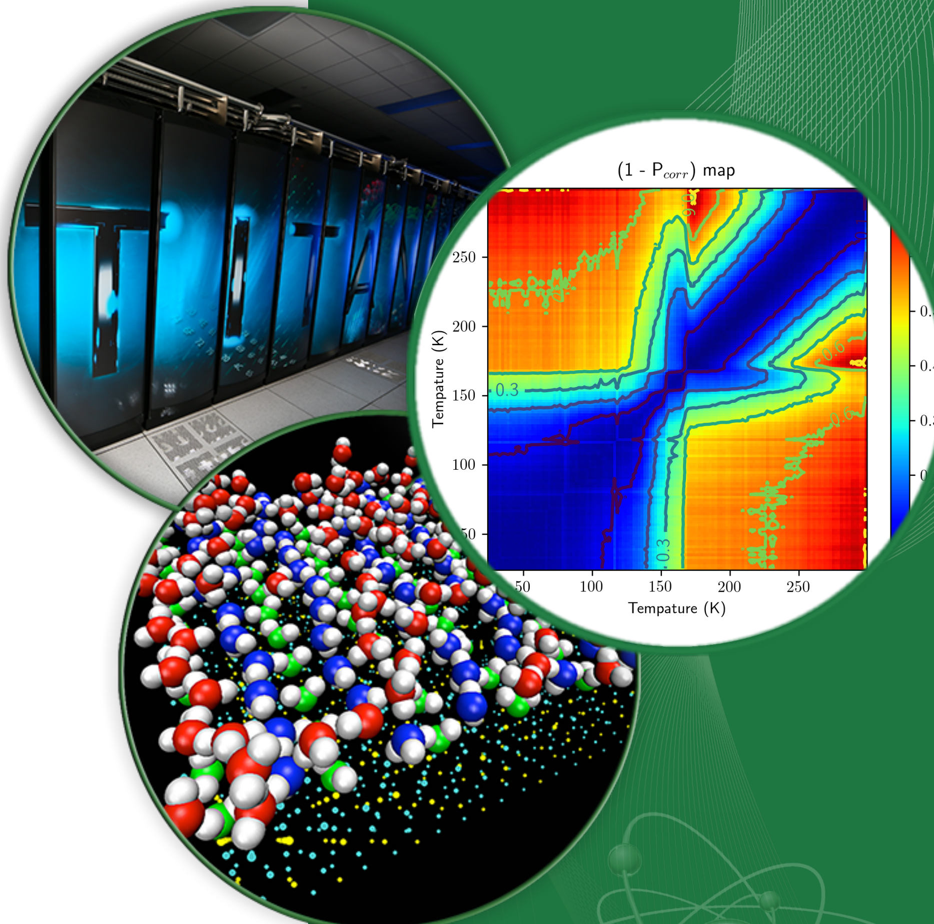
# Challenge 4:

# Automated Discovery of Temperature Dependent Structural Change




Yaohua Liu

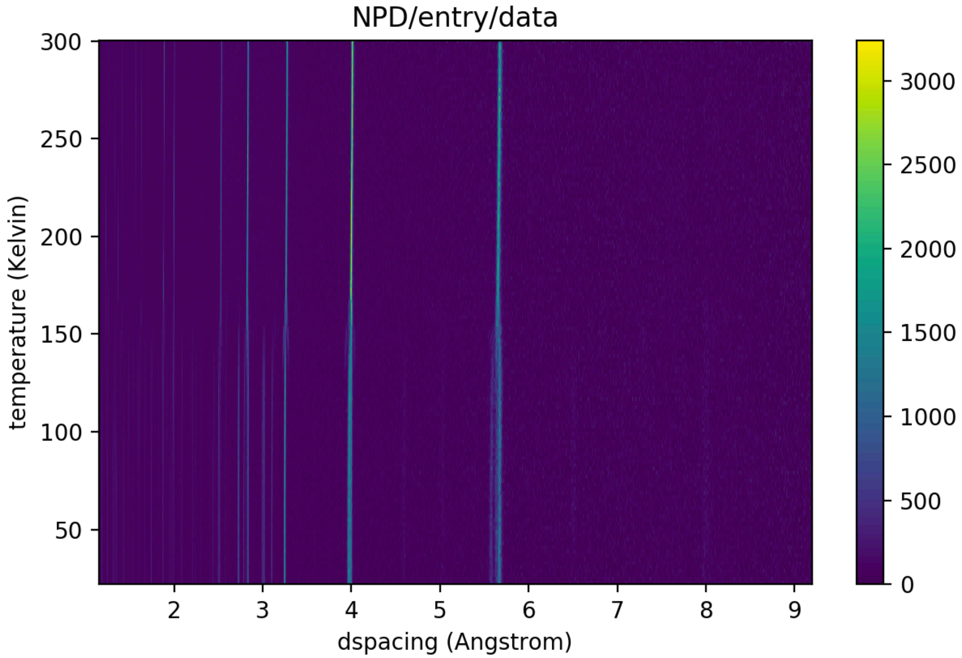
Quantum Condensed Matter Division, ORNL

Yawei Hui

Computer Science and Mathematics Division, ORNL

# Introduction

2D Data: $I = I(d, T)$
Parameterized study of neutron powder diffraction experiments as a function of temperature.

