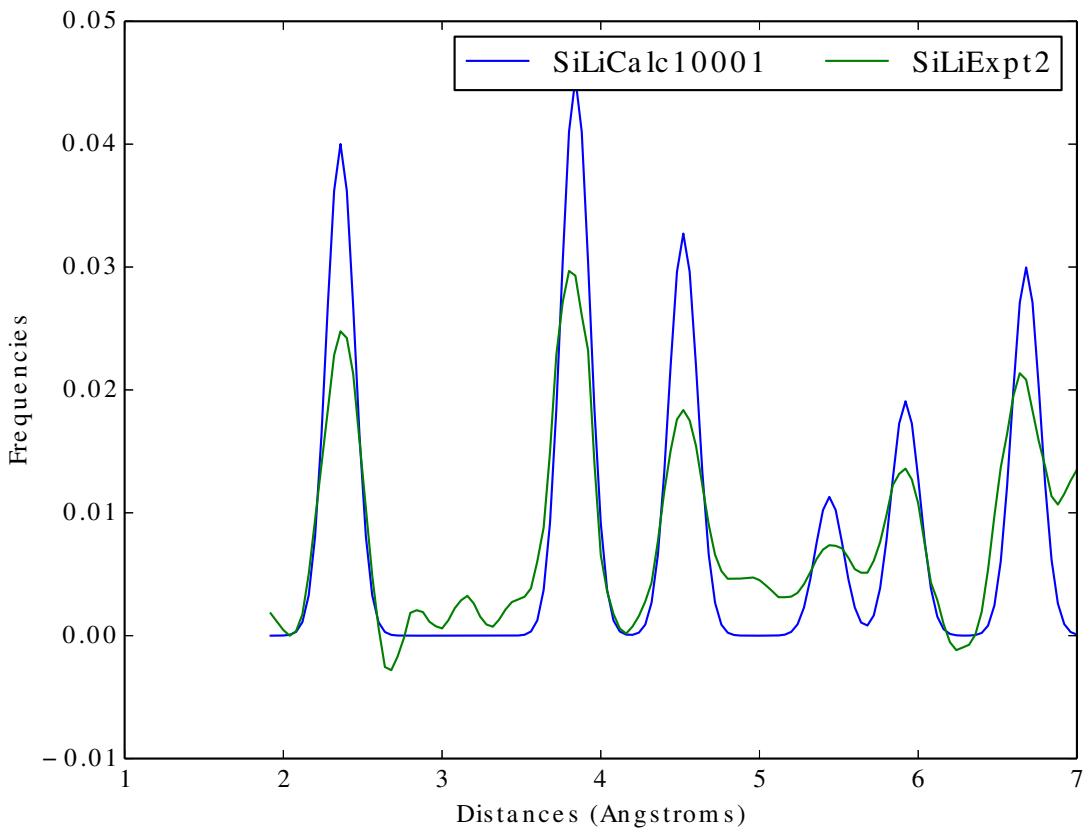


Figure 68: SiLiExpt2, SiLiCalc10001

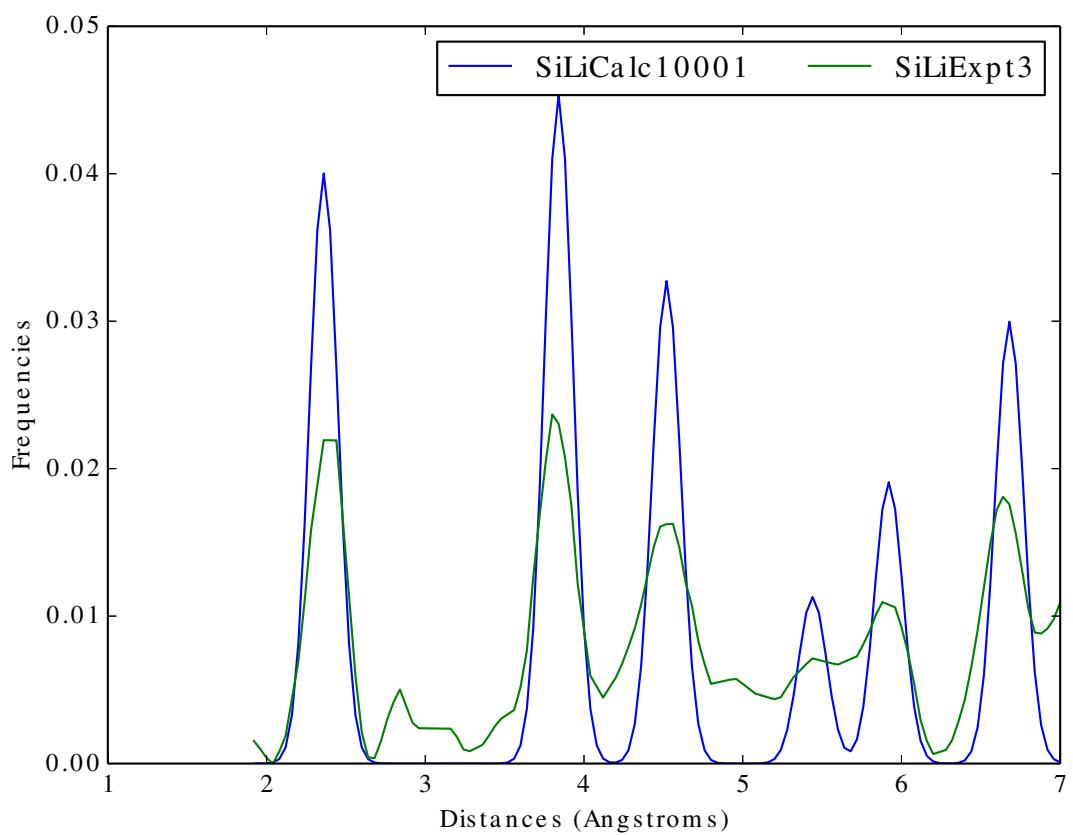


Figure 69: SiLiExpt3, SiLiCalc10001

