Justin Lo (lab partner: Andrzej Wojcieszynski)

October 5, 2006

20.309 Module I: DNA Melting (Hybridization) Curves

Introduction and Setup

The setup was identical to the one presented in the lab manual with only a few exceptions. Features and numbers of interest are listed below for ease of repeating the experiment under identical conditions.

1. The two constant resistors in the Wheatstone bridge were selected to be optimized for sensitivity to the measured resistance, R_x, in the range of 1 kΩ to 10 kΩ. They thus have the value that most closely approximates $\sqrt{10}$ kΩ, which is 3.3 kΩ. The potentiometer was set such that Vab across the Wheatstone bridge's center would be 0 when the thermistor was at room temperature. This resistance was determined to be about 1094 Ω. This measurement is discussed in greater detail in the Calculations section.
2. The inverting op-amp used resistances of R_1 = 100 kΩ, R_2 = 100 Ω, and R_3 = 100 kΩ for a net gain of approximately 10^8 V/A.
3. A capacitor of 0.1 μF was placed in parallel with R_1 in the op-amp in order to serve as a low-pass filter.
4. The LED's voltage was (inadvertently) set to 15 V initially, and this was kept constant throughout so as not to incur any changes that might come with a sudden downgrade of intensity.

Data and Calculations

I. Creating the F vs. T plots.

We collected two sets of data for each DNA sample: the first set consisted of fluorescence data, measured as volts from the photodetector, plotted against time; the second consisted of Wheatstone bridge data (the voltage across the center of the bridge, V_{ab}) plotted against time.

In order to construct fluorescence vs. temperature plots, the measured resistances from the Wheatstone bridge had to be converted to temperatures. Because of limited sensitivity of the DAQ (5 mV), the measured resistance data resembled a step function. However, the actual temperature would be expected to fall slowly according to the Newtonian Law of Cooling. Thus, in order to avoid the step-function appearance (which would preempt any attempts at calculating the curve's derivatives from the data), we took a moving window average of the Wheatstone bridge data using the following MATLAB function:

```
function avg = windowAverage(data,number)
% data: data in the form of an array
% number: number of points to average into a single point
at a time
sizeV = size(data);
avg = zeros(sizeV(1)-(number-1),1);
start = floor(number/2);
for m = start+1:(sizeV(1)-(number-1)+start)
    avg(m-start) = sum(data((m-start):(m+number-1-
start)))./number;
end
```

After some trial and error, it seemed that using a window size of 7 would yield sufficiently good data without clipping off too much of the range on the two temperature extremes (the method sacrifices the window size minus one data points).

Next, we converted the smoothened voltage data into temperature data. The Wheatstone bridge consisted of one branch with a 3.3 kΩ resistor and a potentiometer in series and one branch with a 3.3 kΩ resistor and the thermistor in series. It was originally calibrated such that, when hooked up to the thermistor at room temperature (25ºC), the voltage across the bridge would be 0 V.

The RTD thermistor in the heating block had a resistance-temperature relationship of

$$R = 1000\ \Omega + 3.75 \cdot T, \text{ where T is the temperature in ºC}$$

So, the resistance of the thermistor at room temperature would be approximately 1094 Ω. Since R1/R3 = R2/RX, the potentiometer was expected to also have that value of resistance. A measurement of the potentiometer after calibration returned approximately 1100 Ω, indicating that the behavior of the bridge was as expected.

V_{ab} across the unbalanced bridge may be calculated, using two divider expressions, to be

$$V_{ab} = V_{\max}\left(\frac{R_x}{R_2 + R_x} + \frac{R_3}{R_3 + R_1}\right), \qquad (1)$$

which can be solved for Rx by plugging in $V_{max} = 5$ V, $R_3 = 1094\ \Omega$, and $R_1 = R_2 = 3.3$ kΩ and rearranging the terms (we used a TI-89 to save time). It is then easy to convert the Rx data to usable temperature data.

$$R_x = -3300 \cdot \left(\frac{V_{ab} - 1.2446657185}{V_{ab} + 3.755334282} \right), \text{ and } T = \frac{1000 - R_x}{3.75} \qquad (2)$$

The resulting F vs. T graph showed a normal melting curve shape, but it continued to have a jagged appearance due to having multiple data points at a single temperature. To eliminate this vertical aspect to the data, we ran the data through one last averaging function, this time a "vertical" averager that would take the points with a common T value and replace them with a single data point. Although this could have been done from the start to produce a very smooth graph, it would have drastically diminished the number of data points available, again leaving the derivative out of reach. This function is below, taking in a matrix with rows sorted by the T-data.

```
function avg = verticalAverager(xData,yData);
sizeX = size(xData);
startIndex = 1;
finalData = [];
for i=1:sizeX(1)-1
    if(xData(i) ~= xData(i+1))
        endIndex = i;
        dataPoint = mean(yData(startIndex:endIndex));
        finalData = [finalData; xData(i) dataPoint];
        startIndex = endIndex+1;
    end
end
avg = finalData;
```

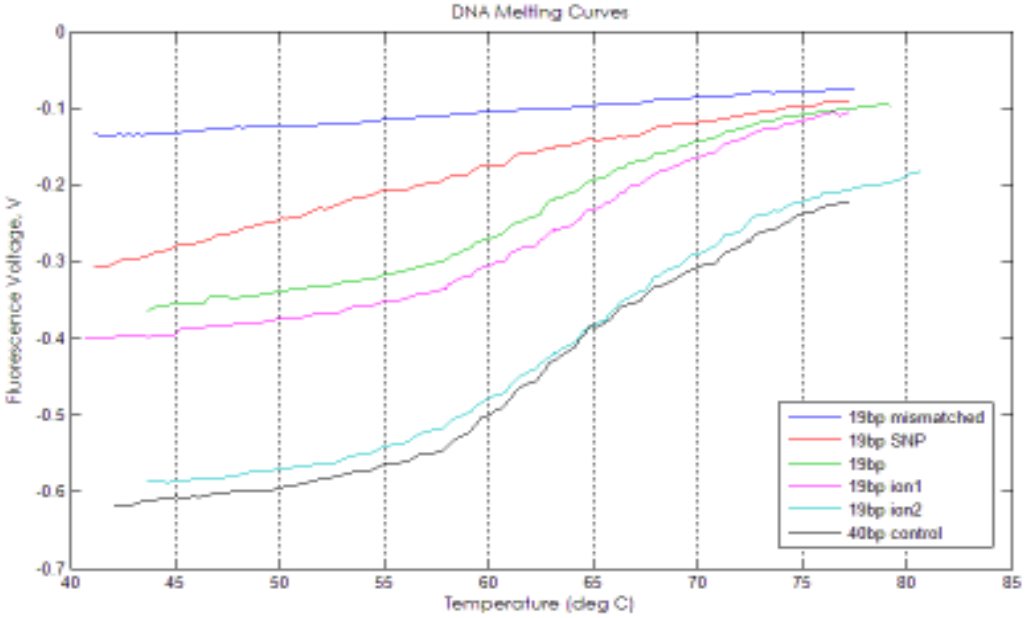


Fig. 1: *Melting curves for the six samples.* The graphs look upside-down (ie, greater signal is more negative) because of the inverting op-amp.