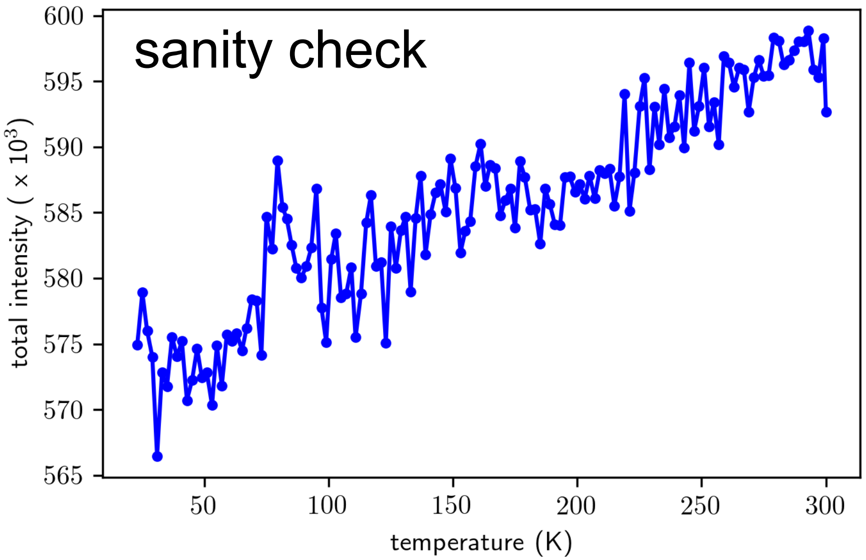


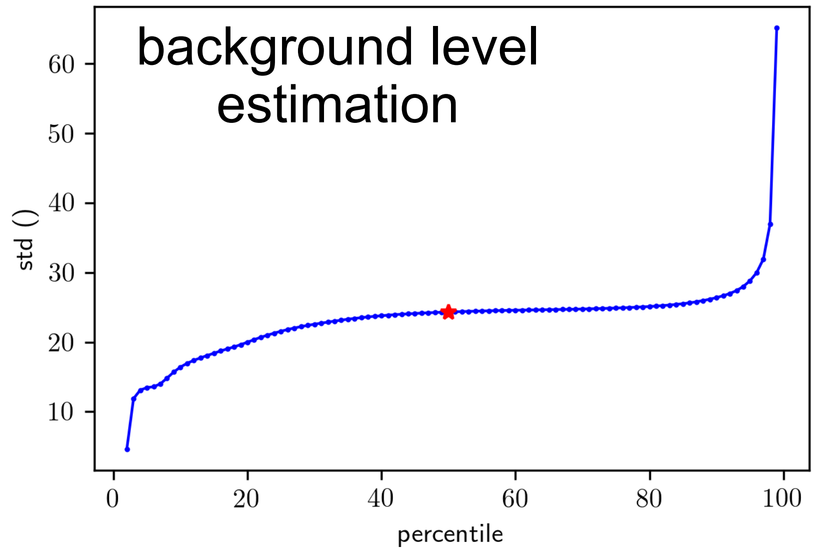

Four tasks (rephrased):

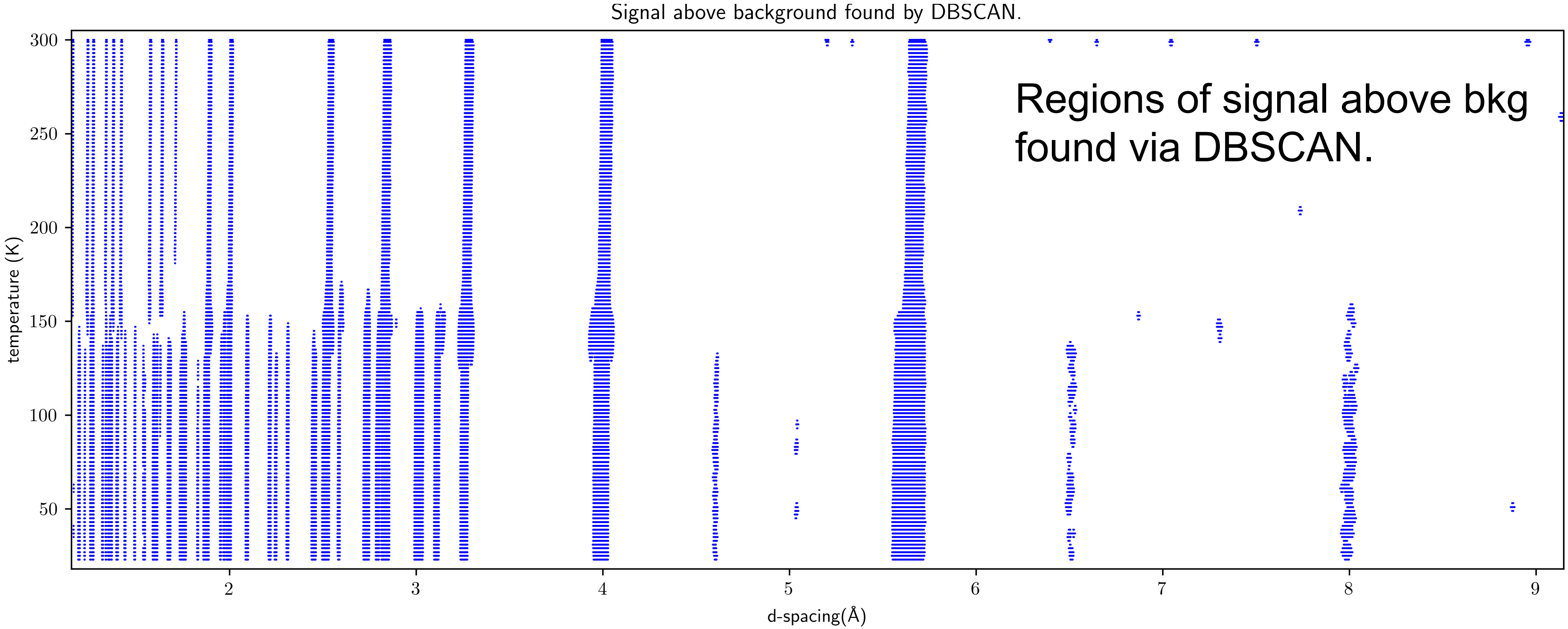
1. Provides the structural transition temperature.
2. Provides the intensity, center and width on the signal between 3.2 and 3.3 Å as a function of temperature.
3. Provides the signal information for all peaks at a given temperature.
4. Indicates if a phase transition occurred between two adjacent temperatures. Generates information for all peaks that are at least 1.5x above background in under 5 seconds.

