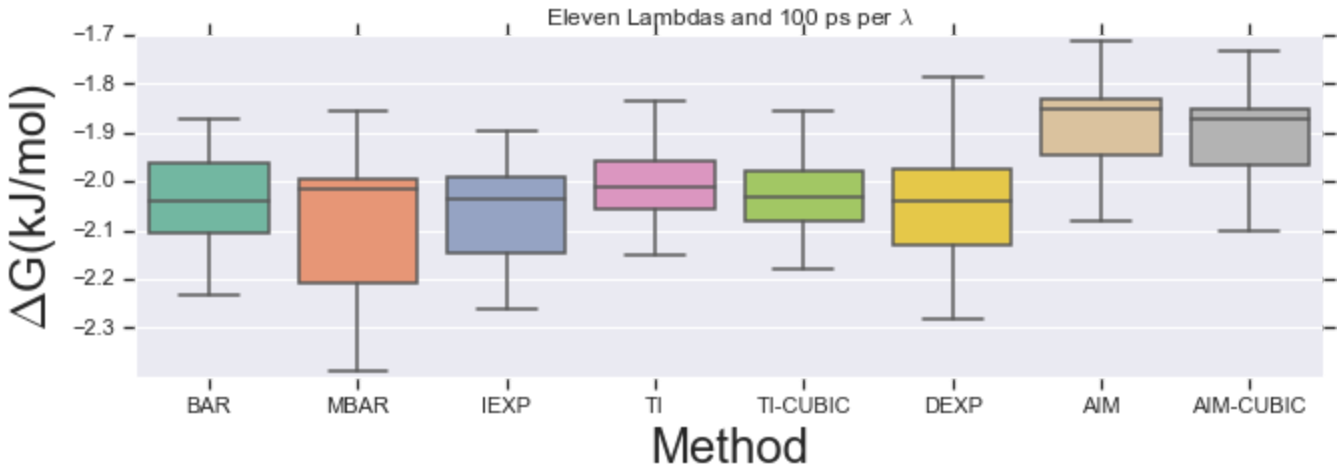
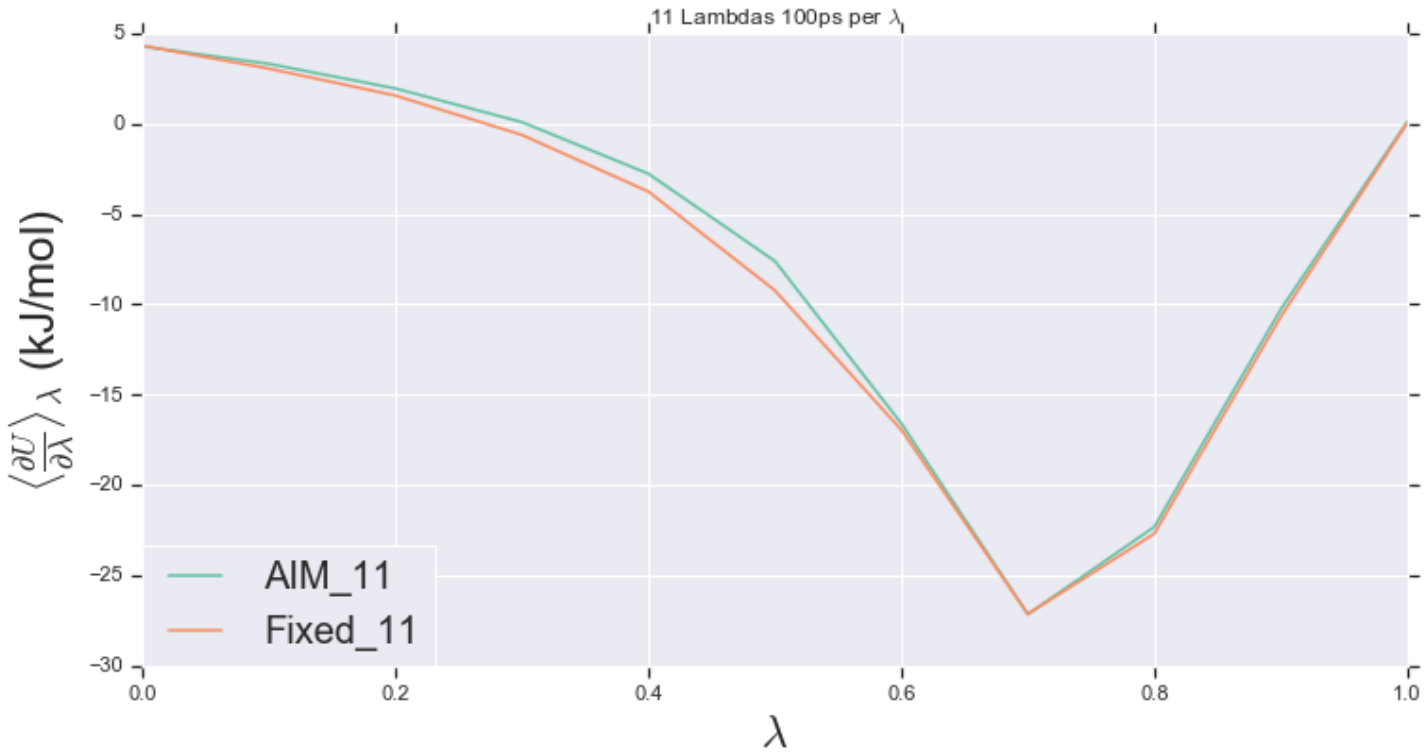
Weekly updates for Chris Mirabzadeh

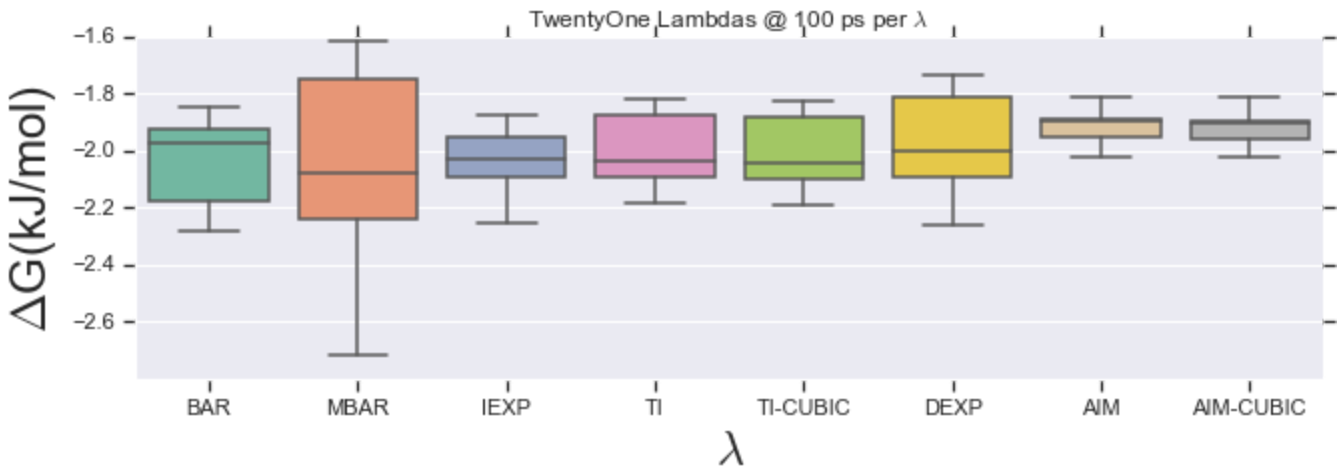
File Structure

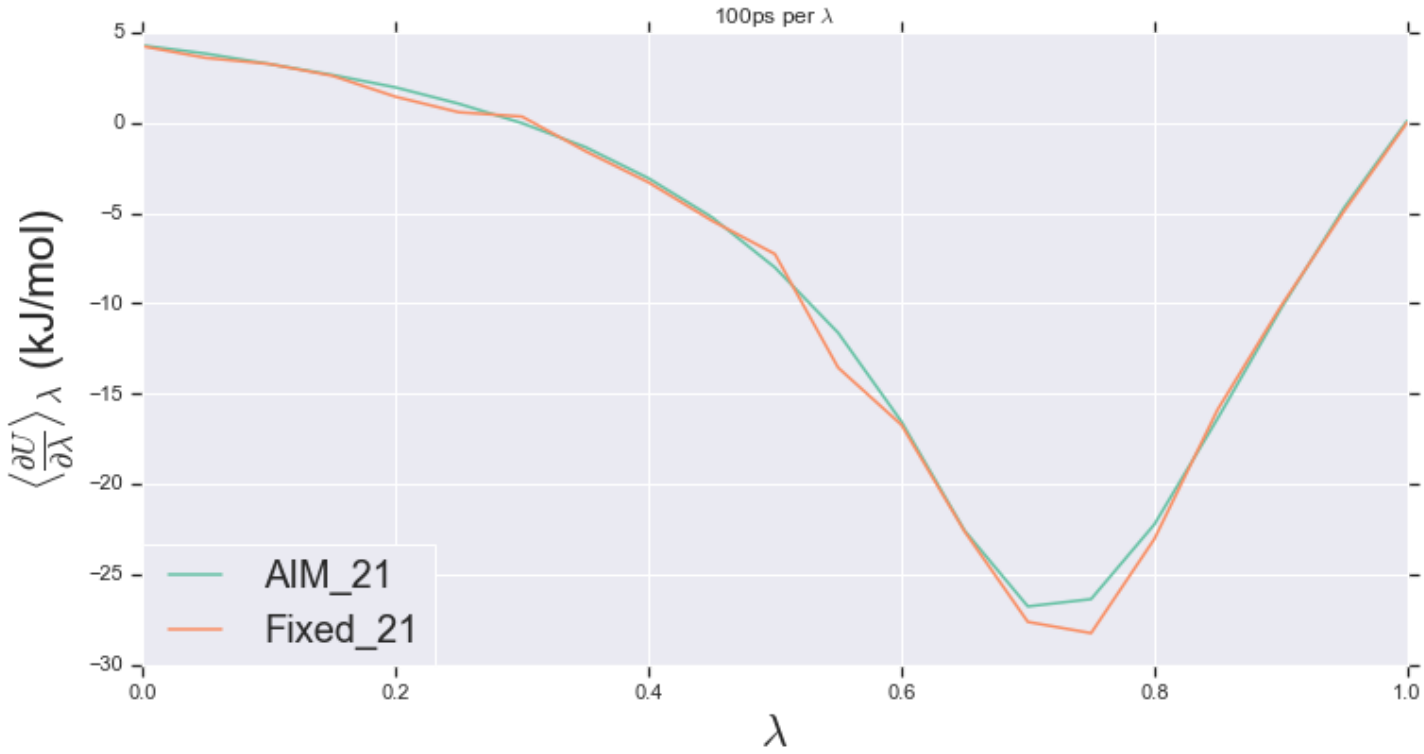
Date: XX/YY/ZZZZ
Current Goal(s): What are you doing?
Update: What have you done?
Current Problem: Where are you stuck?
Possible resolutions: What are you planning to do?