Normalizing the curves proves useful in estimating the values of T_m, at least in a relative sense, among the five samples that seem to have a T_m (the complete mismatch DNA shows minimal hybridization, and while there is a point at which 50% of the maximum hybridization is achieved, there is never a point at which 50% of the single strands are hybridized).

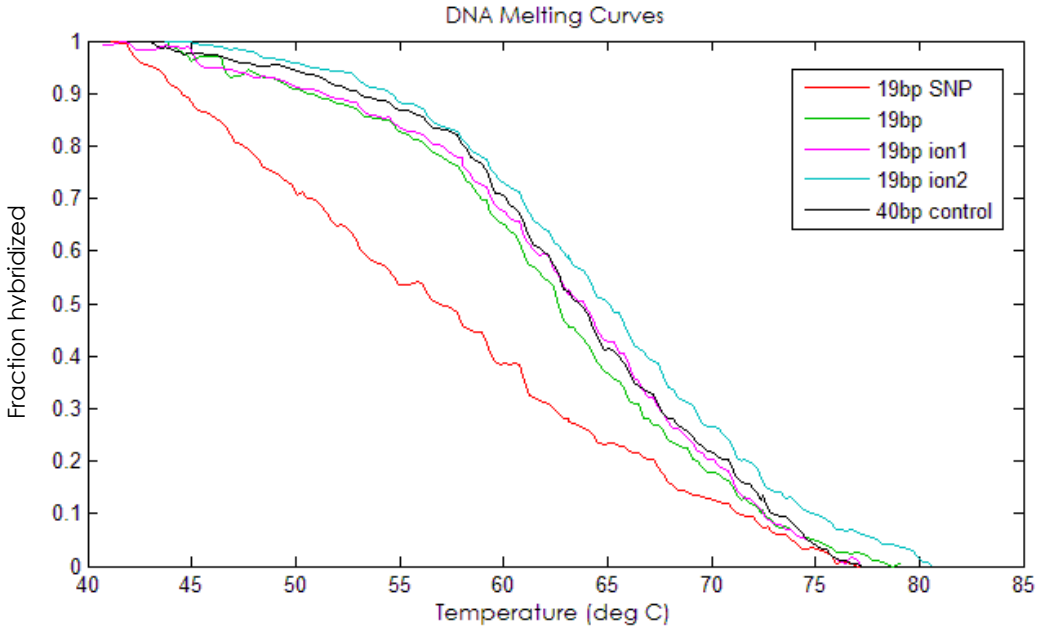


Fig. 2: *Melting curves using normalized data.* From this graph, we can ascertain that the relative order of the T_ms is probably 19bp mismatch < 19 bp < 40 bp ≈ 19 bp I_1 < 19 bp I_2.

II. Taking the Derivatives:

Taking the derivative of these functions without first fitting a model required that not all the data be used. Rather than taking the difference between every two adjacent data points and dividing by the temperature interval, dF/dT was computed by taking every four data points and computing the derivative between those points. This helped eliminate the phenomenon of infinite-valued differentials that would result from the rare case of the data moving vertically.

The T_m values were determined as being the highest point in the first derivative plot. In order to reduce bias from a single version of the derivative plot, we sampled the Tms with various "skipping intervals" and with different window averaging sizes and then took the average.

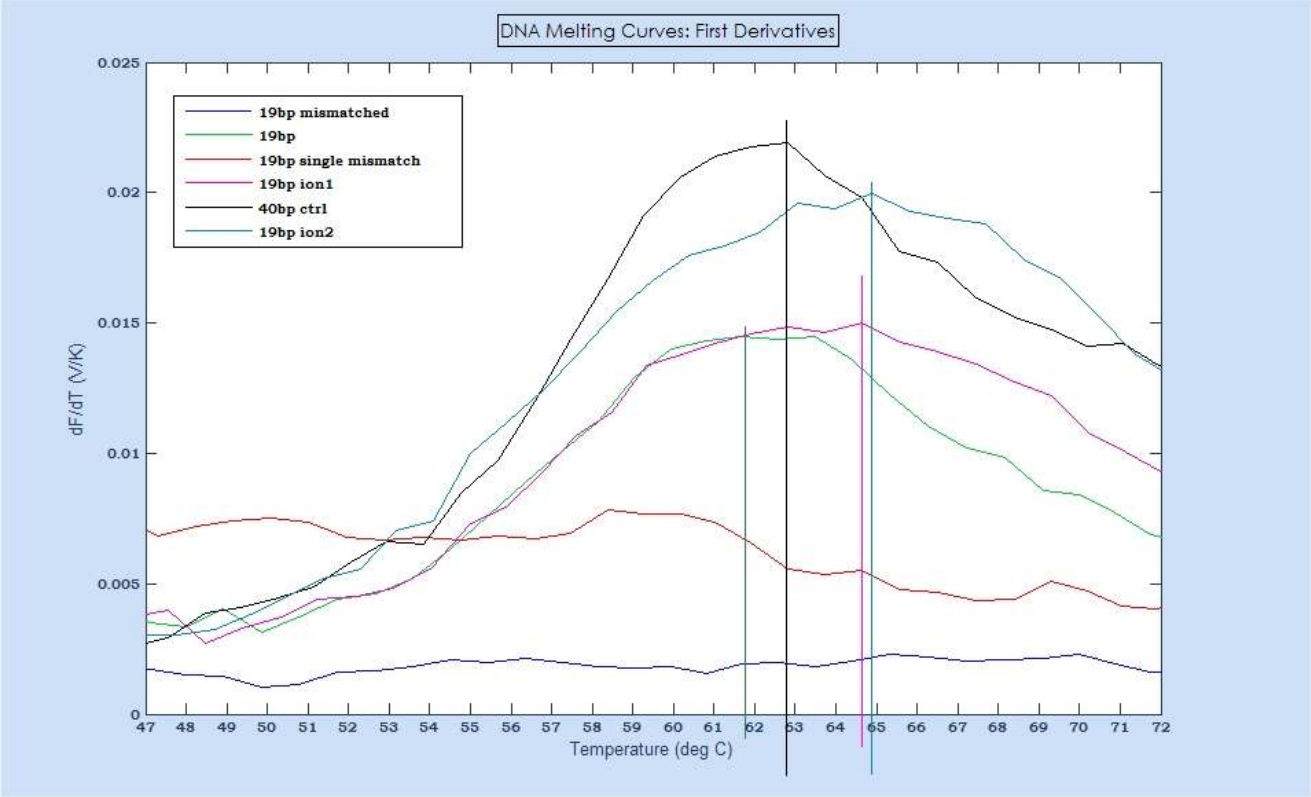


Fig 3: *dF/dT plot from the melting curves as presented in Fig. 1.* The y-axis has units of Volts/ºC. The vertical bars indicate approximate melting points on the various curves.