We implemented algorithms in Python using the **numpy** library and the **jupyter** notebook. We have also used the following packages: **scipy**, **sklearn, matplotlib, nexusformat.** We have also tested **multiprocessing, numba** and **ipyparallel** to speed up for task #4**.**

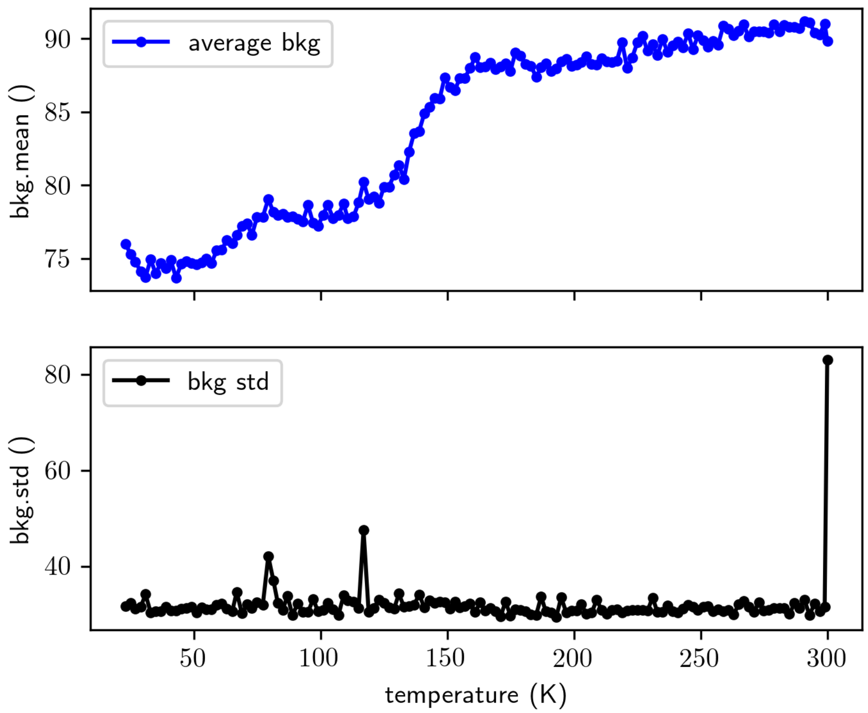
# Q0: Characteristics of the data

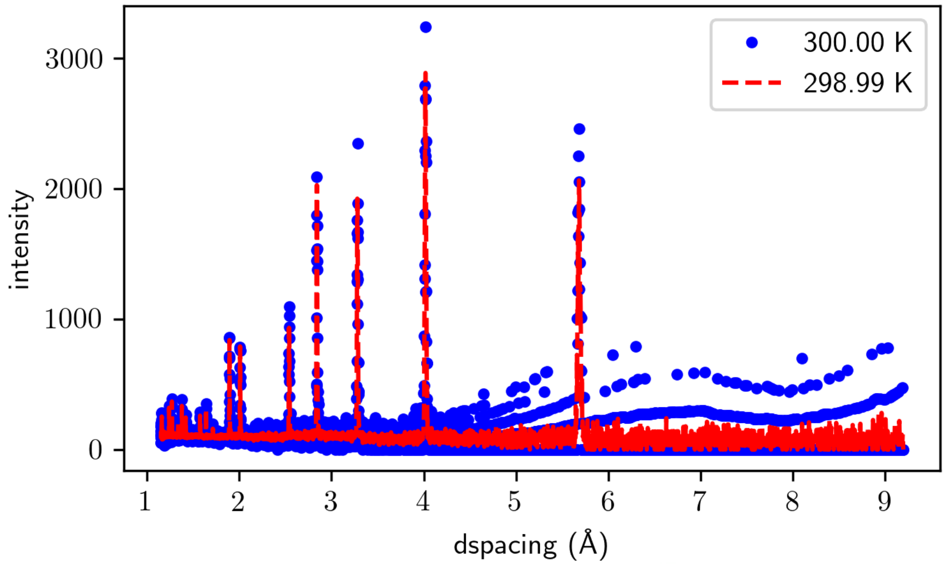


# Q0: Characteristics of the data- cont.



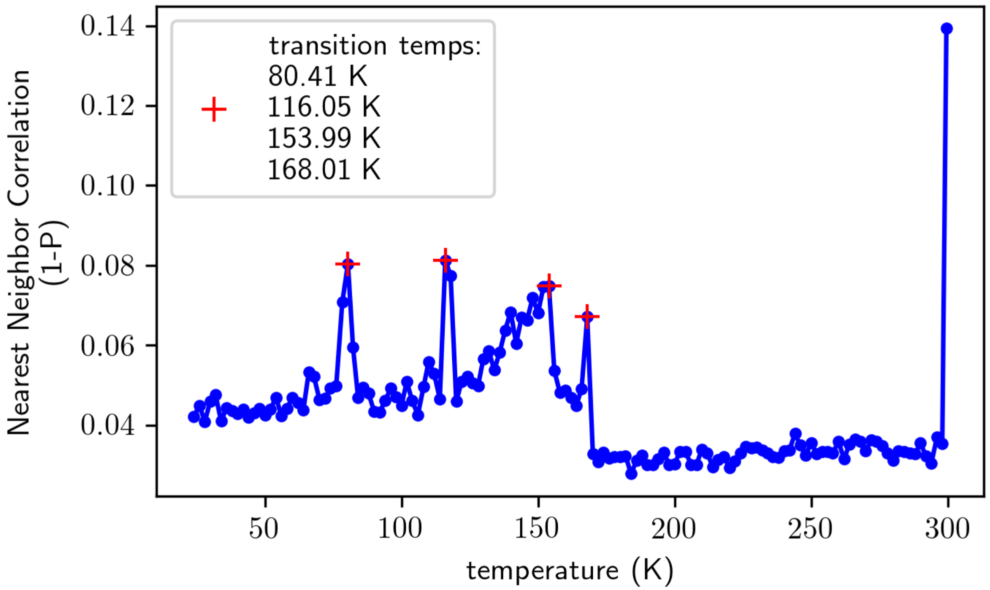

- Statistical analysis on the background found by DBSCAN shows an anomaly at the last temperature, indicated by a high std.
  - ✓ *caused by the high noise level at 300 K.*