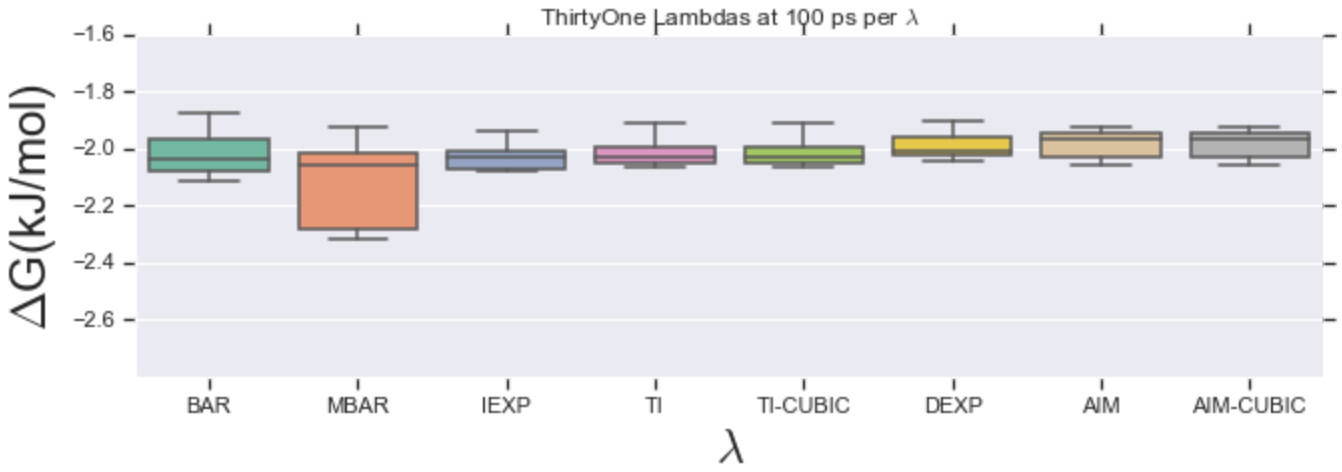
Date: 08/22/2018
Current Goal(s): What are you doing? Clarifying observations made
Update: What have you done? Graphed the components of the <du/dL> curve

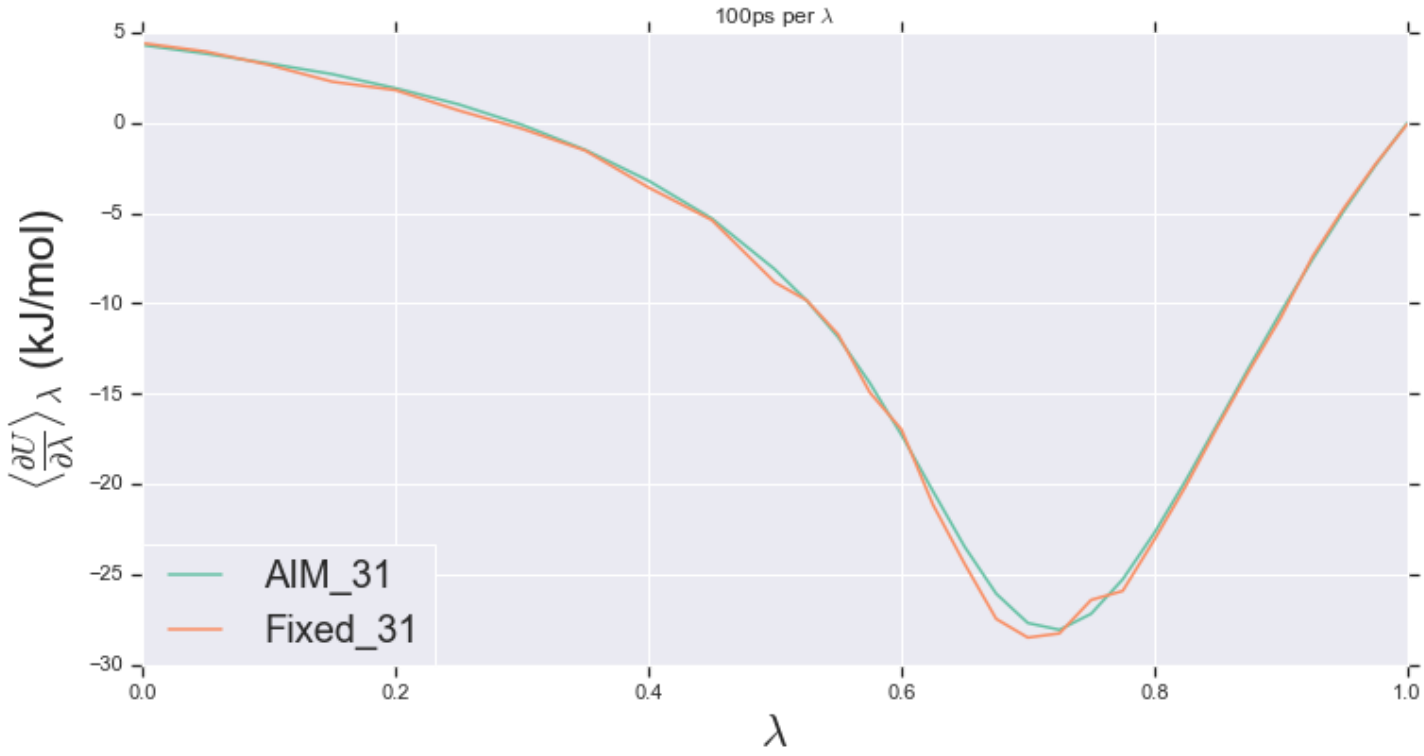





TwentyOne Lambdas @ 100 ps per λ
ΔG(kJ/mol)
−1.6
−1.8
−2.0
−2.2
−2.4
−2.6
BAR
MBAR
IEXP
TI
TI-CUBIC
DEXP
AIM
AIM-CUBIC
λ


100ps per λ
$\left\langle \frac{\partial U}{\partial \lambda} \right\rangle_\lambda$ (kJ/mol)
5
0
−5
−10
−15
−20
−25
−30
0.0
0.2
0.4
0.6
0.8
1.0
AIM_21
Fixed_21
λ





As the curve gets smoother the differences between cubic and non cubic almost vanish. The box plots show this better than the violin plots.

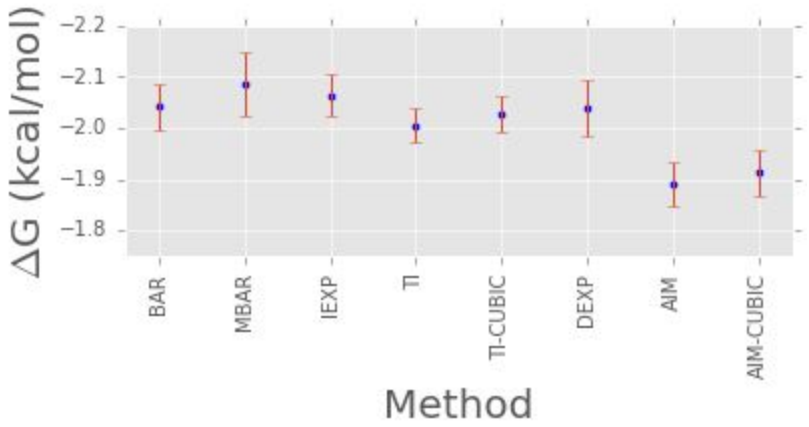
Current Problem: Where are you stuck?
Possible resolutions: What are you planning to do?

Date: 08/19/2018
Current Goal(s): What are you doing? Compiling data, writing and creating visualizations
Update: What have you done? I'm working on a better way to represent the time per N lambdas

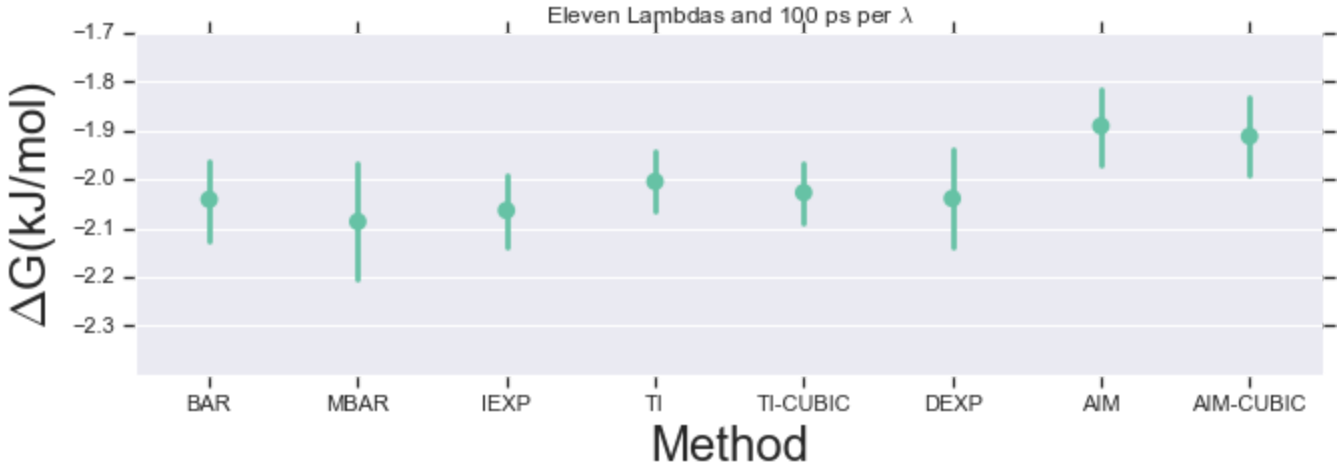
The question I was trying to answer with these plots was, what is the right lambda schedule to start with? I needed a lambda schedule at 100ps per lambda that was dense enough to allow convergence over longer time per lambda.

I have a few different graphs that I would like feedback for.