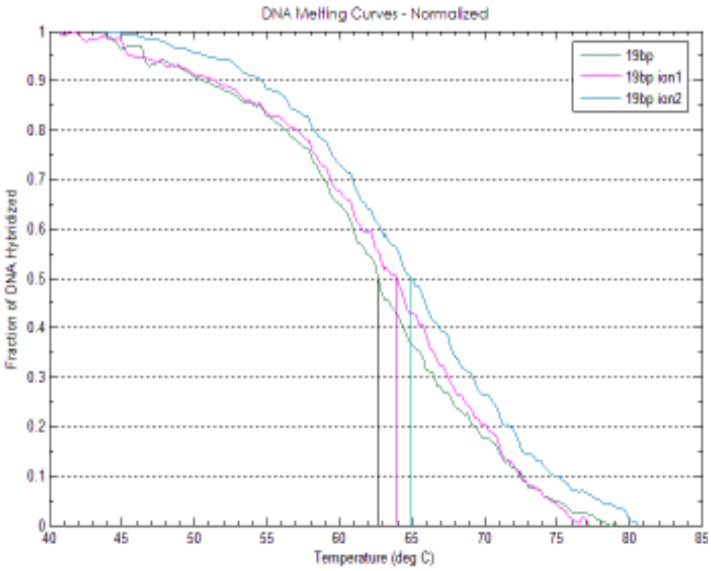
Analysis and Conclusions.

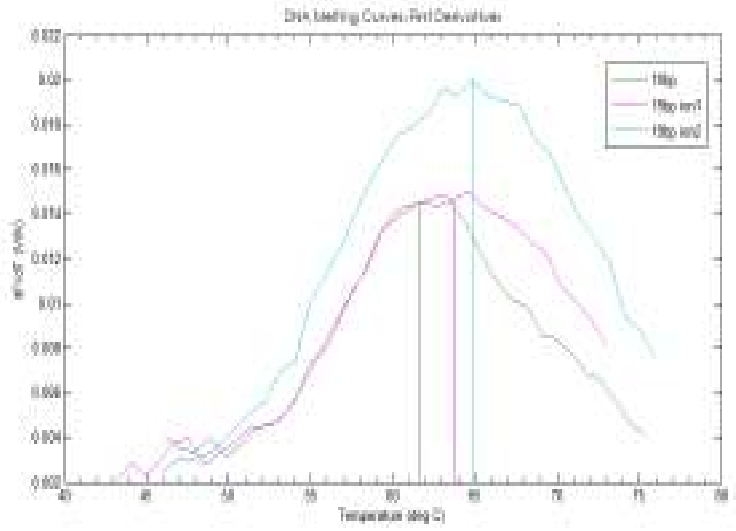
I. Ionic Strength

Increases in the ionic strength of the solution had the effect of shifting the melting curves to the right, having very little effect on the steepness of the hybridization but increasing T_m. The T_ms were measured in two different ways, yielding consistent values. First, the Tm was obtained from the normalized DNA melting curves (with all curves adjusted so that they range from 1, or completely dsDNA, to 0, or completely ssDNA). The Tm was defined as the point at which 50% of the DNA had hybridized. Second, the T_m was obtained from the derivative of the non-normalized DNA melting curves. Here, the T_m was defined as the inflection point – where the first derivative hits its extremum.

Ion Concentration	**T_m from Fractions**	**T_m from Derivative***
Low (default)	62.5ºC	61.5ºC
Medium (1)	64ºC	63-64ºC
High (2)	65ºC	65ºC

* To control for bias that might be introduced by any averaging method, we took the additional precaution of testing this method with different window average sizes and derivative spacing sizes. The overall averages after this testing were 61.9ºC, 64.1ºC, 65.0ºC, in the order of increasing ionic strength, with 14 different derivative plots analyzed.





Regardless of the method used to determine T_m, the trend is clear: increasing the ionic concentration increases the T_m. This observation can be explained through the association of positive ions such as sodium or potassium with the negatively-charged phosphate backbone of DNA. Cations probably stabilize the duplex structure by creating an entropically favorable polar-association "sheath."

Papers that deal with the melting temperature of DNA in relation to ionic concentration show the same trend. For instance, see Figure 5 in *Direct measurement of the melting temperature of supported DNA by electrochemical method* by Rita Meunier-Prest et al. (Nucleic Acids Res. 2003 December 1; 31(23): e150). The increasing sodium chloride concentration has a profound effect in shifting the curve to the right.

II. Mismatches

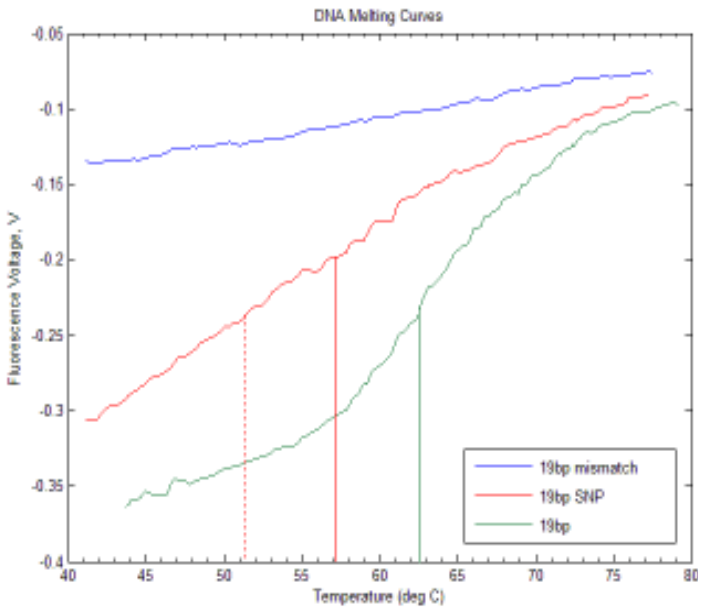
Mismatches have the effect of reducing the T_m and stretching out the transitional portion of the melting curve. In this sense, the mismatches reduce the effective base pair length of the DNA. Shorter strands of well-matched DNA also show the same properties: lower T_m and wider transitions. The lower Tm makes sense because there are less base-pairing associations to break, and the wider transition reflects the fact that the probability of a few bases dissociating at the same time is very possible while the probability of a