# Q1: the structural transition temperature

**Key algorithms:**

**(1) Pearson's correlation function between two temperatures.**

**(2) Wavelet transformation to find the peak location in Pearson correlation function**

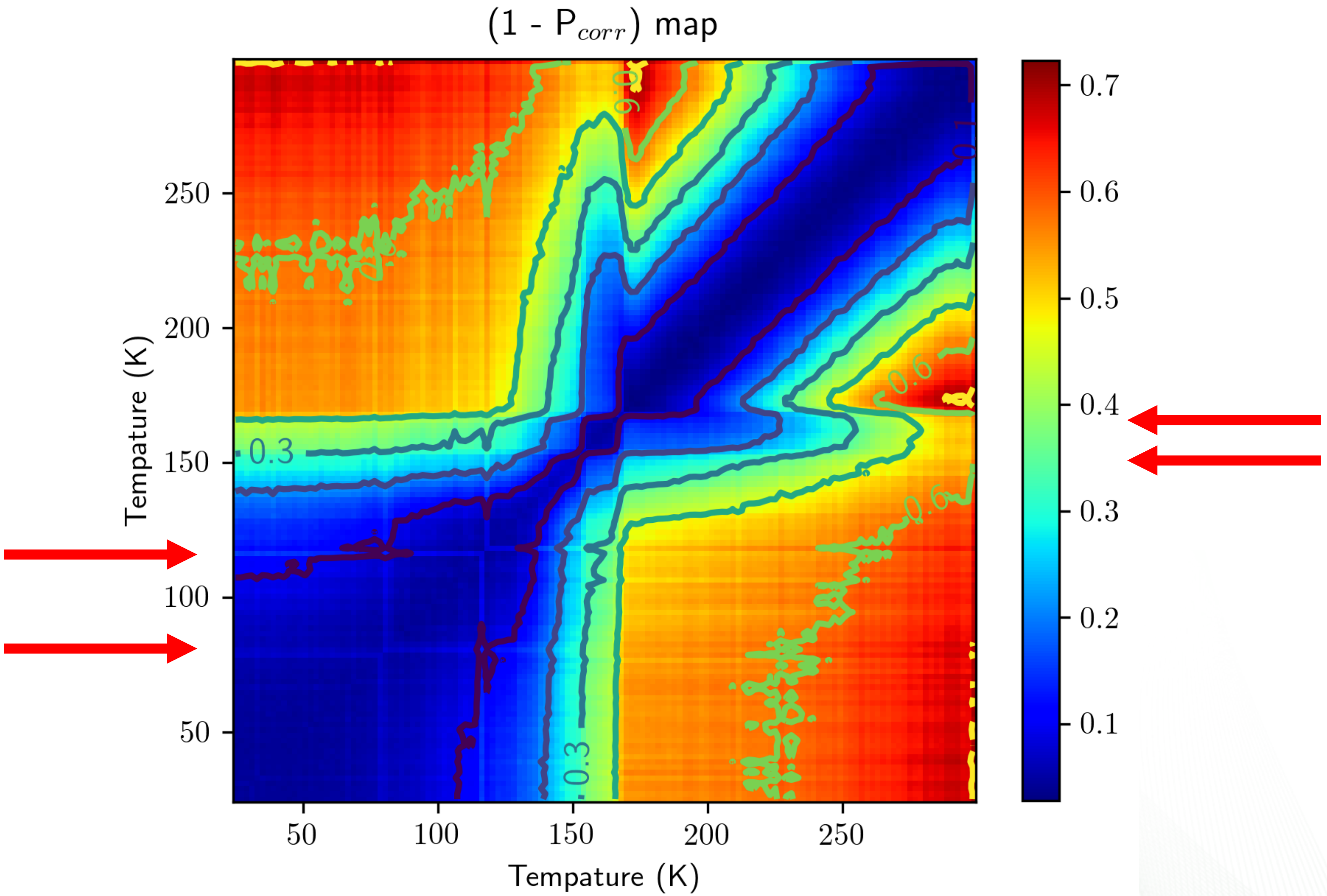



$$P = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Calculated (1-P) and found 4 possible transition temperatures.