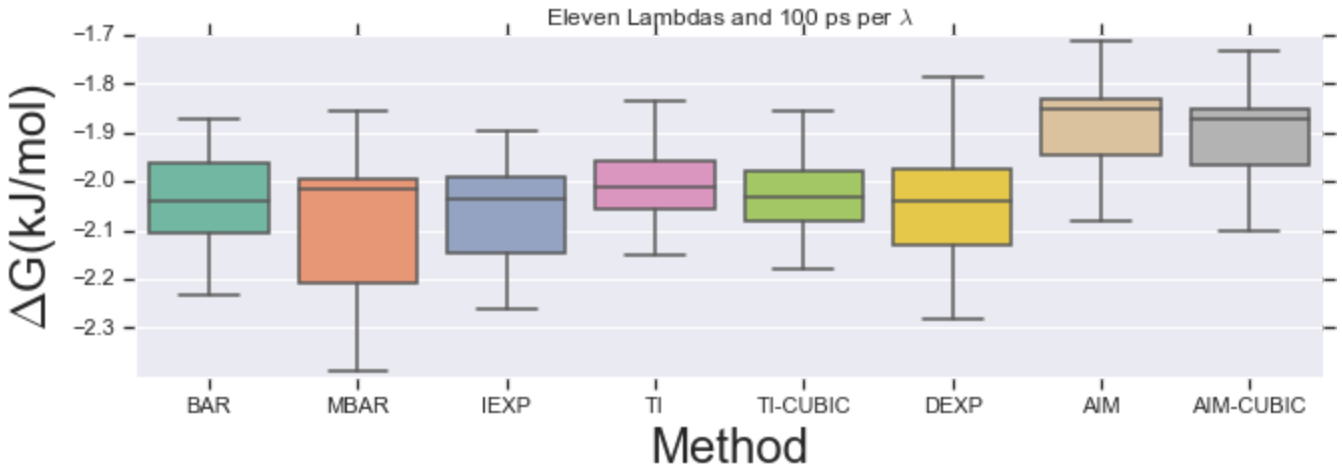
This is the original representation for 11 lambdas 100ps per lambda, 8 trials:



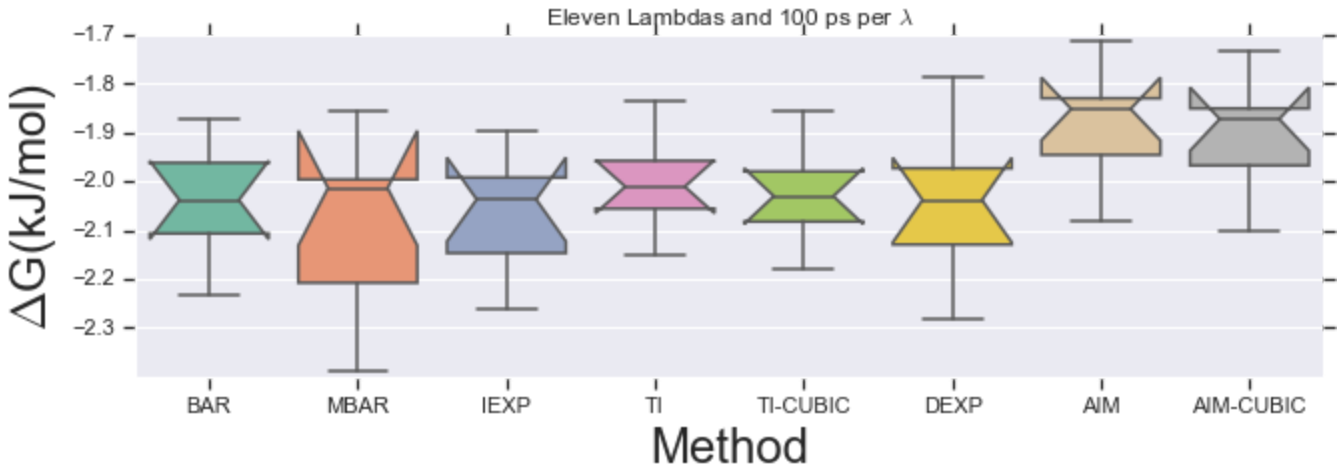
First, I made the plot easier to read.

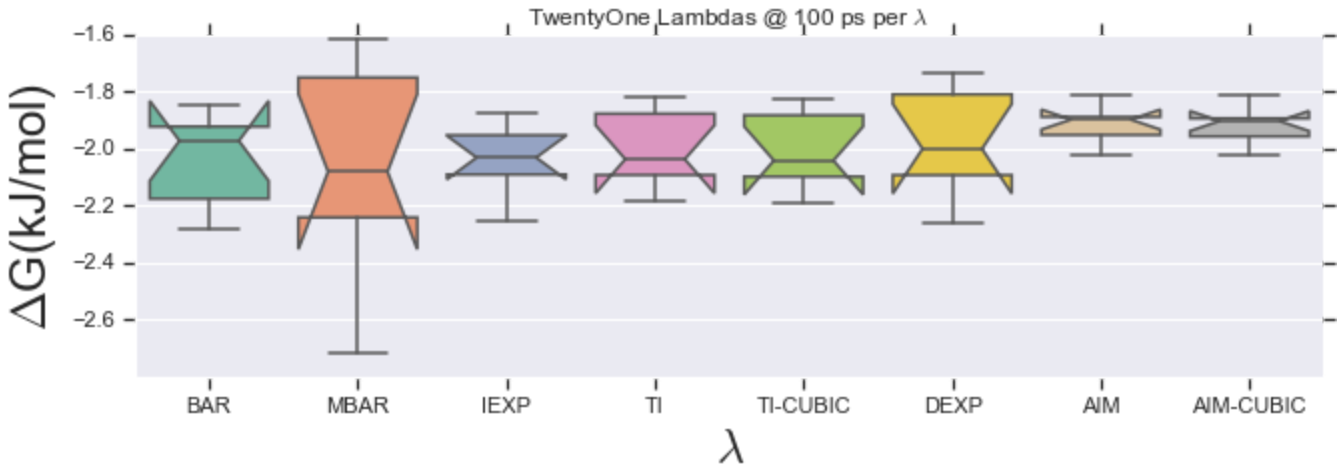


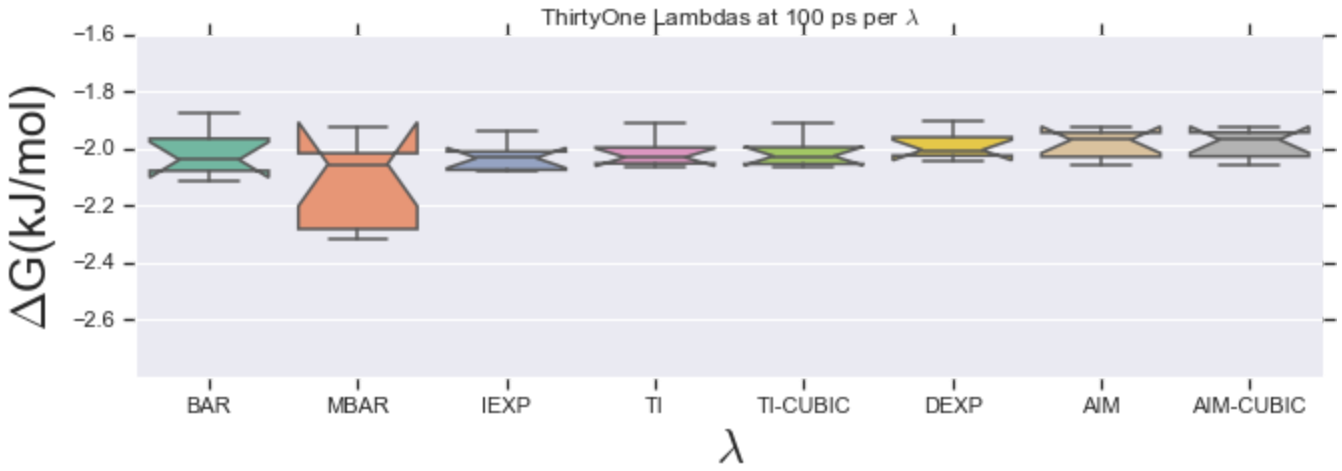
Originally, I was showing one of the above for each N lambdas at T time per lambda of 8 trials. The above plot is not saying very much with simple error bars over the mean. The next thing I've done was to change the plot to a box plot which gives a lot more information:



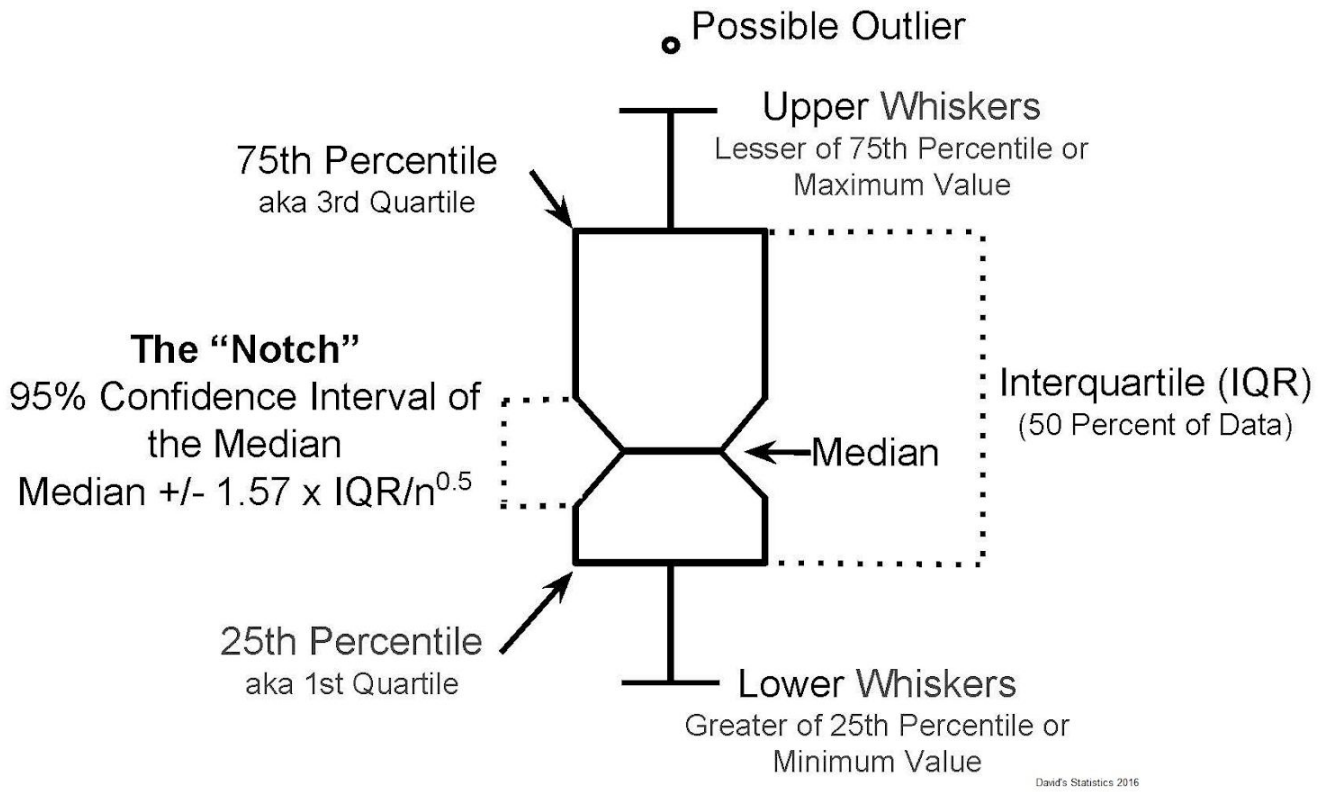
But I wanted to be able to show confidence levels so I added a notch:



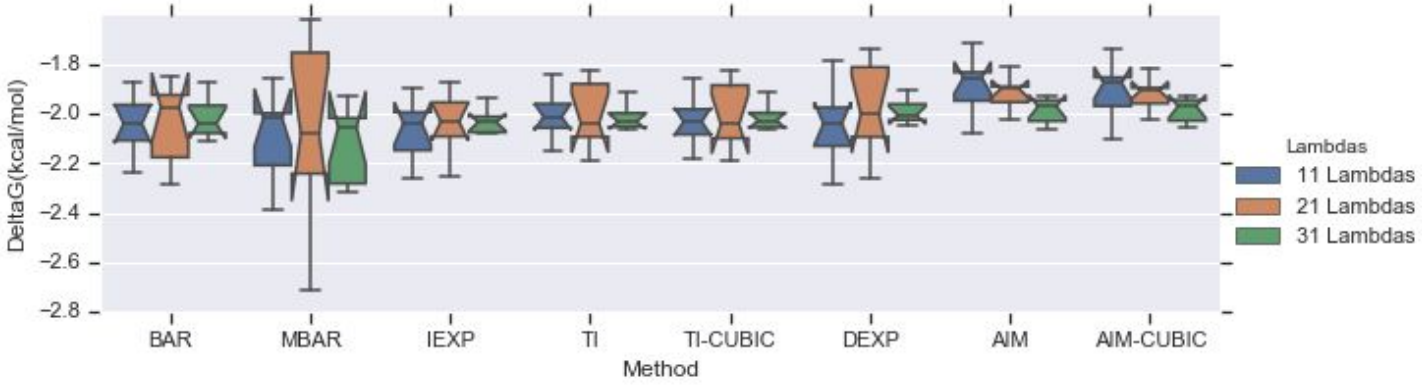

TwentyOne Lambdas @ 100 ps per λ
ΔG(kJ/mol)
−1.6
−1.8
−2.0
−2.2
−2.4
−2.6
BAR
MBAR
IEXP
TI
TI-CUBIC
DEXP
AIM
AIM-CUBIC
λ


ThirtyOne Lambdas at 100 ps per λ
ΔG(kJ/mol)
−1.6
−1.8
−2.0
−2.2
−2.4
−2.6
BAR
MBAR
IEXP
TI
TI-CUBIC
DEXP
AIM
AIM-CUBIC
λ

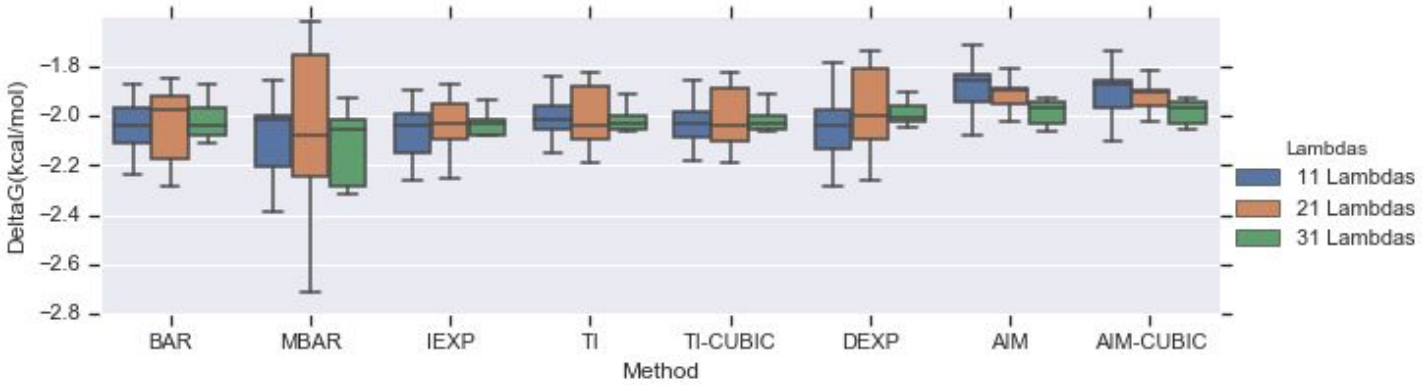
In case you don’t know how to read a boxplot with a notch (cuz I didn’t..):



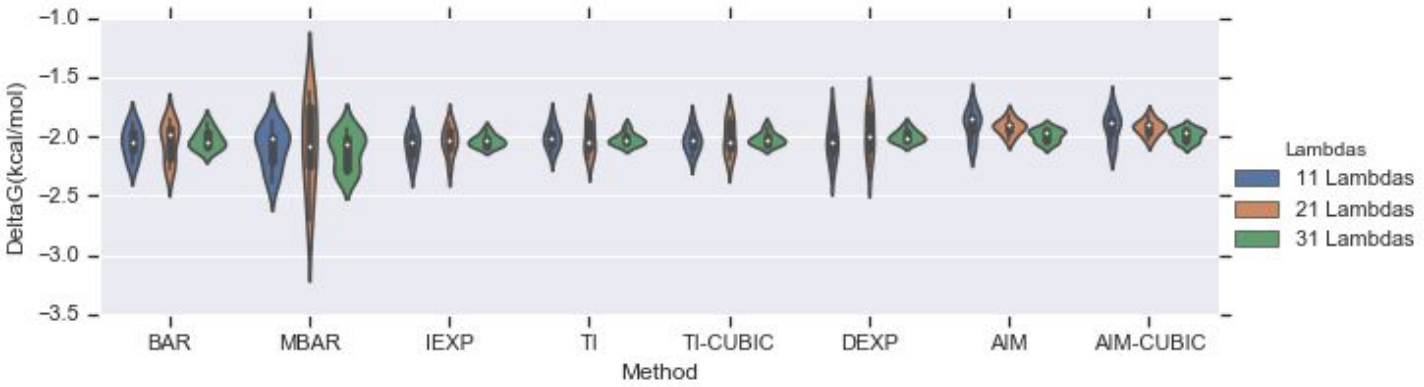
This is all lambda schedules at 100ps per lambda. The point of these simulations was to find the lambda schedule with the best chance to converge bot AIM and Fixed lambda sims.



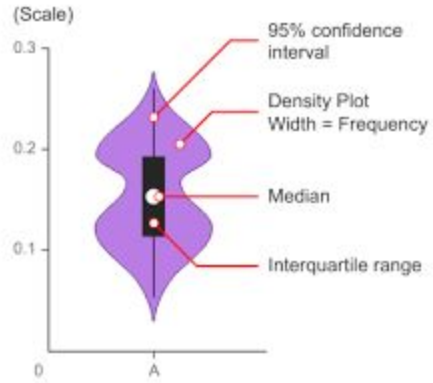
No Notch:



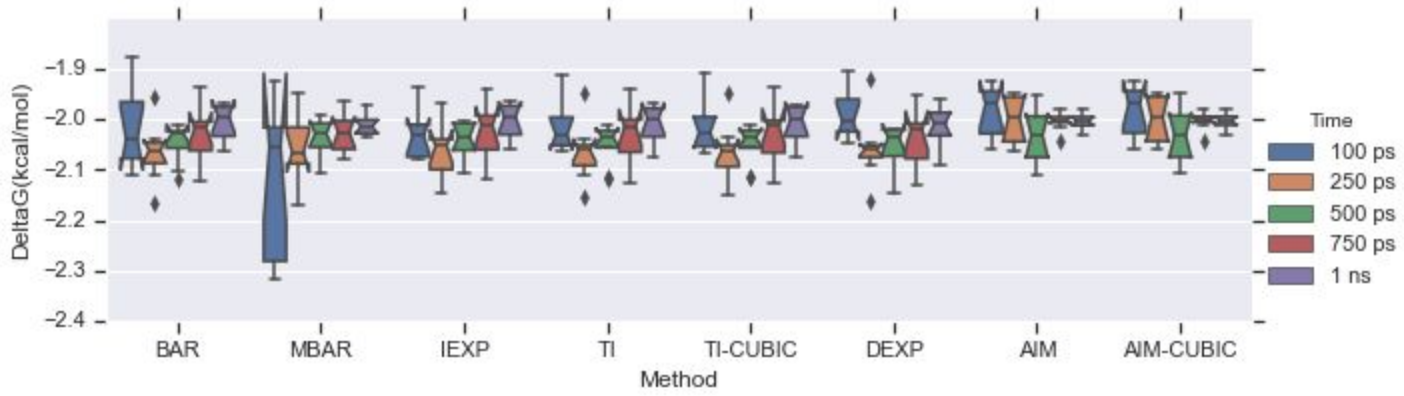
For fun, a violin plot of the same data. I prefer this plot but it may not go over well. Not very many people use it.



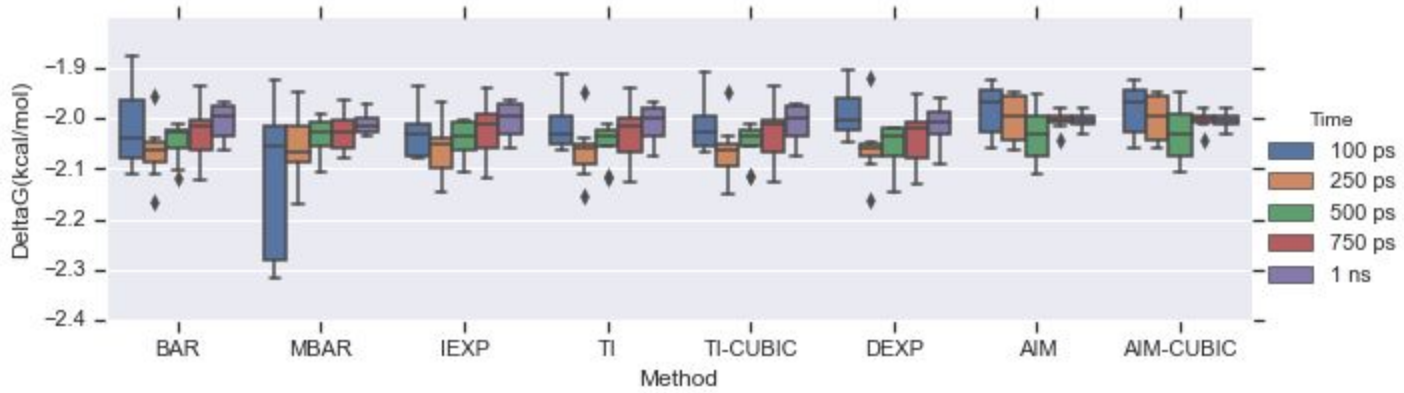
A violin plot is interesting because it shows the distribution of the data and the box plot as well. A violin plot has four layers. The outer shape represents all possible results, with thickness indicating how common. (Thus the thickest section represents the mode average.) The next layer inside represents the values that occur 95% of the time. The next layer (if it exists) inside represents the values that occur 50% of the time. The central dot represents the median average value.