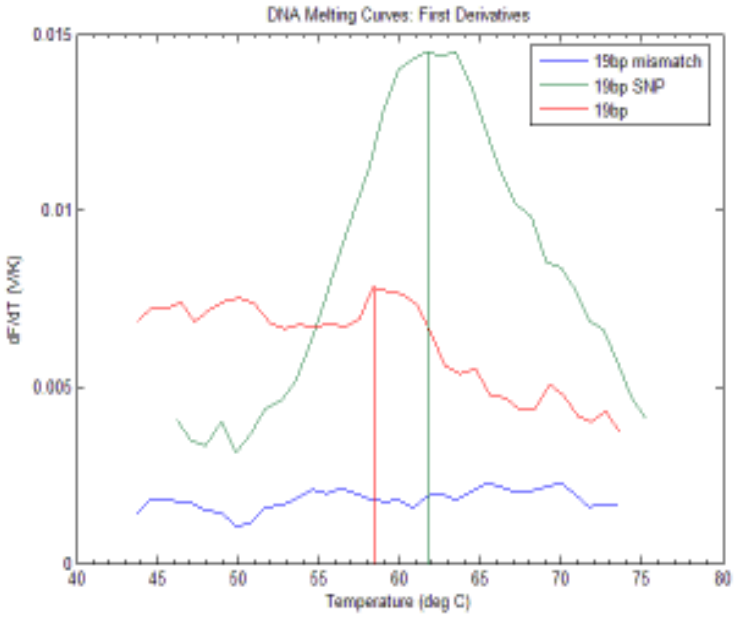
great deal of bases dissociating at the same time is much more difficult. Thus, longer strands of DNA have a much sharper transition, which occurs when there is just enough thermal energy to break all the hydrogen bonds.

Unfortunately, the "normalization" method that yields fraction data on the y-axis does not help in this situation: it is clear from the melting curves that only the perfectly matched sequence was able to complete hybridization by 40ºC, and so it is not possible to determine, except through extrapolation, what level of fluorescence would constitute the maximum hybridization for the mismatched samples. We determined Tm values from inspection of the original F vs. T plots and from the first derivative plot. Due to the near-linearity of the SNP and mismatch samples, however, the inflection point was very hard to detect. In fact, there may not be a formal T_m for the completely mismatched sample at all: at most four or so of its bases have the opportunity to bind at the same time, and they are interrupted by regions of ill-fitting pairs; at most two are contiguous matches. This sample would probably never reach that 50% hybridization level demanded by the non-derivative definition of T_m.

Sample	**T_m from melting curves**	**T_m from first derivatives**
19 bp perfect match	62.5ºC	61.5ºC (as above)
19 bp single mismatch	57ºC†	58ºC
19 bp complete mismatch	N/A (< 40ºC)	N/A

† This value has some degree of subjectivity because of the persistent linearity below 60 degrees. The dotted line below, which yields 51.5ºC, is based on the assumption (which may be incorrect) that the fluorescence should be the same when full hybridization occurs. The solid line instead assumes that the melting is roughly complete by around 40ºC.





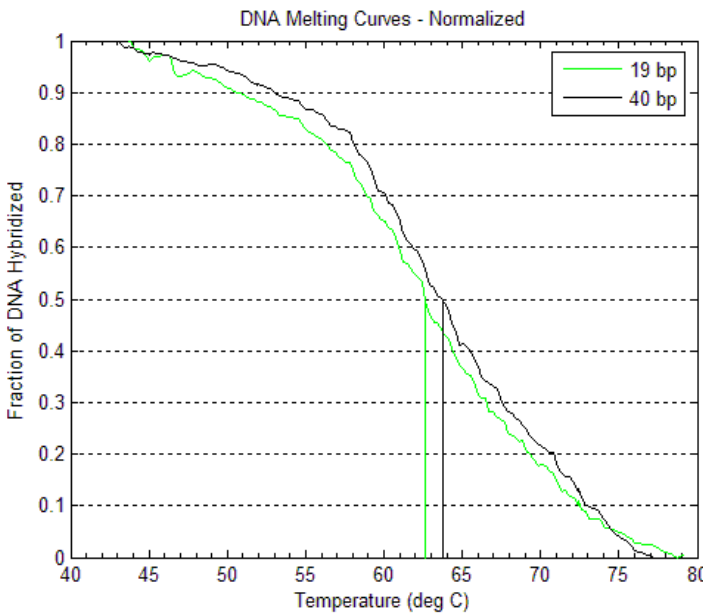
Using a T_m approximator for oligos, the T_m is expected to drop around 3 degrees when going form a 19-bp sequence to an 18-bp sequence (this was done using the tool at http://www.biophp.org/minitools/melting_temperature/demo.php). Indeed, the T_m drops here from 62.5ºC to about 57-58ºC with a single basepair mismatch. The effect of mismatches is essentially the same as reducing the number of effective bases, with the added factor of steric hindrance caused by the poor fits in between (especially purine-purine, such as the G-G bulge at position 14). It may be possible to guess at the T_m by using the number of matched bases and then subtracting some penalty for the mismatches in between.

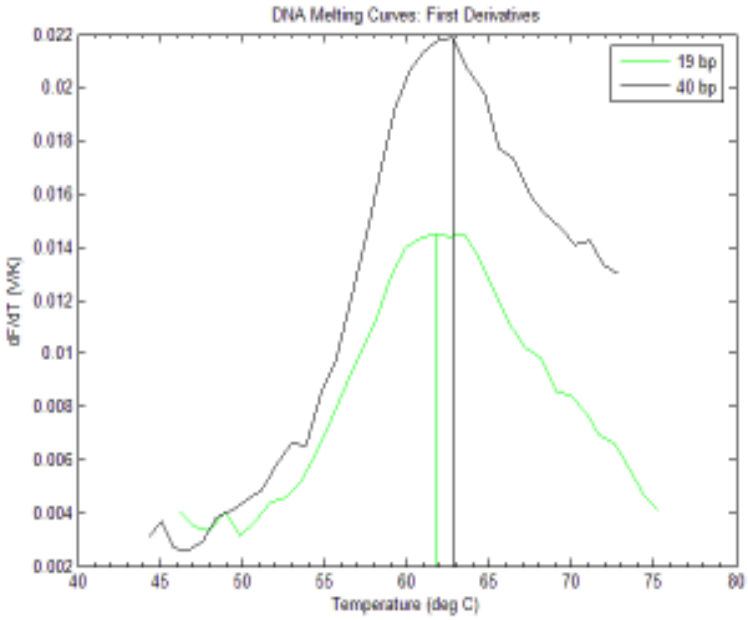
On a final note on these mismatch samples, we identified A as being the complete mismatch, B as the perfect match, and C as the SNP. They correspond to the blue, green, and red lines in the above graphs, respectively.