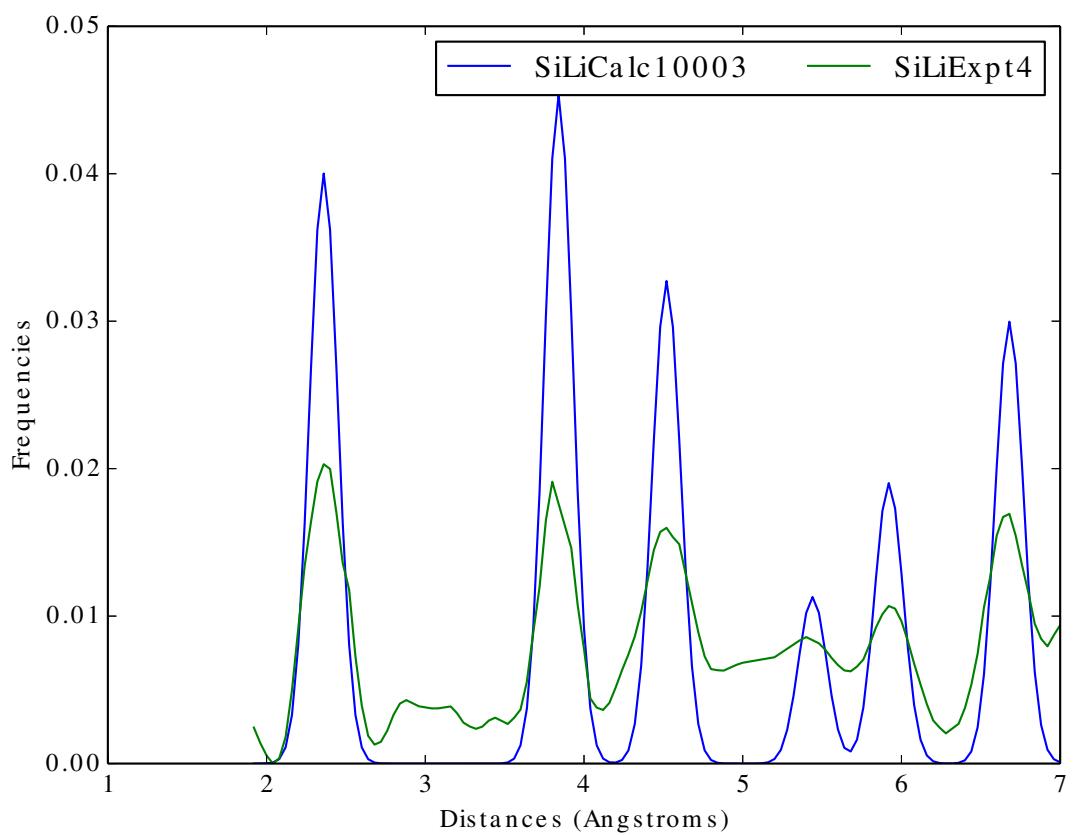


Figure 70: SiLiExpt4, SiLiCalc10003

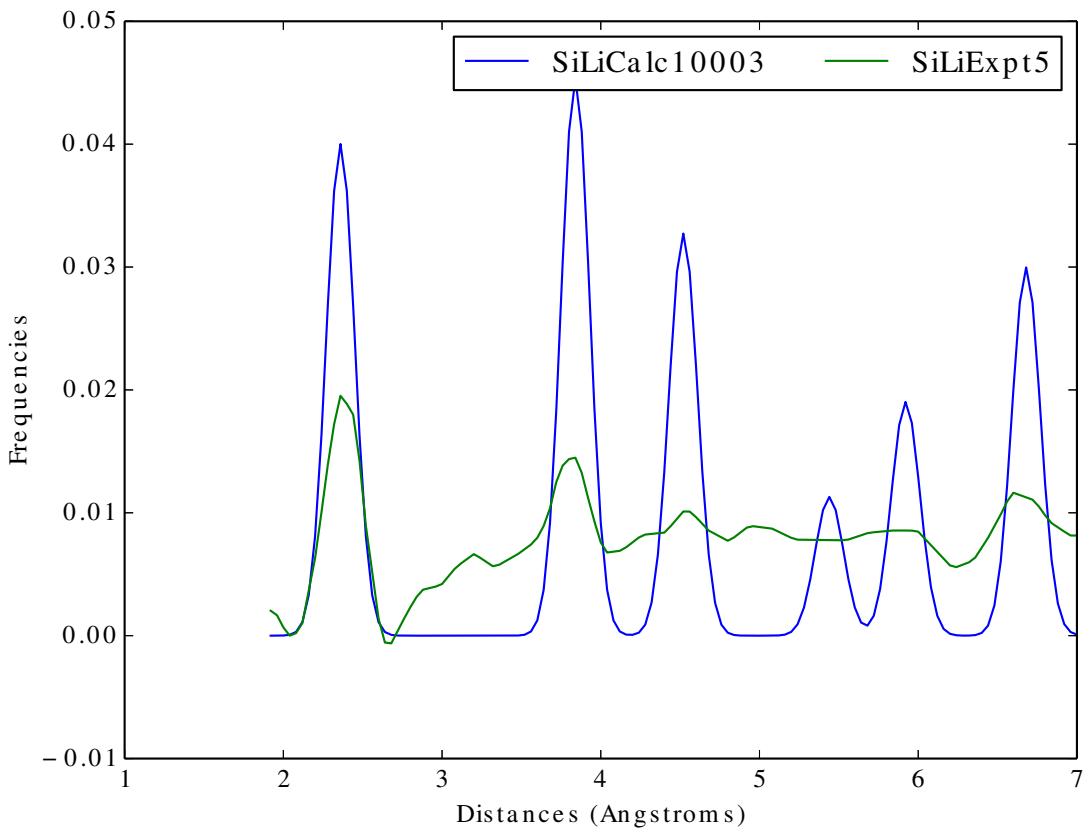


Figure 71: SiLiExpt5, SiLiCalc10003

