Automatic Peak Detection for Enthalpy Calculation

Hampton Creek – Logan Graham – August, 2015

The only thing you need to know

This software automatically calculates a potentially valuable data point (enthalpy) for a substance you are interested in. This saves time by replacing a manual process for humans (peak detection). This might allow you to scale up the process from dozens or hundreds of samples a day.

Summary

Enthalpy (the energy released in a reaction) could be a useful datapoint to determine whether or not a substance would make a good egg substitute. In particular, it helps to determine how well this substance would gel – which is useful to figure out if it would make good scrambled eggs, for example. Currently, we use differential scanning calorimetry (DSC) to calculate enthalpy from a reaction.

However, the current process to calculate enthalpy is slow. In order to measure enthalpy, right now you need a human to not only conduct an experiment which involves a denaturation, but then that human has to look at the data, then manually measure enthalpy by clicking on the points of the reaction. This takes time and is not scalable to hundreds of thousands of samples. It is still prone to human error.This hasn't been solved yet because it is still difficult to precisely pinpoint a reaction. Isolating the start and end points of a reaction is mostly a problem of human judgment.

I have devised a system that automatically calculates enthalpy (and other reaction statistics) for you. This system works by finding how fast the heat flow of the reaction is changing at certain points and if that change is large enough, it signals that the reaction has begun or ended. Technically, this system uses the rate of change in linear patterns – the second derivative.

How to use the program

Code is available at github.com/logangraham/enthalpy

To run the process on a .txt file from DSC data, you can use the command line:

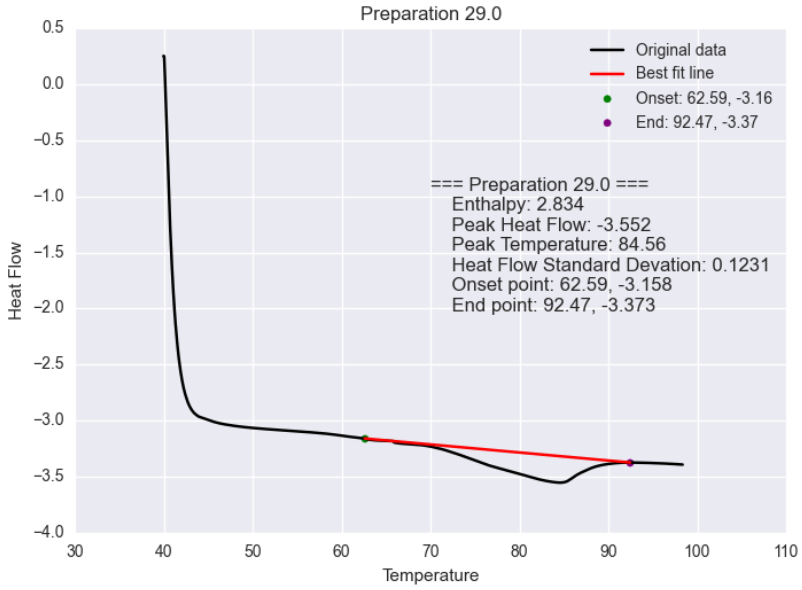
```
python main.py your_filename.txt
```

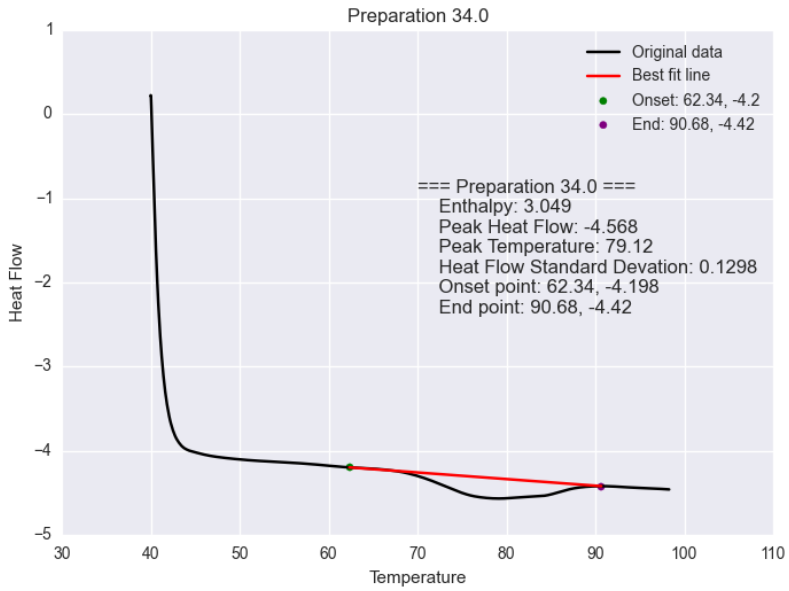
For more granular control, you can import any of the files below to a Python IDE.