III. Length

The effect of DNA length on the melting curves is rather ambiguous from the data. Although it makes sense, especially based on the principles explained above, for length to play a large role in increasing the T_m and broadening the curve, the observed difference in this experiment was less than one degree accompanied by only the slightest of broadening.

Sample	T_m from Fractions	T_m from Derivatives
19 bp DNA	62.5ºC	61.5ºC
40 bp DNA	64ºC	63ºC





As can be seen in the above left graph, the 19 bp DNA begins to melt earlier but fully melts later than the 40 bp DNA. This is indicative of a broadening of the curve, although the difference is not as pronounced as might be expected given a two-fold increase in the DNA length. The reason for this phenomenon may be due to low salt concentrations, as described in section V.

IV. Comparisons to Peers

Our T_m values seem to be typical among the groups with which we compared data. Although the actual fluorescence values varied greatly due to different setups, the derivative graphs seem highly regular. Groups that modeled their data before taking the derivatives probably had an easier time picking out T_m values for the mismatched samples.

Variation in T_m values themselves could arise from either the apparatus or the data analysis method. Choosing different values for the resistors in the Wheatstone bridge would change the sensitivity of the bridge to the thermistor. Due to the 5mV minimum change detected by the DAT, groups with greater sensitivity to changes in R_x would have the most reliable temperature data. Additionally, the LED brightness would influence the magnitude of the fluorescence readings; brighter LEDs would make the

melting curve's hybridization region more pronounced in contrast to the surrounding noise that is at roughly a fixed level.

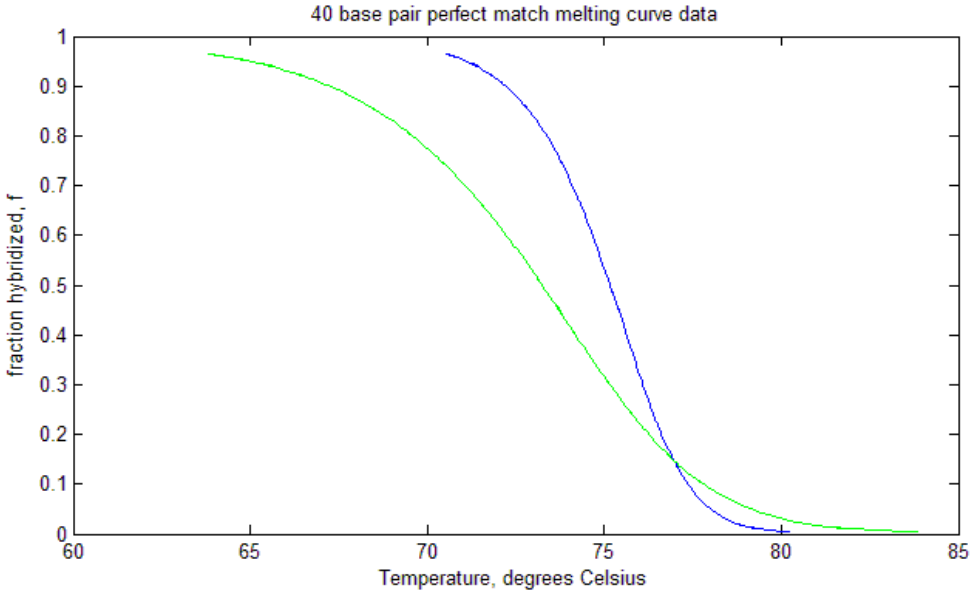
V. Models

The nearest-neighbor (NN) model is a simple method for approximating the enthalpy, entropy, and free energy values for DNA based on the interactions between immediately adjacent pairs of bases. There are ten such 5' → 3' combinations possible.

Each paper published on this model has presented wildly different values for the various parameters in question. For the purposes of estimating the expected ΔH and ΔS values, we have used the same source as presented in class, the 1998 SantaLucia paper that incorporates many experiments into one.

To compute the ΔH and ΔS values reliably, we created a small Python program (code at end) that would take in any 5' → 3' sequence and return the NN predictions of the thermodynamic values.

	ΔH° in kcal/mol	**ΔS° in cal/mol-K**	**ΔG° in kcal/mol**
40 bp perfect match	-308	-861	-51
19 bp perfect match	-149	-404	-28.6



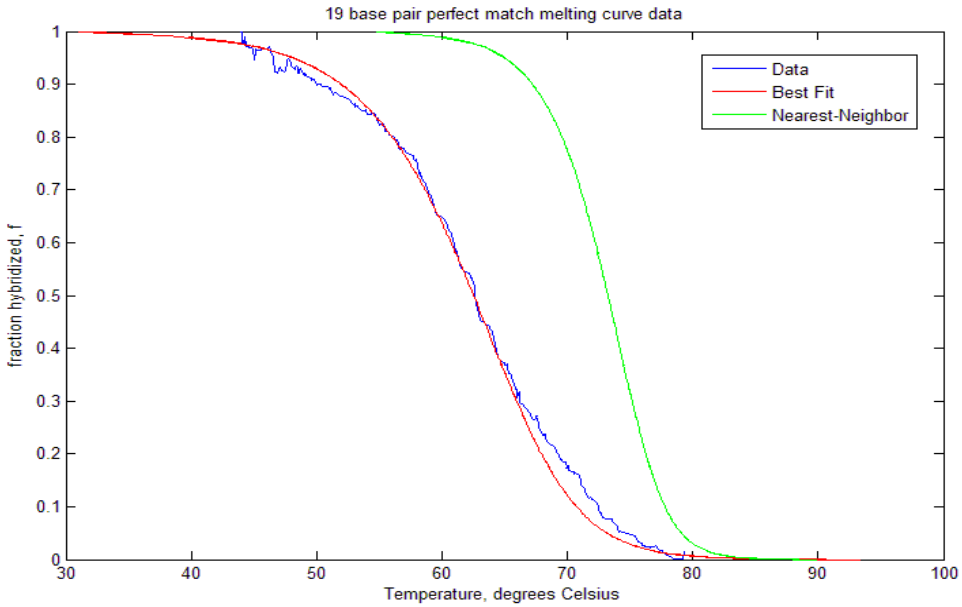
This model, when processed through the equation 3 shown below, predicts significantly higher T_m values than measured in the data: around 75ºC for the 40-bp strand and 73ºC for the 19-bp strand. It must be

noted that these predictions run contrary to T_m estimations that may be made using other methods; a calculator found at biophp.org for oligonucleotide sequences returns 63.5ºC for this particular 40-bp strand and 46.8ºC for the 19-bp strand. The T_m predictions will be revisited later in this section.

$$T(f) = \frac{\Delta H}{\Delta S - R \ln\left(\frac{2f}{C_T(1-f)^2}\right)}, \text{ where R = 1.987 cal/mol-K.} \qquad (3)$$

We fit our melting curve data to this equation 3 in order to obtain enthalpy and entropy values to compare the nearest-neighbor model to reality. However, it is also possible that equation 3 is not perfectly reliable.