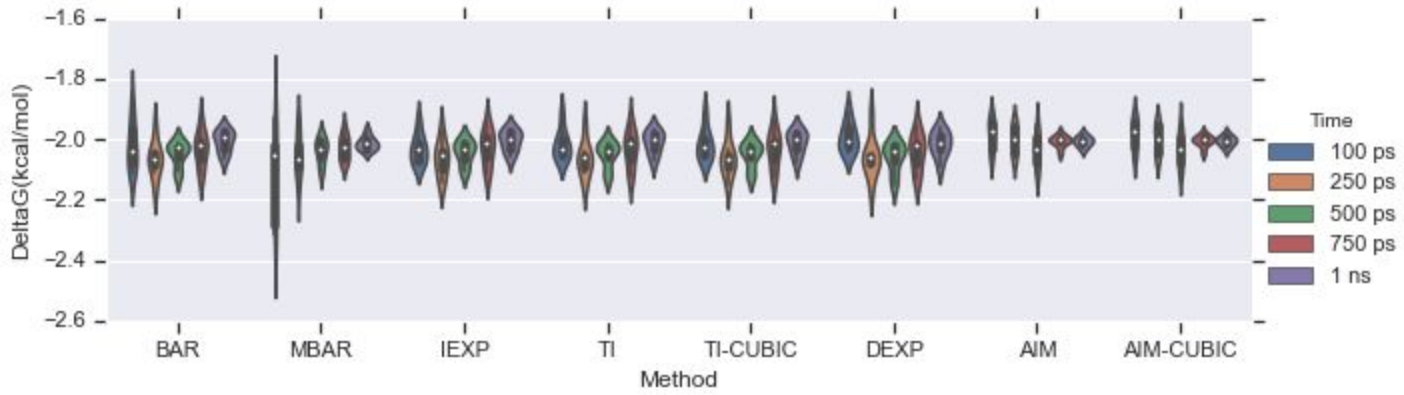
And then I extended the box plot to the final result of 31 lambdas and different times per lambda:



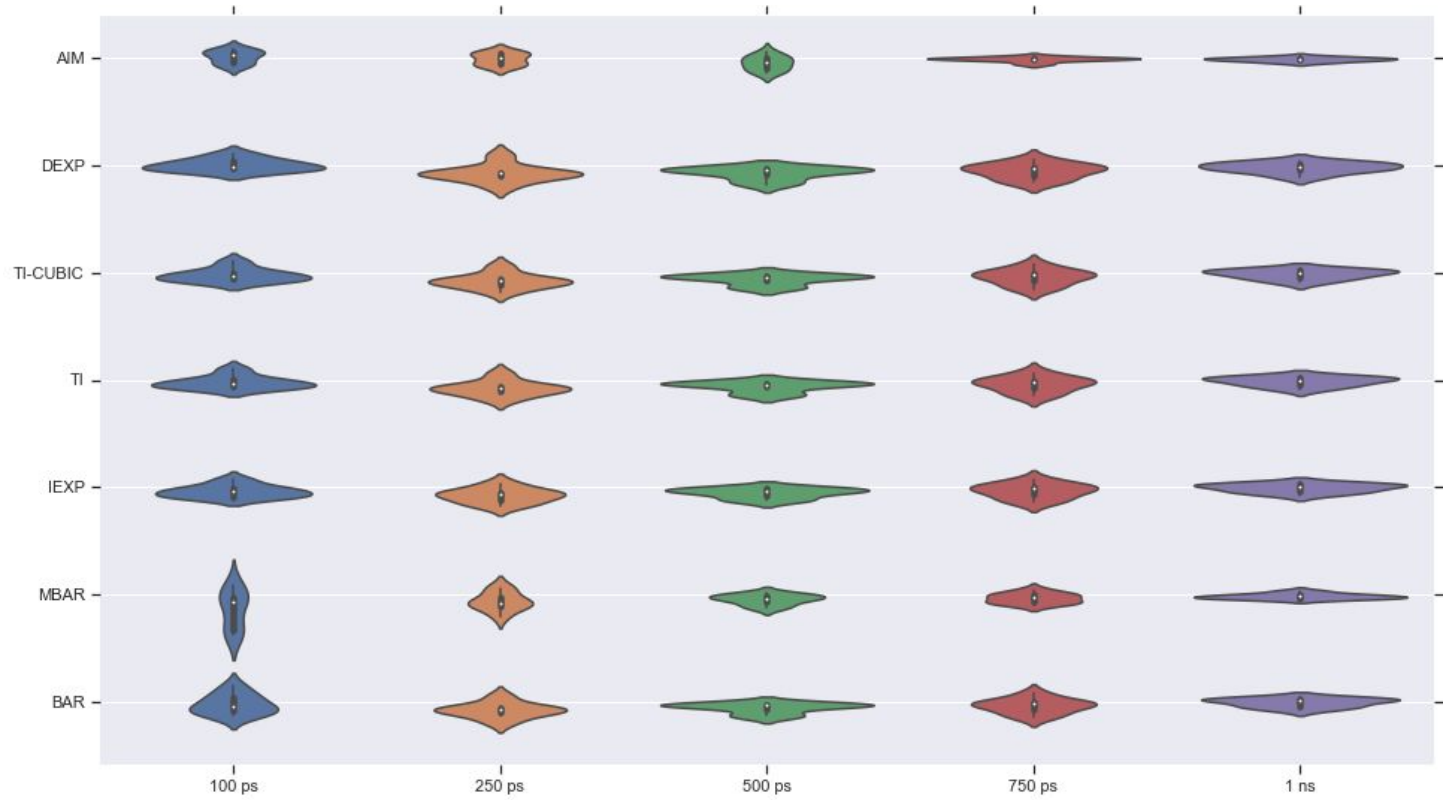
No Notch



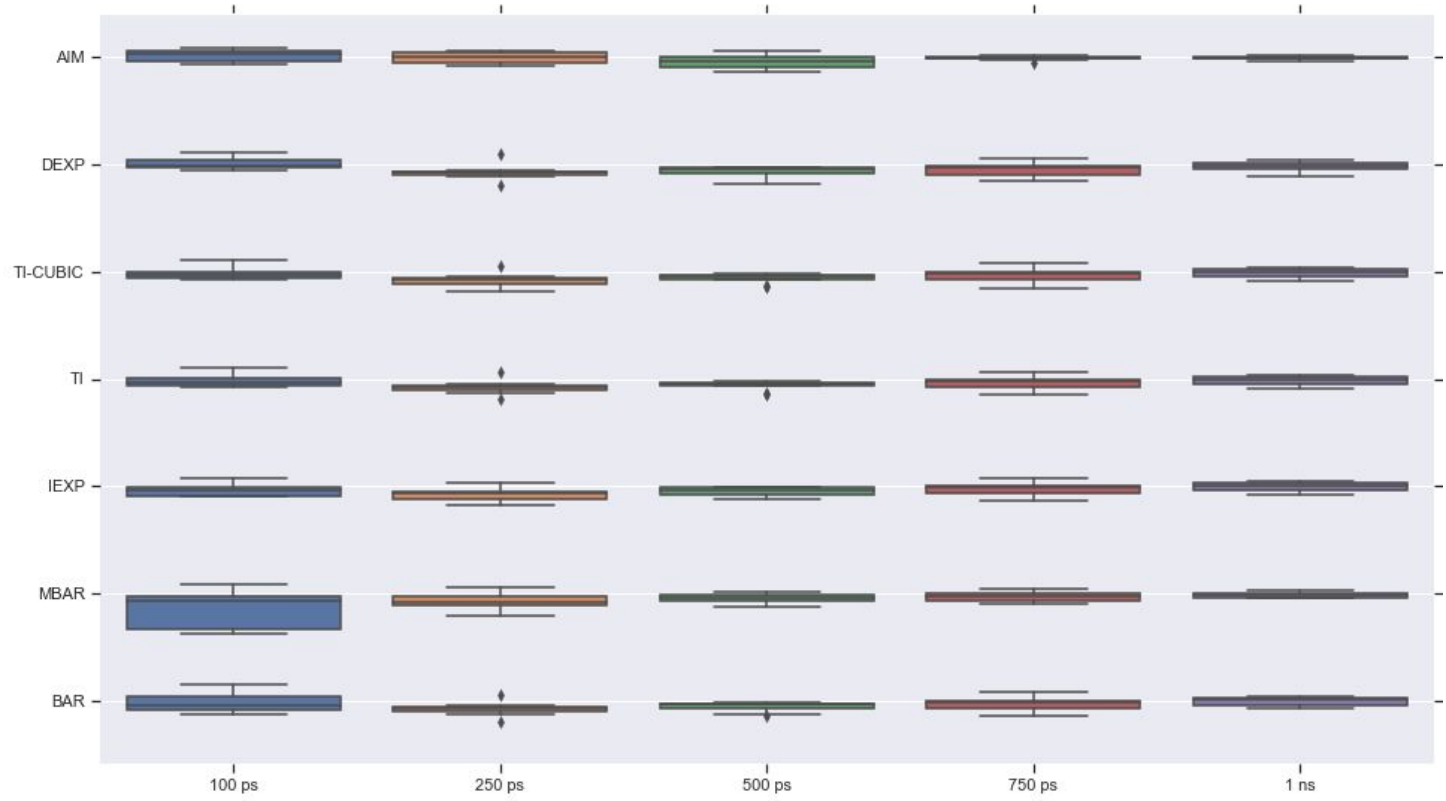
Violin plot of the same data



And finally a violin plot of the 31 lambdas and different time scales:



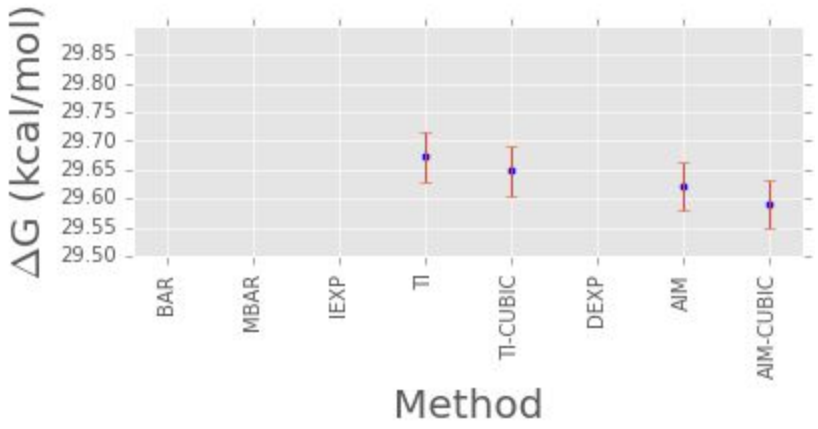
A box plot of the above did not look as appealing.