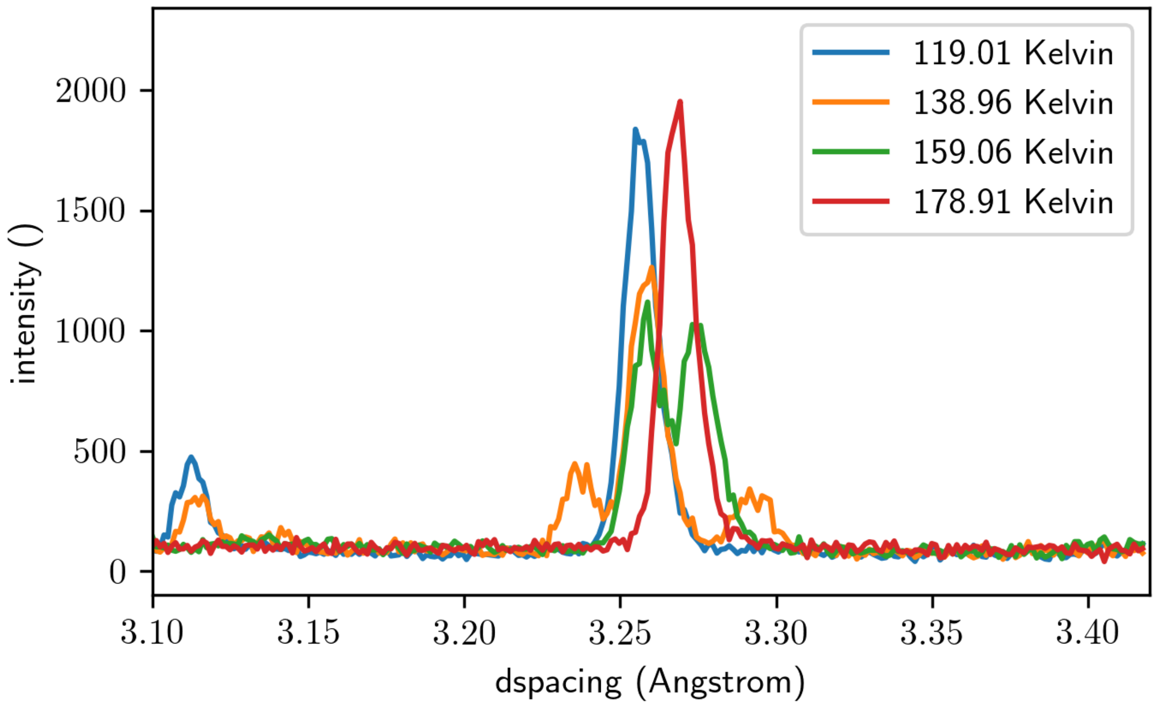
The spurious point between the last two temperatures was due to the high noise level in the 300 K dataset.

# Q1: the structural transition temperature




2D Pearson map to better visualize the results.

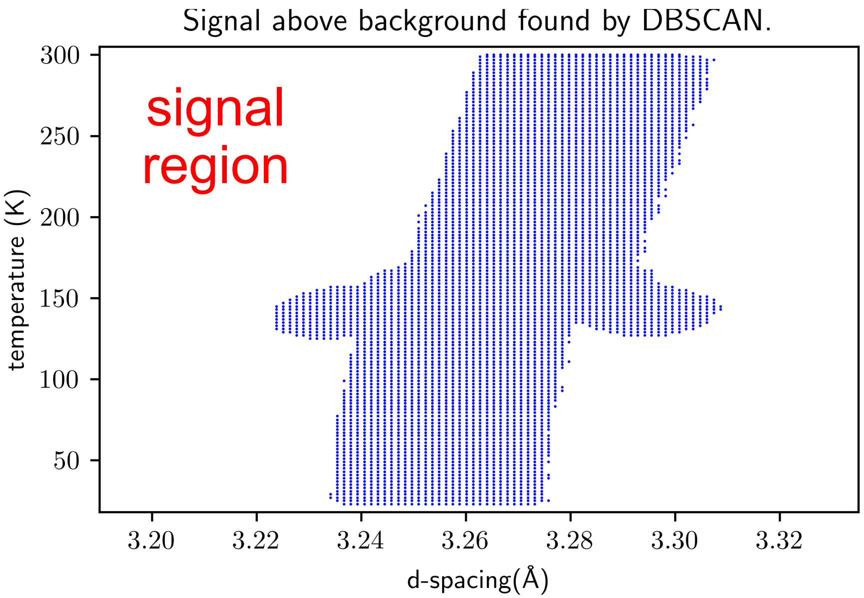
# Q2: Signal between 3.22 and 3.32 Å as a function of T