Using lsqcurvefit on a function essentially the same as the one presented in the lab manual, we determined that, for the 19-bp strand, **ΔH = -77.9 kcal/mol** and **ΔS = -209 cal/mol-K,** giving **ΔG = -15.6 kcal/mol**. The value of R was corrected for these units, as 1.987 cal/mol-K was used in place of 8.314 J/mol-K. The melting curve's enthalpy and entropy values both have an absolute value drastically lower than those predicted by the nearest-neighbor model, which is ordinarily rather accurate. This discrepancy can be seen in the graph below.

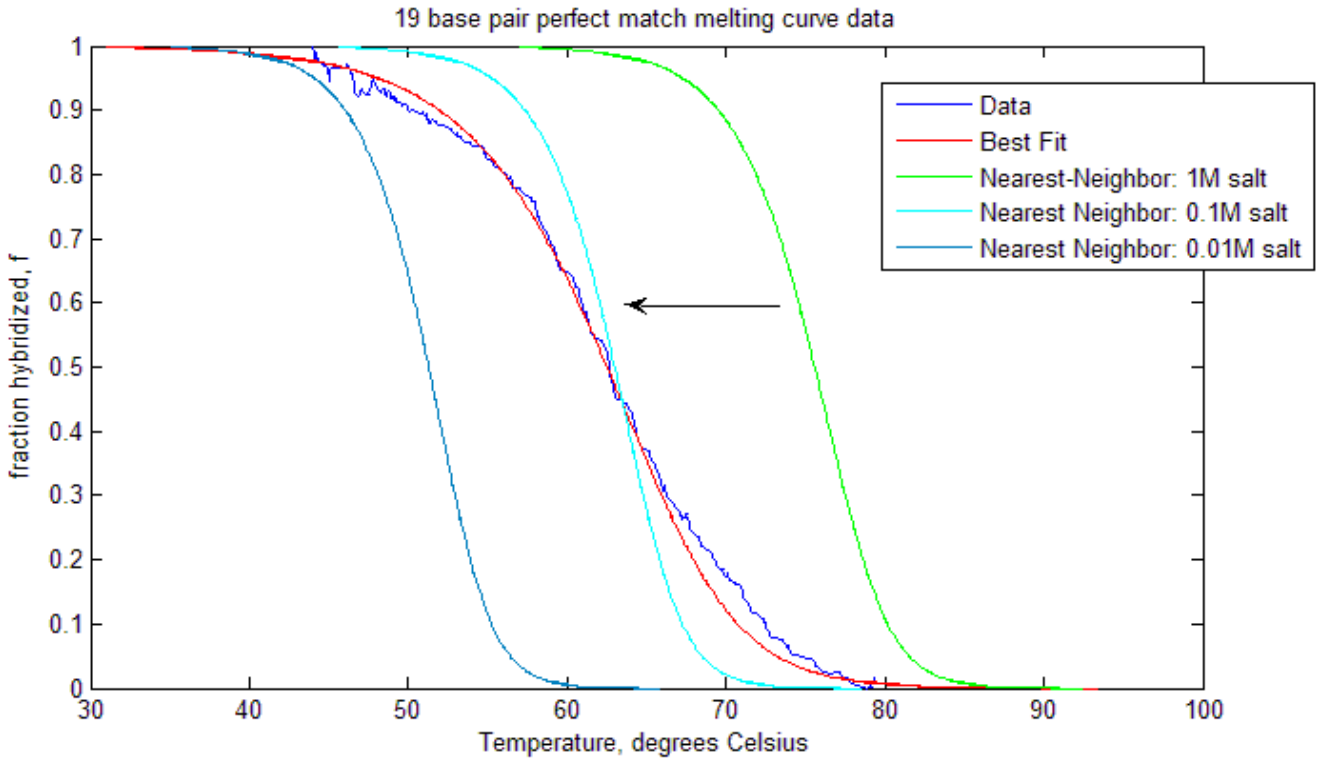


The best fit curve and the data have a lower T_m and a somewhat less steep curve. One possible source of the difference could be the ionic concentration of the sample as compared to the concentrations used to measure the nearest-neighbor model values, which were based on 1 M NaCl.

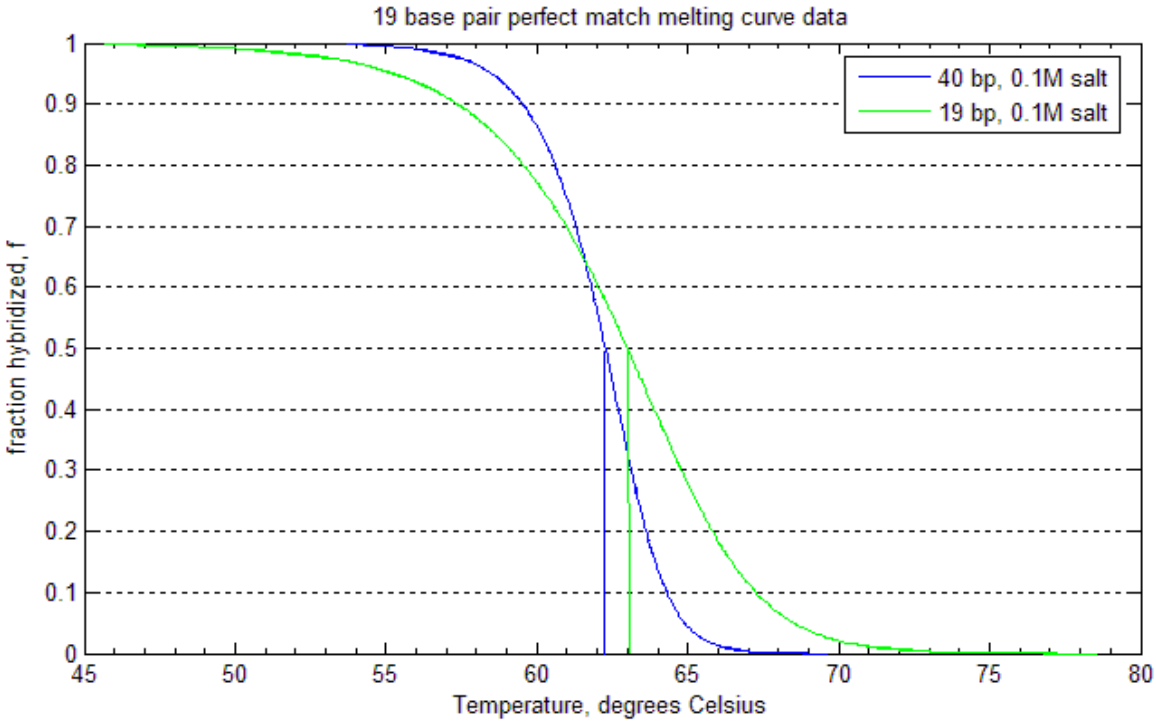
Let us consider the effect of ion concentration first on ΔG. SantaLucia proposes that ΔG be corrected for salt concentration by adding the term $-0.114 \cdot N \cdot \ln([Na^+])$. If the salt concentration were 0.1 M in our sample, then the correction will bring the N-N predicted value to –23.6 kcal/mol. A concentration of 0.01 M would bring the value of ΔG° to –18.6 kcal/mol. Since the 19 kb sample was prepared in 10 mM Tris buffer with no additional salts added, this last value seems most likely, and it also is very close to the measured value of –15.6 kcal/mol – at least, much moreso than the original prediction of –28.6 kcal/mol.