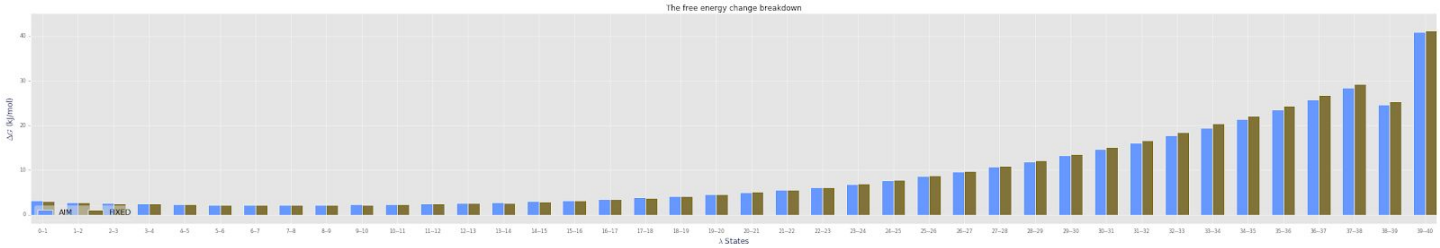


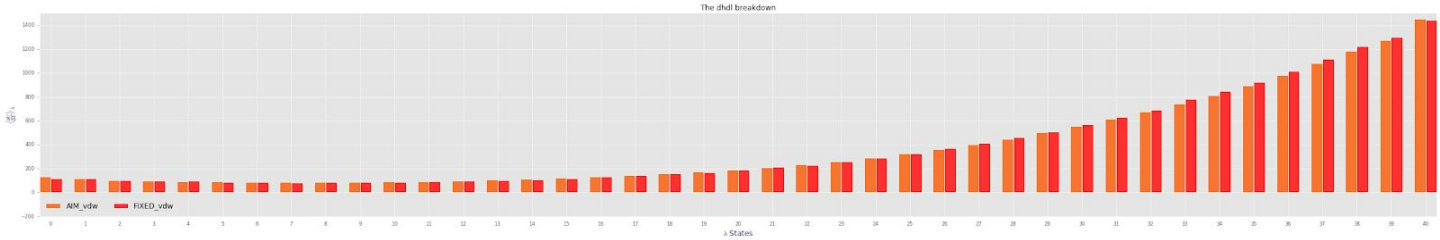
Current Problem: Where are you stuck?
Possible resolutions: What are you planning to do?

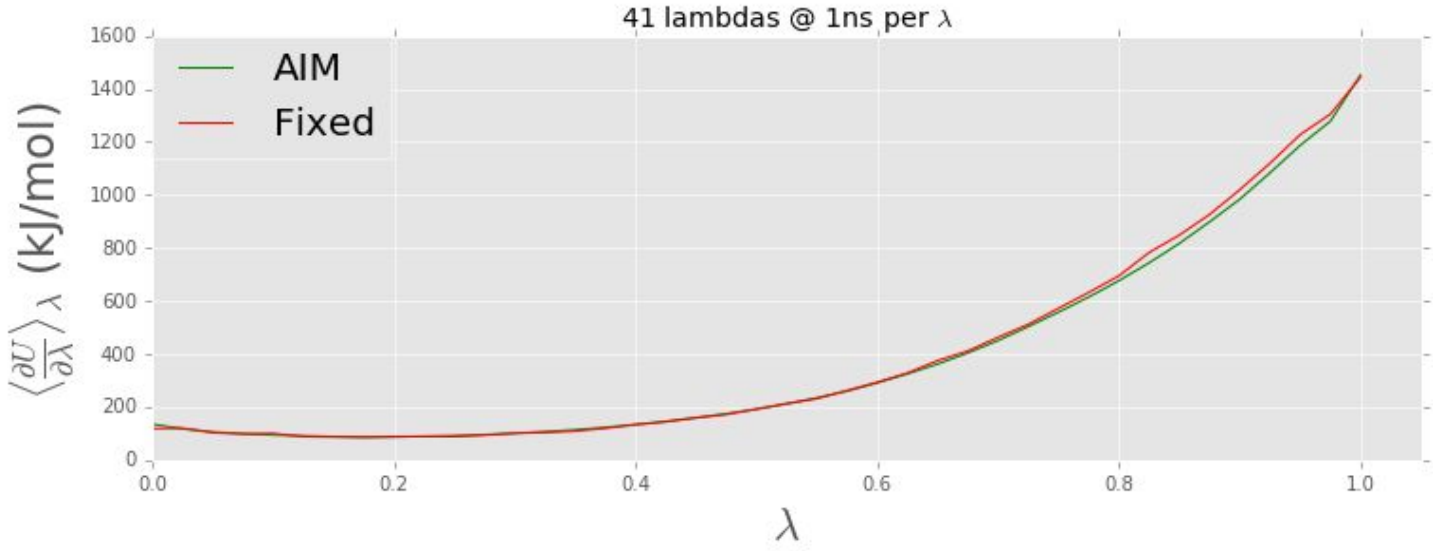
Date: 07/21/2018
Current Goal(s): What are you doing? Viewing results from 41 lambdas, 1ns per lambda sim

Update: What have you done?
I've run 8 simulations of 41 lambdas at 1ns per lambda.









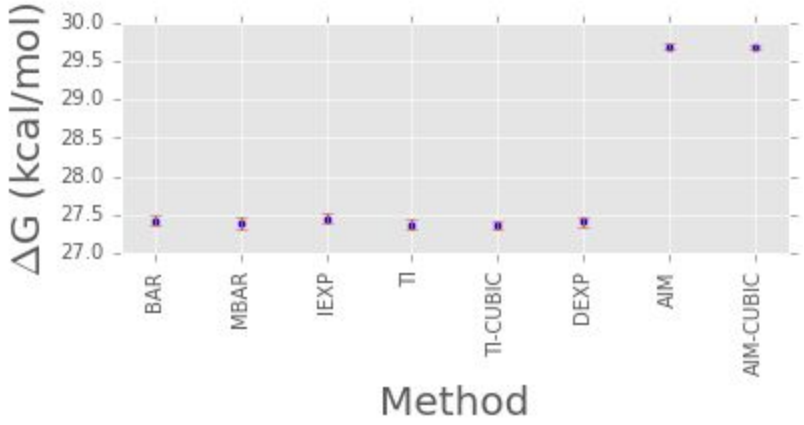
Current Problem: Where are you stuck? I don't have the resources to calculate the other methods. I need more RAM.

Possible resolutions: What are you planning to do? I believe I've succeeded at my PhD project. I can't be certain that a publication will pass peer review since I don't have the resources to complete the analysis nor would I have the resources to do an analysis on a longer simulation.

Date: 07/16/2018
Current Goal(s): What are you doing? Trying to figure out why the simulations aren't similar.

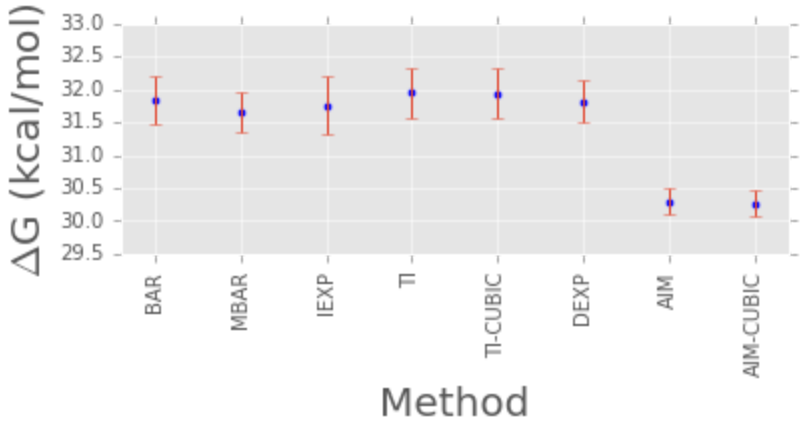
Each of the sims since update 04/18/2018 have had similar results to this last one. The fixed lambda sims don't match the AIM sims.

84 lambdas at 100ps per lambda with incorrect constraints.



Update: What have you done? After seeing the results of the 84 lambdas I noticed the pattern was the same for 41, 53 and 84 lambdas and decided to look further into a common cause in the mdp options. I found that the constraints for the fixed lambda simulations were not the same as the AIM simulations. Fixed lambda sims were using h-bonds and AIM was using all-bonds as the constraints.