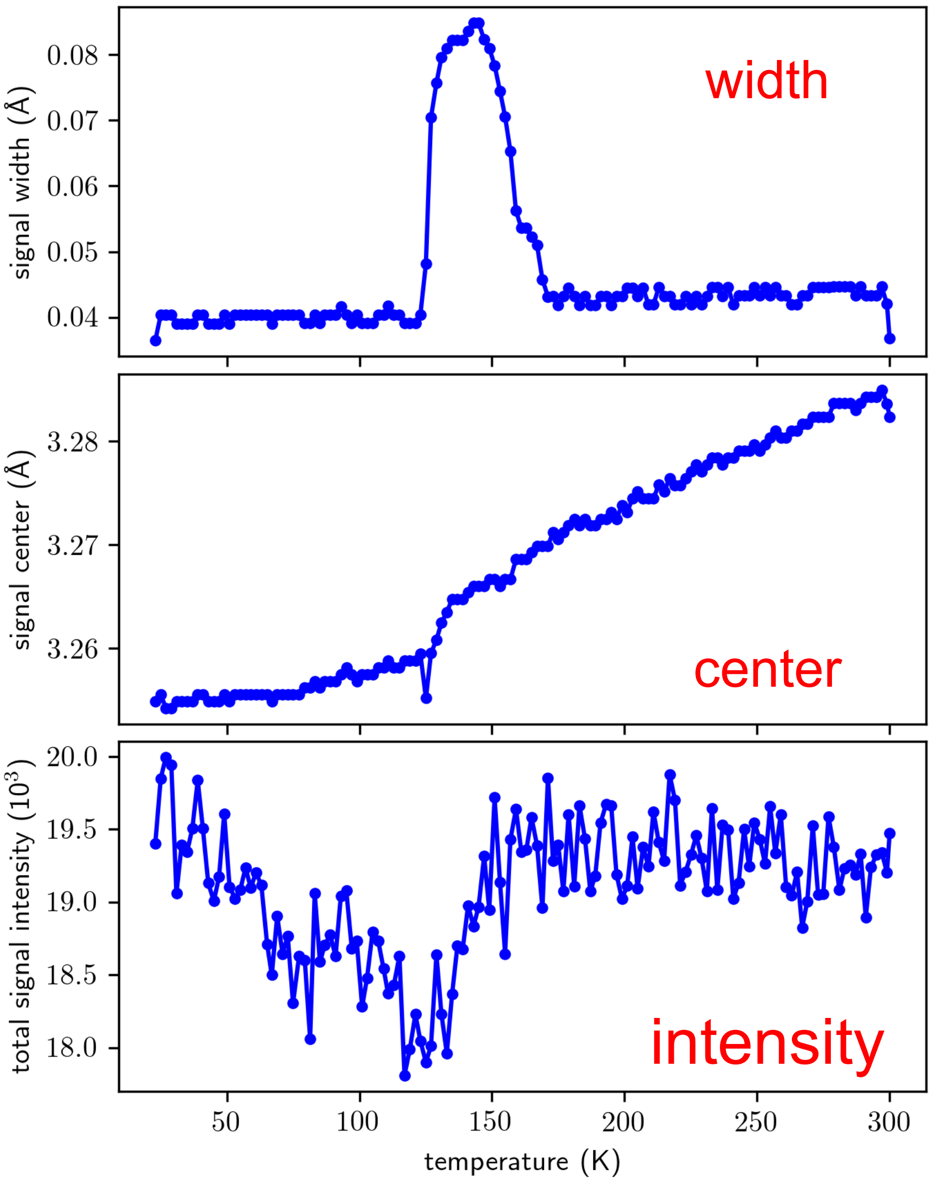



- In a wide range of temperatures, there are multiple peaks between 3.2 and 3.3 Å. The center and characteristic width of *the peak* are not well defined.
- One could find the number of peaks and then fit the data with multiple peaks, then the question becomes a simpler version of Q3.

- We used a clustering method to extract the signal from the background, regardless of the number of peaks.