There are four main files: 1. `main.py`: the main file you need; a single function that will run everything for you. 2. `file_helpers.py`: functions that clean and locate data. 3. `peak_detection.py`: functions that detect the peak and end points. 4. `peak_statistics.py`: functions that calculate statistics of the reaction.

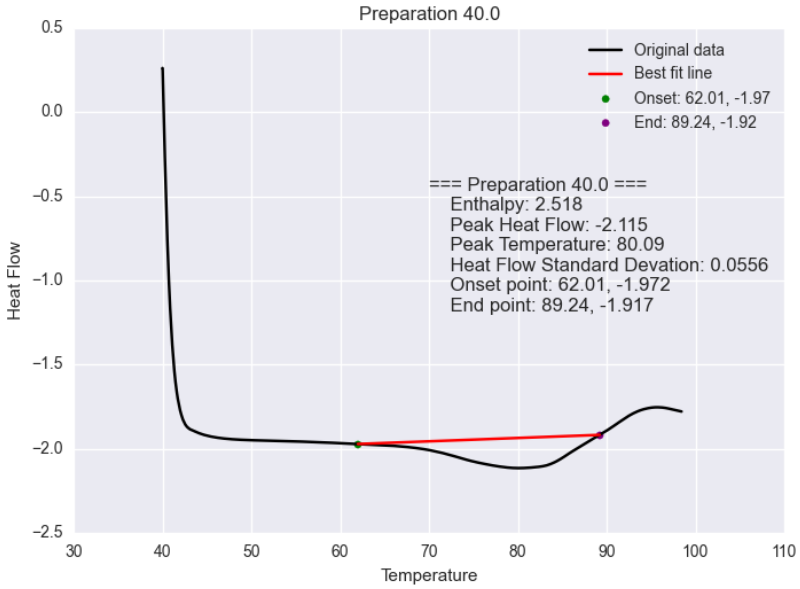
Results

Generally, the model works very well.

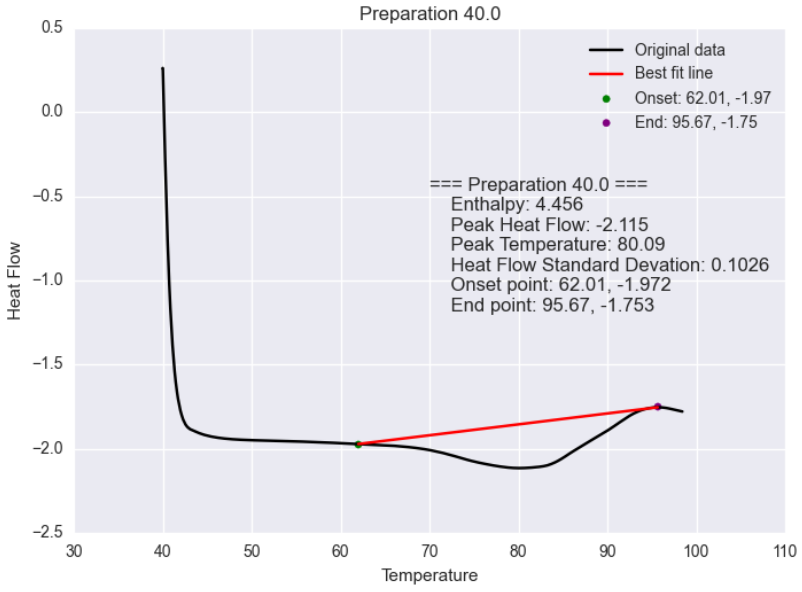




However, in some odd reactions, the system will misclassify the end point. In these cases, it might be more advantages to just take the point at which heat flow is maximized post-peak. I've built that in as the `use_alternative` parameter in `peak_detection.find_onset_point()` if you'd like to do that. For example, it