SantaLucia also provides a correction term for ΔS, assuming that ΔH is roughly invariant with salt concentration, which coincides with the previous assertion in the ionic concentrations section of this report that the stabilization of dsDNA by salt is primarily an entropic effect. The ΔS correction term is given as $0.368 \cdot N \cdot \ln([Na^+])$. At 0.1M, the ΔS predicted decreases to –420 cal/mol-K; at 0.01M, the ΔS predicted decreases further to –436 cal/mol-K.

Despite the fact that this seems to be adjusting in the "wrong direction," a plot suffices to show that in fact, the fit with these new values is much better than before.



In fact, the problem with the unusually high predicted melting temperatures has also been addressed.



Although the 0.1M salt concentration that places the curves right in the middle of the region observed through our measurements is probably higher than the actual concentration, the graph still begins to explain some of the peculiarities of the earlier data. For instance, the T_m values are in fact very close if not overlapping at this range of salt concentrations; the pronounced difference based on length is not present until higher concentrations around 1.0 M. In fact, the above graph shows that the T_m for the 19-bp DNA is actually higher, not lower, than that for the 40-bp DNA in this salt concentration region, which some other groups might have measured (while our T_m for the 40-bp DNA was just a hair higher).

There remains the large difference in ΔH and ΔS values that seem irreconcilable. Curves centered on a particular T_m can have vastly different ΔH and ΔS values, as long as they remain roughly in proportion. The fact is that the observed data forms a much flatter curve than would be expected, and that is probably influencing the best fit's selection of ΔH and ΔS. Admittedly, the accuracy of the ΔG figure is much more important, as it governs the actual melting reaction.

[All pages after this point are MATLAB and Python Code]

1. Python Code for Calculating the NN model enthalpy and entropy

```
## This function will take in a 5' to 3' sequence and return its nearest-
neighbor energy
## for the double-stranded form.  Only input one strand.

import math

def NNEnergy(seq):
    ## make sure seq is 5' to 3'
    enthalpy = 0;
    entropy = 0;
    salt = 0.01; ## The salt concentration in M
##    dict = {'AA':-1,'AT':-.88,'TA':-.58,'CA':-1.45,'TG':-1.45,'GT':-
1.44,'AC':-1.44,
##            'CT':-1.28,'AG':-1.28,'GA':-1.3,'TC':-1.3,'CG':-2.17,'GC':-
2.24,'GG':-1.84,'CC':-1.84,
##            'TT':-1} ## The Gibbs' free energy data, if you ever want to
use it
    HDict = {'AA':-7.9,'TT':-7.9,'AT':-7.2,'TA':-7.2,'CA':-8.5,'TG':-
8.5,'GT':-8.4,'AC':-.84,
            'CT':-7.8,'AG':-7.8,'GA':-8.2,'TC':-8.2,'CG':-10.6,'GC':-
9.8,'GG':-8.0,'CC':-8.0} ## kcal/mol
    SDict = {'AA':-22.2,'TT':-22.2,'AT':-20.4,'TA':-21.3,'CA':-22.7,'TG':-
22.7,'GT':-22.4,'AC':-22.4,
            'CT':-21.0,'AG':-21.0,'GA':-22.2,'TC':-22.2,'CG':-27.2,'GC':-
24.4,'GG':-19.9,'CC':-19.9} ## cal/mol-K
    for i in range(0,len(seq)-1):
        enthalpy += HDict[seq[i:i+2]];
        entropy += SDict[seq[i:i+2]];
    if(seq[len(seq)-1] == 'A' or seq[len(seq)-1] == 'T'): ## Initiation terms
        enthalpy += 0.1;
        entropy += -2.8;
    else:
        energy += 2.3;
        entropy += 4.1;
    if(seq[0] == 'A' or seq[0] == 'T'):
        enthalpy += 0.1;
        entropy += -2.8;
    else:
        energy += 2.3;
        entropy += 4.1;
    entropy = entropy + 0.368*len(seq)*math.log(salt);
    return [enthalpy,entropy]

print 'AGCAGCCATGCAAATGTTAAAAGAGACTATCAATGAGGAA'
print NNEnergy('AGCAGCCATGCAAATGTTAAAAGAGACTATCAATGAGGAA')

print 'ATCAAGCAGCCATGCAAAT'
print NNEnergy('ATCAAGCAGCCATGCAAAT')
```