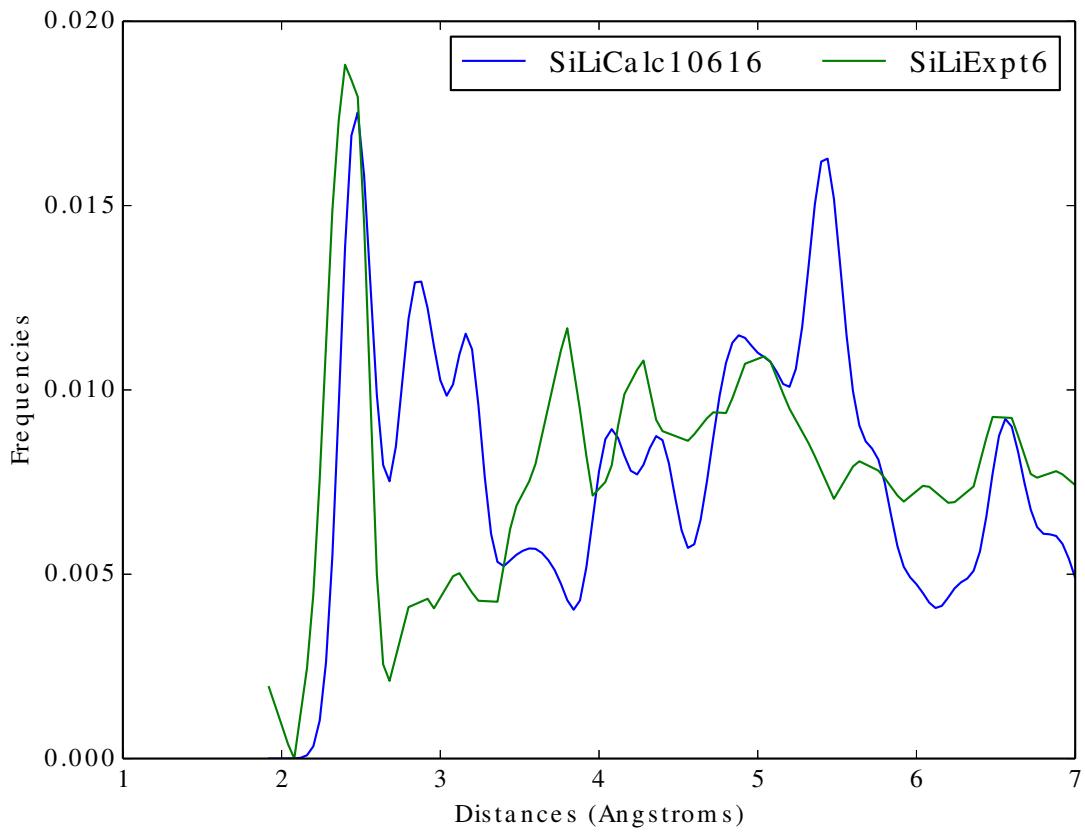


Figure 72: SiLiExpt6, SiLiCalc10616

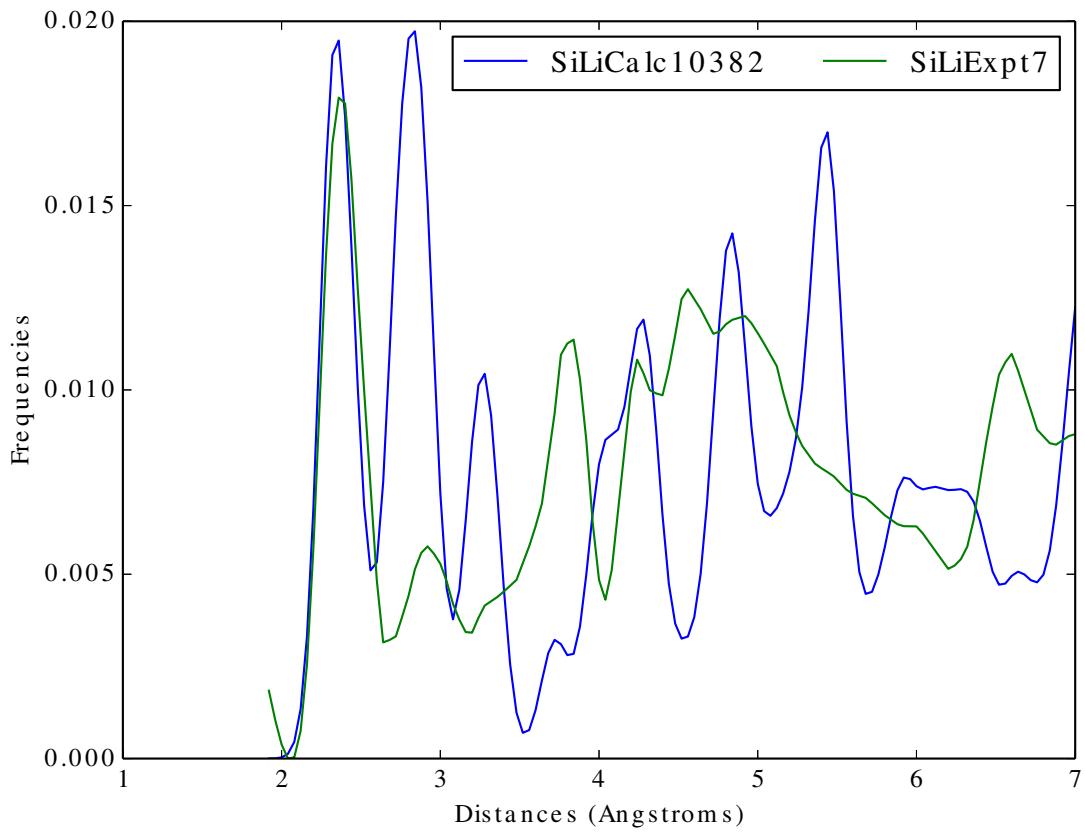


Figure 73: SiLiExpt7, SiLiCalc10382