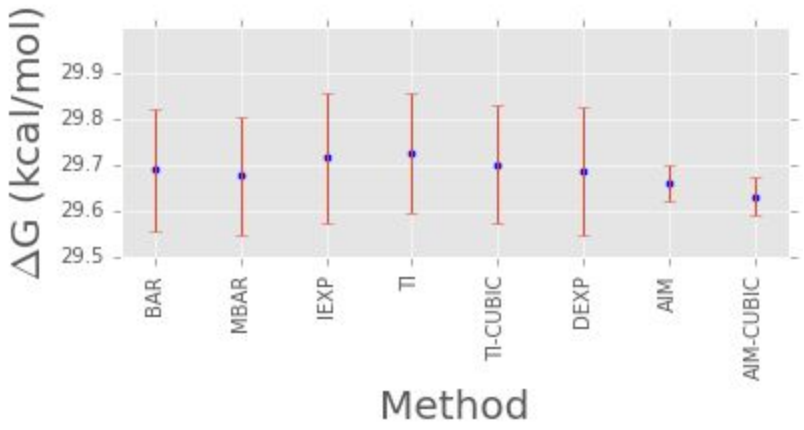
As a quick test I first ran a sim at 1ps per lambda and 41 lambdas with matching constraints



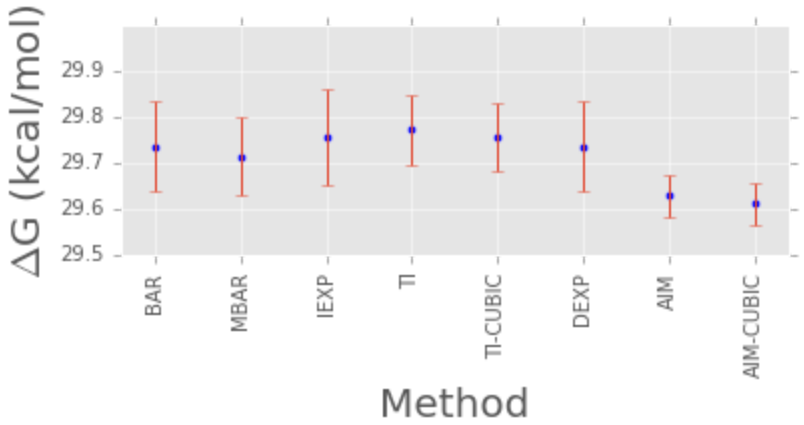
Once I saw the results I knew the constraints setting was the culprit and ran 41 lambas at 100ps in tandem with 53 lambdas at 100ps

With proper constraints, 100ps per lambda

41 lambdas



53 lambdas

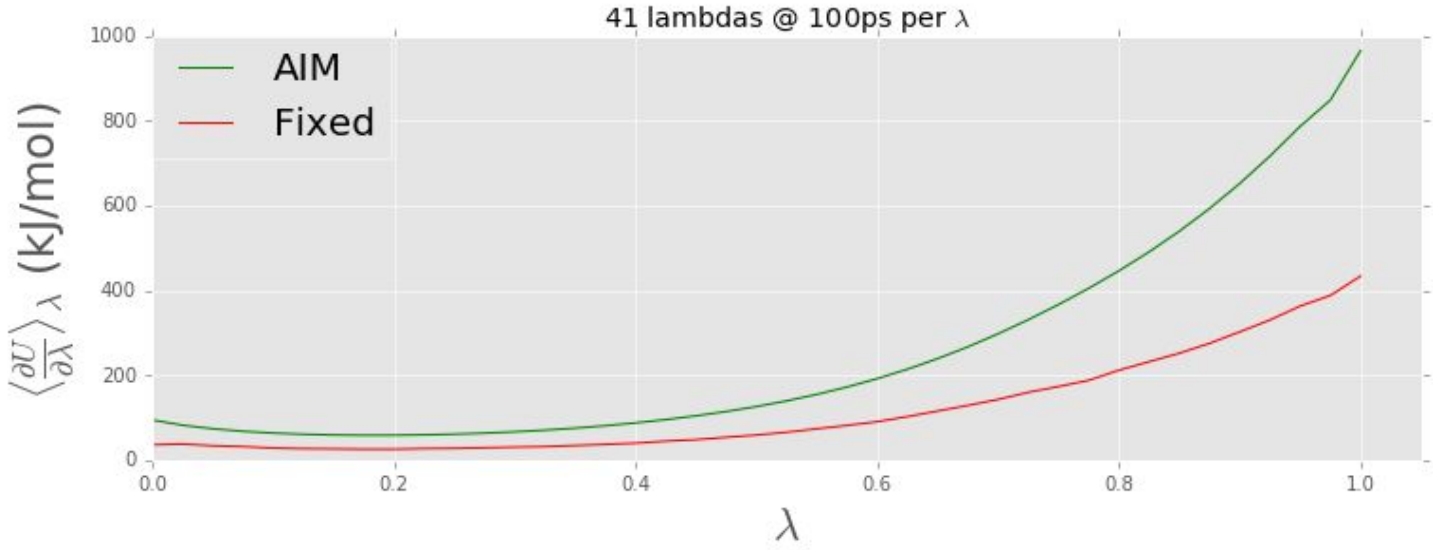


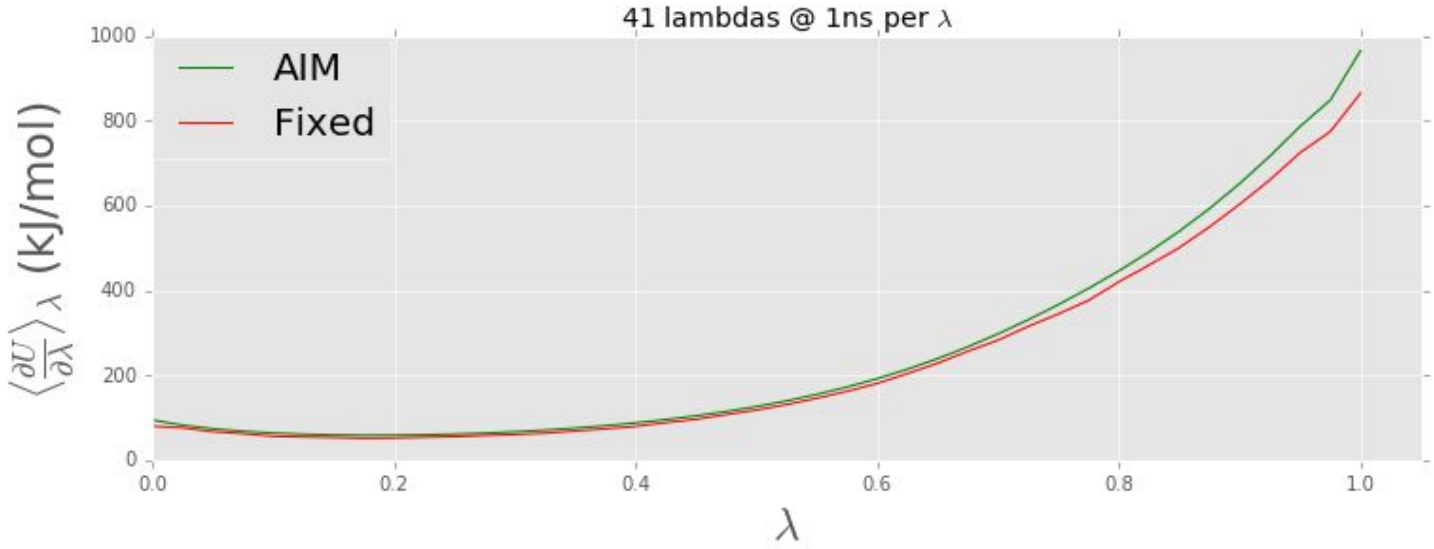
Current Problem: Where are you stuck? Not stuck, finally free.
Possible resolutions: What are you planning to do? Marty suggested that I run the simulation longer in order to show the same (or better) magnitude of convergence as we saw for the Methane sims. It looks like 41 lambdas is plenty so I will run those sims for 1ns per lambda.

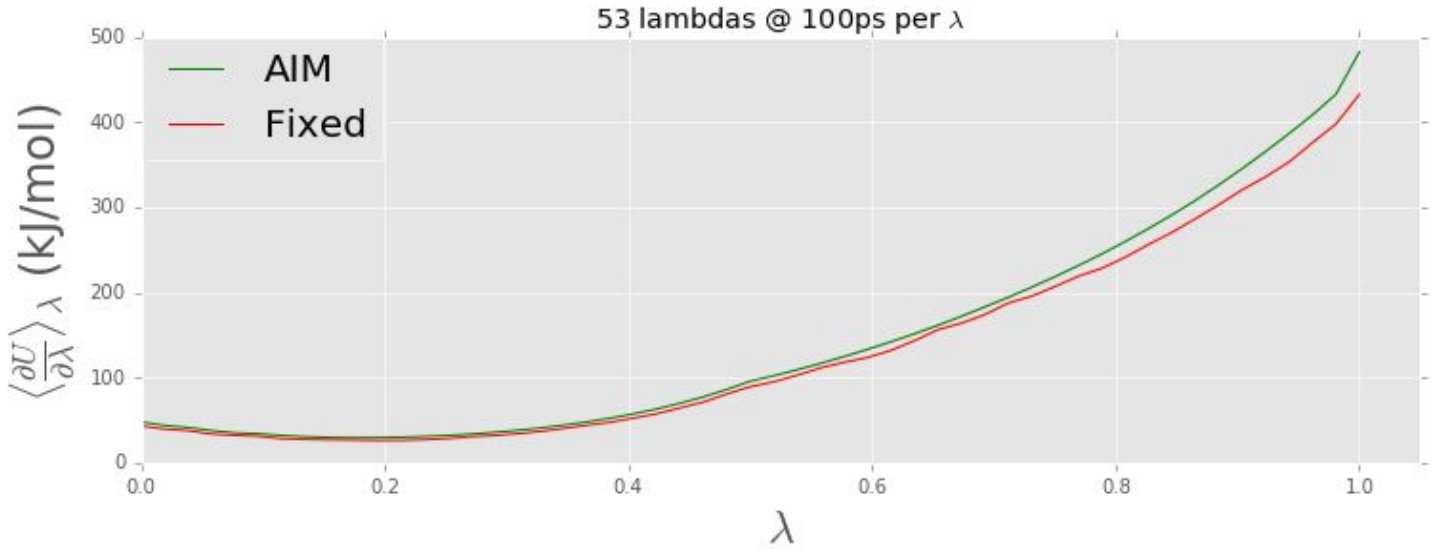
Date: 07/06/2018
Current Goal(s): What are you doing? Testing more lambdas and mdp options
Update: What have you done?

Mdp options:
I tried changing the coulombtype to PME and Cut-off. I'm still only getting about 7 ns/day.
The majority of the time is during force calculations and the free energy kernel. This is due to simulation being a mutation using data generated by PMX. I can't find a way to fix this.

MoreLambdas:
If we compare number of lambdas versus time per lambda we see that it looks like we are better off testing for more lambdas as opposed to running longer sims which is I've concluded previously. Once the correct lambda schedule is found, then running longer sims may be needed.