2. MATLAB Parent Script for Generating Plots

```
% Module 1 - Andrzej W. and Justin L.
% September 29, 2006
% rev. October 2, 2006
% Modeling script for DNA melting data

sampleA = load('awjl_A.txt');
sampleB = load('awjl_B.txt');
sampleC = load('awjl_C.txt');
sampleI1 = load('awjl_I1.txt');
sampleI2 = load('awjl_I2.txt');
sample40 = load('awjl_40.txt');

% FitVals = lsqcurvefit(@awjlMelt, [1,1],

% The fluorescence readings
n=7;
d=4;
FA = windowAverage(sampleA(:,2),n);
FB = windowAverage(sampleB(:,2),n);
FC = windowAverage(sampleC(:,2),n);
FI1 = windowAverage(sampleI1(:,2),n);
FI2 = windowAverage(sampleI2(:,2),n);
F40 = windowAverage(sample40(:,2),n);

% Temperature calculations
TA = voltageToTemp(sampleA,n);
TB = voltageToTemp(sampleB,n);
TC = voltageToTemp(sampleC,n);
TI1 = voltageToTemp(sampleI1,n);
TI2 = voltageToTemp(sampleI2,n);
T40 = voltageToTemp(sample40,n);

A = sortrows([TA FA]);
B = sortrows([TB FB]);
C = sortrows([TC FC]);
I1 = sortrows([TI1 FI1]);
I2 = sortrows([TI2 FI2]);
S40 = sortrows([T40 F40]);


A = verticalAverager(A(:,1),A(:,2))
B = verticalAverager(B(:,1),B(:,2));
C = verticalAverager(C(:,1),C(:,2));
I1 = verticalAverager(I1(:,1),I1(:,2));
I2 = verticalAverager(I2(:,1),I2(:,2));
S40 = verticalAverager(S40(:,1),S40(:,2));

figure(1)
plot(A(:,1),A(:,2),'b-
',C(:,1),C(:,2),'r',B(:,1),B(:,2),'g',I1(:,1),I1(:,2),'m-
',I2(:,1),I2(:,2),'c-',S40(:,1),S40(:,2),'k-')
xlabel('Temperature (deg C)'); ylabel('Voltage, V');
legend('19bp mismatched','19bp SNP','19bp','19bp ion1','19bp ion2','40bp
control');
title('DNA Melting Curves');
```

```
figure(3) % derivatives graph
dFAdT = differential(A(:,1),A(:,2),d);
dFBdT = differential(B(:,1),B(:,2),d);
dFCdT = differential(C(:,1),C(:,2),d);
dFI1dT = differential(I1(:,1),I1(:,2),d);
dFI2dT = differential(I2(:,1),I2(:,2),d);
dF40dT = differential(S40(:,1),S40(:,2),d);

dFAdT = [windowAverage(dFAdT(:,1),n),windowAverage(dFAdT(:,2),n)];
dFBdT = [windowAverage(dFBdT(:,1),n),windowAverage(dFBdT(:,2),n)];
dFCdT = [windowAverage(dFCdT(:,1),n),windowAverage(dFCdT(:,2),n)];
dFI1dT = [windowAverage(dFI1dT(:,1),n),windowAverage(dFI1dT(:,2),n)];
dFI2dT = [windowAverage(dFI2dT(:,1),n),windowAverage(dFI2dT(:,2),n)];
dF40dT = [windowAverage(dF40dT(:,1),n),windowAverage(dF40dT(:,2),n)];
plot(dFAdT(:,1),dFAdT(:,2),'b-',dFBdT(:,1),dFBdT(:,2),'g-
',dFCdT(:,1),dFCdT(:,2),'r-',dFI1dT(:,1),dFI1dT(:,2),'m-
',dFI2dT(:,1),dFI2dT(:,2),'c-',dF40dT(:,1),dF40dT(:,2),'k-')
xlabel('Temperature (deg C)'); ylabel('dF/dT (V/K)');
legend('19bp mismatched','19bp','19bp single mismatch','19bp ion1','19bp
ion2','40bp ctrl');
title('DNA Melting Curves: First Derivatives'); set(gca,'XTick',40:1:80);
grid on
```

3. Conversion of Voltage to Temperature

```
function tempBR = voltageToTemp(rawData,number)
voltageSet = rawData(:,3); % resistance data
voltageSet = windowAverage(voltageSet,number);
Rx = -3.3.*(voltageSet - 1.2446657185)./(voltageSet + 3.755334282); % Convert
to real resistances
tempBR = (Rx.*1000-1000)./3.75; % convert to temperature
```

4. Taking the derivative

```
function dYdX = differential(xData,yData,n)
% This allows a non-evenly-spaced xData
a = size(yData);
counter = 1; % the counter that iterates
dYdX = []; % initialize the output matrix
while(counter <= a(1)-n)
    dX = xData(counter+n)-xData(counter);
    dY = yData(counter+n)-yData(counter);
    dYdX = [dYdX; xData(counter) dY/dX];
    counter = counter+n;
end
```

5. Converting data to fractions

```
function adjustedData = normalizeData(data)
maxY = max(data)
minY = min(data)
sizeData = size(data);
adjustedData = zeros(sizeData(1),1);
for i=1:sizeData(1)
```

```
    adjustedData(i,1) = (data(i) - minY)./(maxY - minY); % Converting
fluorescent voltage into a fraction
end
```

6. Modeling Section 6.2

```
a = xlsread('dataB.xls');
avg = verticalAverager(a(:,1),a(:,2));
maxY = max(a(:,2))
minY = min(a(:,2))
sizeData = size(avg);
adjustedData = zeros(sizeData(1),1);
for i=1:sizeData(1)
    adjustedData(i,1) = (avg(i,2) - minY)./(maxY - minY); % Converting
fluorescent voltage into a fraction
end
fractionData = 0.9999-adjustedData(:,1);
originalTemperatures = avg(:,1);

options = optimset('Display','iter')
FitVals = lsqcurvefit(@awjlMelt, [-144000 -404], fractionData,
originalTemperatures, [], [],options); % avg(:,2) is f.  avg(:,1) is T.
FitVals2 = [-308000 -895]
FitVals3 = [-149000 -420]

R = 1.987; % cal/mol-K
C_T = 33e-6; % M
Tmodel = awjlMelt(FitVals,fractionData);
Tmodel2 = awjlMelt(FitVals2,fractionData);
Tmodel3 = awjlMelt(FitVals3,fractionData);

plot(originalTemperatures,fractionData,'b-',Tmodel,fractionData,'r-')
xlabel('Temperature, degrees Celsius'); ylabel('fraction hybridized, f');
title('19 base pair perfect match melting curve data');
legend('Original Data','Best Fit')
% legend('Data','Best Fit','Nearest-Neighbor: 1M salt','Nearest Neighbor:
0.1M salt','Nearest Neighbor: 0.01M salt')
```

7. The function called by lsqcurvefit above

```
function Tf = awjlMelt(constants, f) % constants: enthalpy and entropy
% f: fraction of DNA melted.
R = 1.987; % L-atm/mol-K
C_T = 33e-6;
dH = constants(1); % enthalpy
dS = constants(2); % entropy
Tf = dH./(dS - R*log(2*f./(C_T*(1-f).^2)))-273.15;
```

There's more, but do you really want to see it …….