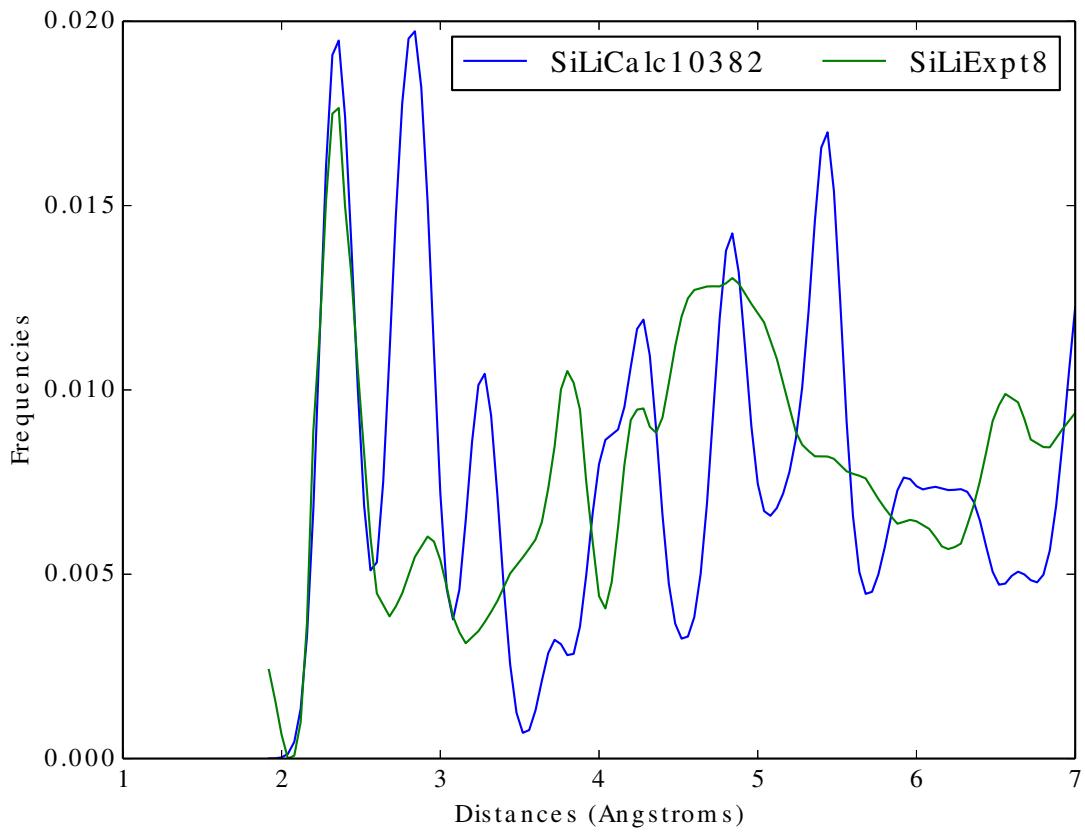


Figure 74: SiLiExpt8, SiLiCalc10382

5.2 Synthetic Experimental Image Recognition

Running the synthetic experimental image recognition analysis with the sparse representation approach yields surprisingly bad results. Below the results are plotted with different error bounds.