# Q2: Signal between 3.22 and 3.32 Å as a function of T

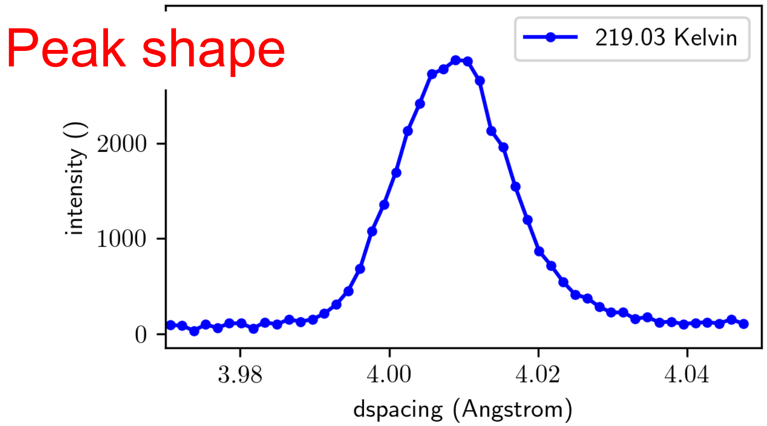


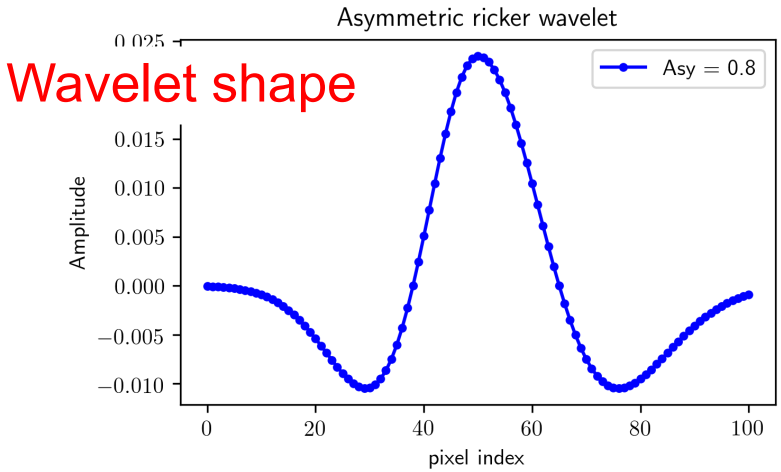



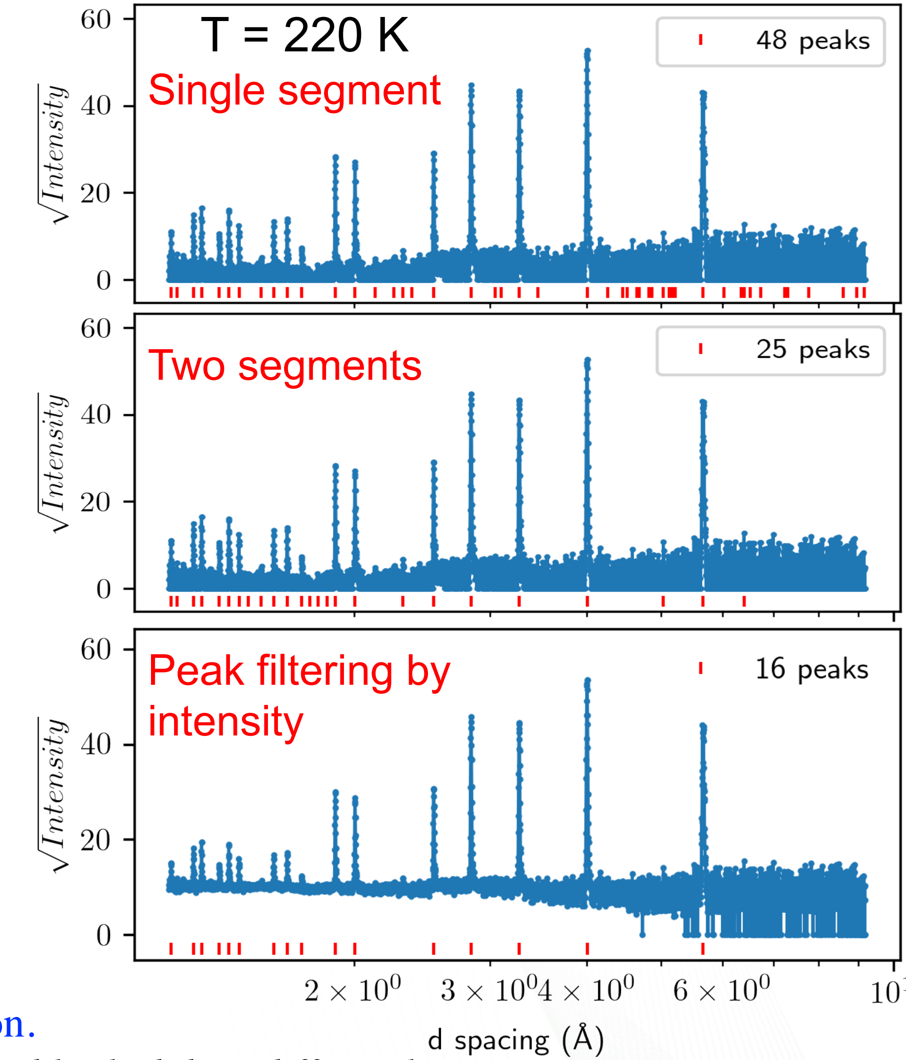

Key steps:

- Determine the background region and signal above background region via DBSCAN.
- Parameterize the background level as a polynomial function of temperature and d-spacing.
- Remove the background from the raw data to calculate the signal intensity.

# Q3 integrated intensity, center & width for all peaks at one T




(1) Wavelet transformation to find the peak location.
- *Used an asymmetric wavelet; treated low-d and high-d data differently.*

(2) Least square optimization for determining the peak amplitudes and locations.
- *Le Bail fitting; peak width was parameterized as a function of d-spacing.*

(3) Remove low intensity peaks.