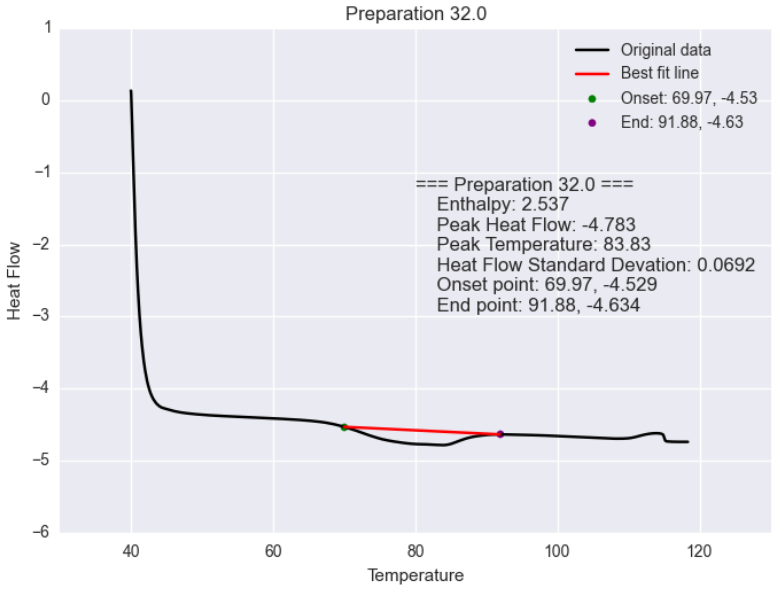


fixes preparation 40:



To this

However, the original methodology is much better in cases like this:



Advantages of this model

- This model saves time for a researcher.
- This model can spot points in the reaction that are more appropriate beginning and end points than the researcher might evaluate.

Limitations

- Note – some of the files are formatted badly. You'll need to treat them specially, or write your own clean_file() function for these files.
- The model is only reliable for normal, endothermic reactions, and should be tested on:
- Exothermic reactions
- Crystallization reactions
- Cases where the preparation explodes
- Reactions with impurities
- The model calculates enthalpy accurately, but start and end points are not always close to where a human would set them.
- Should probably be accompanied by a "human factor" when getting set up. That is, a human should be there to make sure nothing is going wrong for the first few reactions.
- Is untested on DSC protocol changes.

Data

I used 34 sets of DSC data based on 23 different preparations. Each dataset was generated from standard DSC procedure, where a preparation undergoes a thermal reaction and is heated from 40 to 100 degrees celsius at 5 degrees a minute.

The DSC device measures about 5 data points per second, resulting in 3552 observations. The device records the time, temperature of the preparation, heat flow of the preparation, and purge flow of the preparation. I used only temperature and heat flow.

How it works

Conceptually, the program determines approximate onset and end points of a reaction, draws a line between the two points, the integrates the difference between the line and the reaction data. This integration is the enthalpy of the reaction.

The onset and end points can be thought of as the points just before and after the reaction. Before and after the points, the data is linear; there is a constant heat flow. However, at these points, the heat flows starts to seriously diverge from the linear trend. This is where the reaction is thought to begin (and end.)

The problem then amounts to finding the points of significant deviation from the linear trend. I accomplish this using higher order derivatives.

Finding the Onset Point

1. First, the program approximates the data using a fifth-order spline. For normal DSC data, this approximation is good.
2. Second, the program finds the second derivative of the approximation. The second derivative is zero when the line slope is constant. However, under the approximation, the second derivative is minimized just before the reaction begins.
3. Third, the program chooses the point at which the second derivative is minimized.

Finding the End Point

1. First, the program approximates the data after the peak point using a fifth-order spline.
2. Second, the program finds the second derivative of the approximation.
3. Then, the program takes the absolute value of the second derivative, to find the distances of each point from zero.

4. Fourth, the program finds the last point for which the second derivative is close to zero. This can be thought of as the "linear" part of the data after the reaction, which is after the heat flow has stabilized.
5. Last, if this point is lower than the maximum post-peak point, the maximum post-peak point is chosen. Else, this point is chosen.

Things I would do with more time

- Try to classify double-peak reactions – *Gosia mentioned this would be valuable*
- Experiment with these approaches:
- Peak Detection in Chemometrics
- Peak Detection in MATLAB
- Make the code object-oriented, which is a little easier to read
- Give you the ability to analyze exothermic reactions with just one more input parameter
- Try a more accurate "peak point" method: fit a linear line like the one, subtract the line so you're left with a non-slanted difference, then find the peak point.

Acknowledgments

- Meng for teaching me so much in a very little time
- Gosia for her expert help to conceptualize the problem and an ideal solution Vivian for giving me first-hand access to the experiment
- Lee & Meng for coordinating this project
- And the rest of the Hampton Creek crew for letting me help out for a week and a bit!