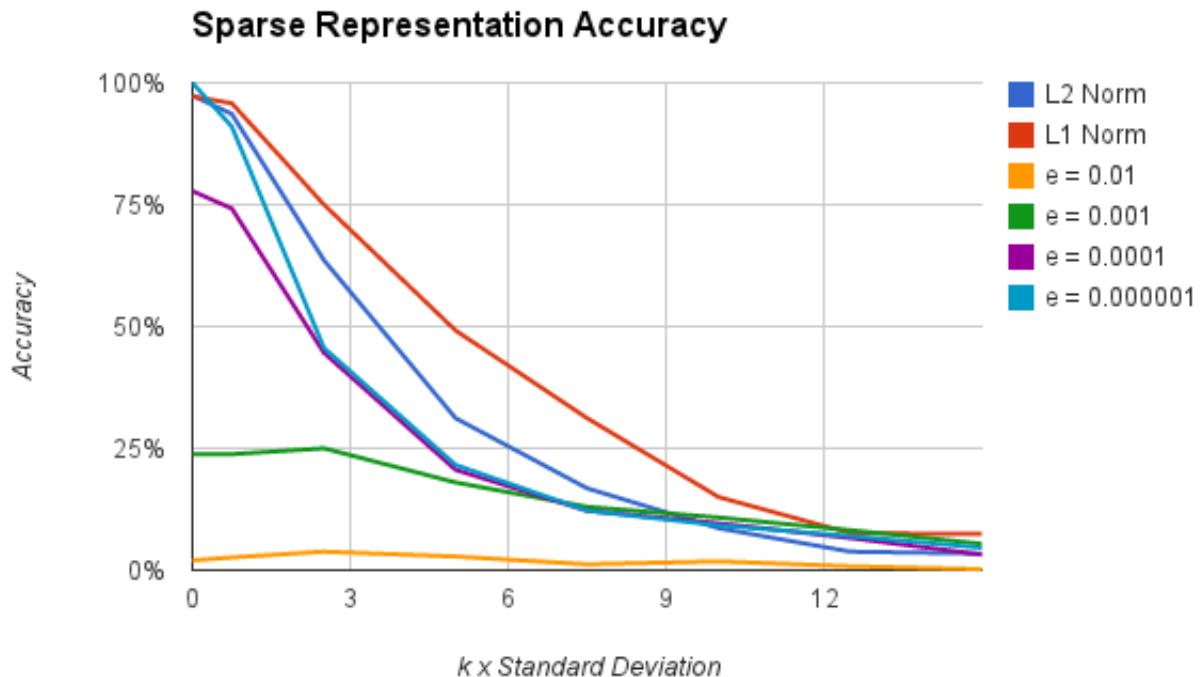


Figure 75: Synthetic Experimental Image Recognition Accuracy

6 Code

The code used to produce the results in this document as well as the document itself can be found here: <https://github.com/chadvoegele/xraysprectrapy>

7 Sources

http://en.wikipedia.org/wiki/Atom_vibrations

http://en.wikipedia.org/wiki/Radial_distribution_function

http://en.wikipedia.org/wiki/Weierstrass_transform

http://matplotlib.org/api/mlab_api.html

http://en.wikipedia.org/wiki/Principal_components_analysis

First Principles Simulations of the Electrochemical Lithiation and Delithiation of Faceted Crystalline Silicon
of Faceted Crystalline Silicon

Maria K. Y. Chan, C. Wolverton, and Jeffrey P. Greeley

Journal of the American Chemical Society 2012 134 (35), 14362-14374

Pair Distribution Functions Analysis.

Petkov, V. 2012.

Characterization of Materials. 1-14.

"Face recognition using eigenfaces,"

Turk, M.A.; Pentland, A.P.,

Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91.,

IEEE Computer Society Conference on , vol., no., pp.586,591, 3-6 Jun 1991

"Robust Face Recognition via Sparse Representation,"

Wright, J.; Yang, A.Y.; Ganesh, A.; Sastry, S.S.; Yi Ma,

Pattern Analysis and Machine Intelligence, IEEE Transactions on ,

vol.31, no.2, pp.210,227, Feb. 2009