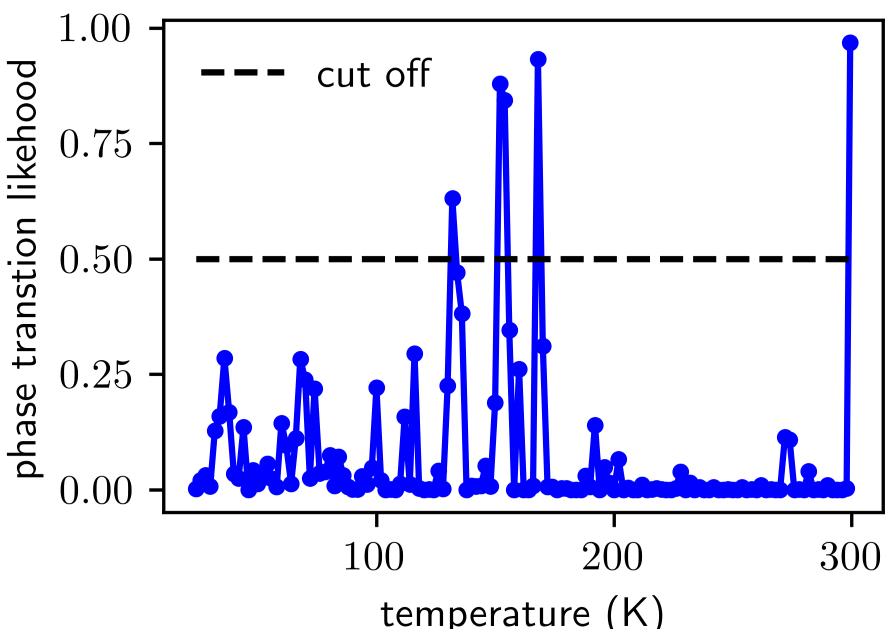
(4) 250 ms – 700 ms per temperature.

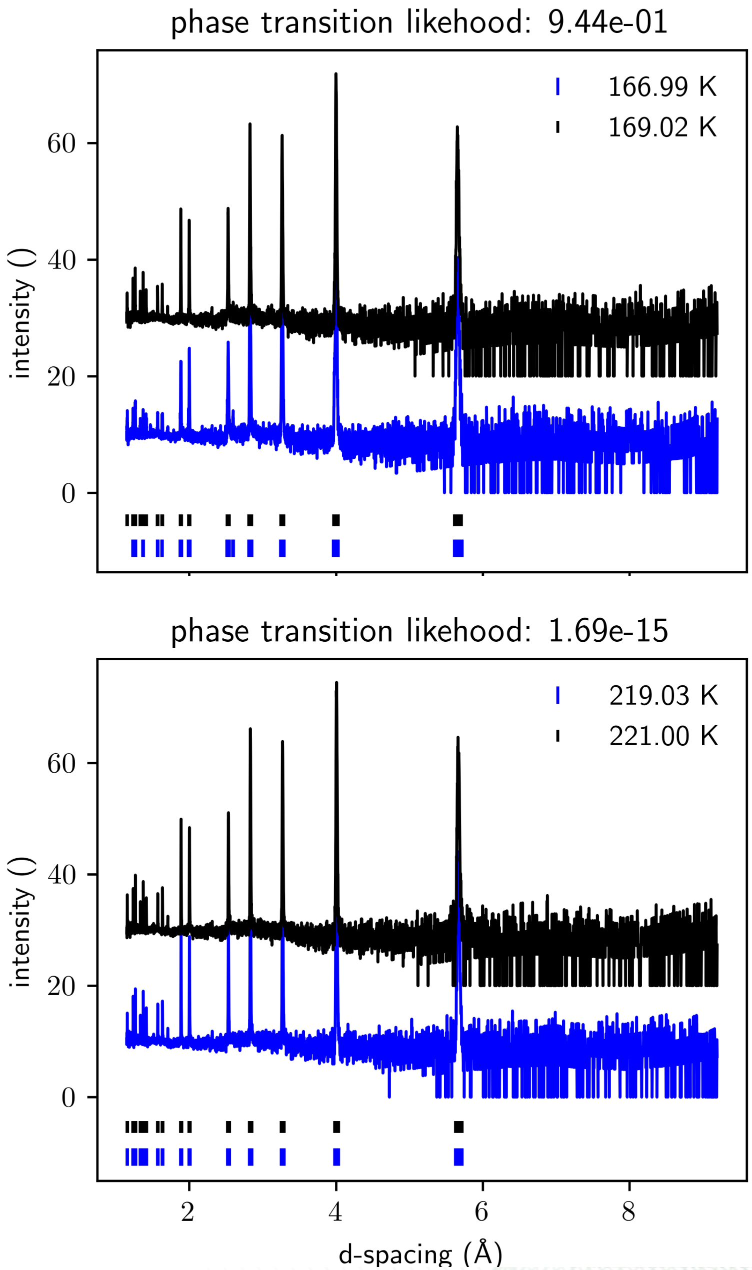
# Q4 – (a) Likehood of a phase transition occurred between two adjacent temperatures

(1) Determine regions of signal above the background (RoSaB) for each of the two adjacent temperatures via DBSCAN.

(2) Calculate numbers of shared and unshared signal bins between the two temperatures.
- More shared signal bin ➔ less likely
- More unshared signal bin ➔ more likely

In contrast to Q1, only data from two temperatures are used for calculation.



# Q4 – (b) Generates information for all peaks that are at least 1.5x above background within 5 seconds.

In [7]:
```
# peak table at a single temperature.
%timeit  find_peaks_flow(intsy, temps, dd_cs, par_val=240)
```

```
1 loop, best of 3: 319 ms per loop
```

In [8]:
```
# peak tables for all temperatures.
%timeit cal_peaktables_fullT(intsy, temps, dd_cs)
```

```
1 loop, best of 3: 1min 24s per loop
```