41 lambdas @ 100ps per λ
AIM
Fixed
$\left\langle \frac{\partial U}{\partial \lambda} \right\rangle_\lambda$ (kJ/mol)
λ

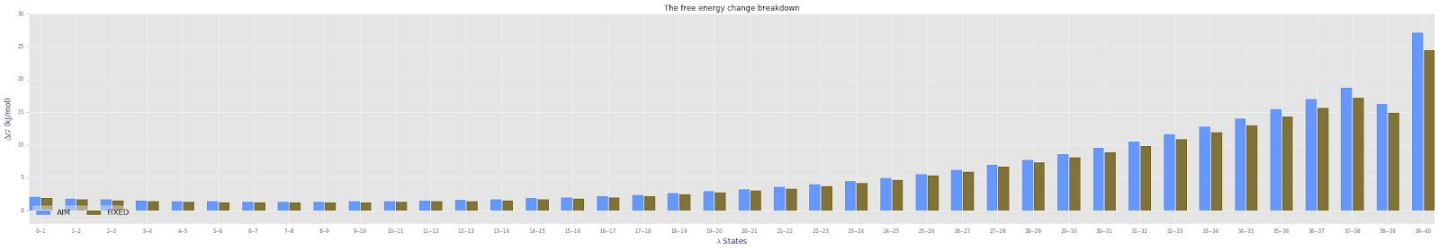

41 lambdas @ 1ns per λ
AIM
Fixed
$\left\langle \frac{\partial U}{\partial \lambda} \right\rangle_\lambda$ (kJ/mol)
λ

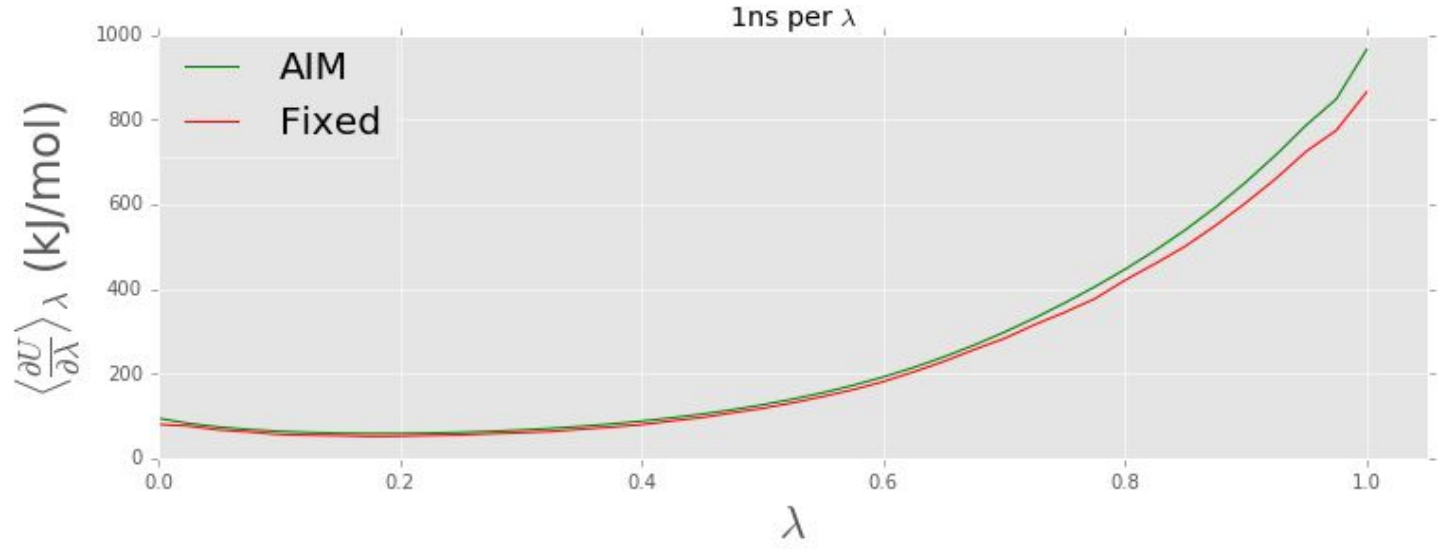

53 lambdas @ 100ps per λ
AIM
Fixed
$\left\langle \frac{\partial U}{\partial \lambda} \right\rangle_\lambda$ (kJ/mol)
λ

Current Problem: Where are you stuck? It looks like I need to increase the number of lambdas from 0.7 to 1.0 in order to decrease the gap between the two models but there is no guarantee this is the solution. However, increasing the number of lambdas is easier than running longer sims with my limited resources.
Possible resolutions: What are you planning to do? I'm going to increase the number of lambdas between lambda = 0.7 and lambda = 1.0.

Date: 07/01/2018
Current Goal(s): What are you doing? Comparing AIM to fixed lambda sims using A2V mutation.
Update: What have you done? I ran 8, 1ns per lambda sims with 41 lambdas using AIM and fixed models.

1ns per lambda





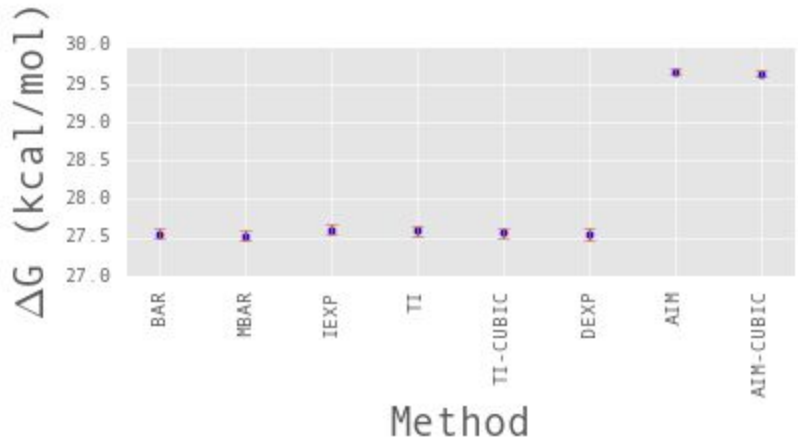
As you can see, the AIM solution doesn't change much from 100ps to 1ns per lambda. See 06/16/2018 update for comparison to 100ps. Fixed lambda simulations at 1ns per lambda have nearly the same sum total except that the constituent parts, the dhdl breakdown, is now more similar to AIM. The total, however, hasn't changed much. This is because the values starting at lambda = 0.8 make up the majority of the sum total. The values are very large starting around lambda = 0.8.

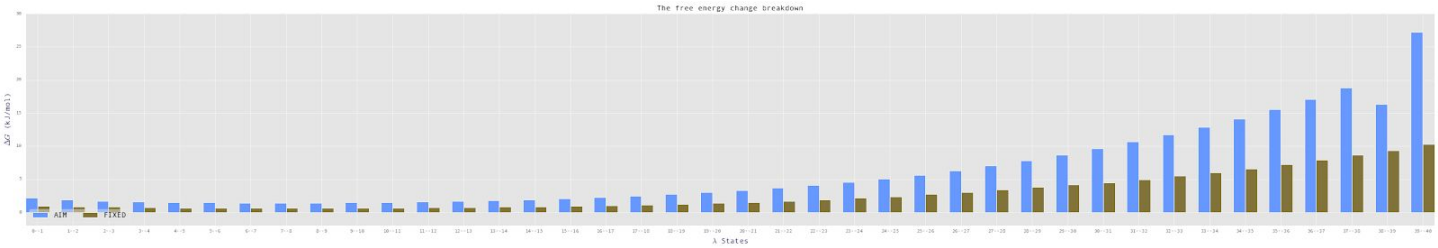
Current Problem: Where are you stuck? I'm not sure if more time or more lambdas is necessary. I ran a sim of 51 lambdas but there was a mistake in the lambda schedule that resulted in a spike and lots of confusion and cursing from me. More lambdas is quicker than longer sims so that's where I've started.
Possible resolutions: What are you planning to do? If more lambdas doesn't have a better product then I will run the 41 lambdas longer. I should have an update tomorrow.

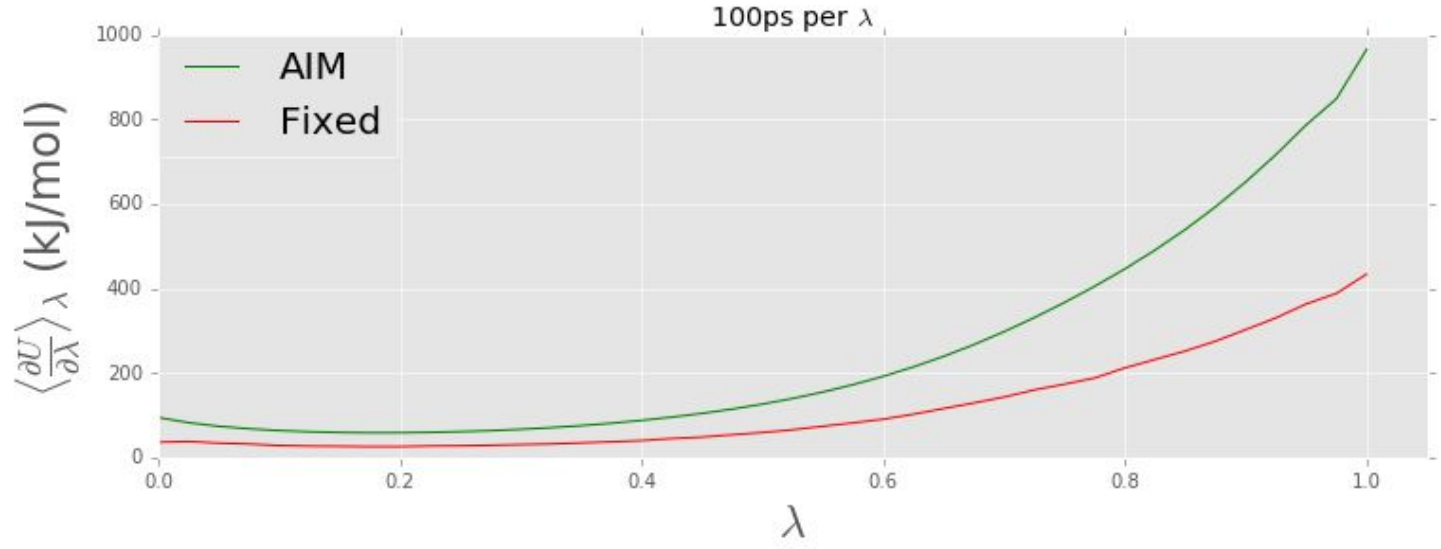
Date: 06/16/2018

Current Goal(s): What are you doing? Running 1ns per lambda sims against the A2V mutant with 41 lambdas
Update: What have you done?Started the cpu run last week after viewing the 100ps per lambda sims. Jagdish is using the GPUs.

100 ps results:







Both methods are converged but do not equal each other at 100ps per lambda which is why I decided to try 1ns per lambda.

Current Problem: Where are you stuck? The simulations are taking a long time and the cluster timed out with the error:
slurmstepd: error: *** JOB 475127 ON n050 CANCELLED AT 2018-06-14T19:20:02 DUE TO TIME LIMIT ***
I've tried different settings in the mdp file but I can only get 7 ns/day.

Possible resolutions: What are you planning to do? Restart the 1ns per lambda simulations at the stopped lambda.

Date: 04/28/2018
Current Goal(s): What are you doing? Gathering and collecting results.

I'm going through all of the current results and all of the simulations that I've done to try and decide what to keep. The Ethanol simulations seem important but they aren't really finished. I learned a lot by working through them and trying to explain my results but there is also a lot of confusion and hastily written code. The Methane simulations and results are much simpler and more complete.

If I stick with Methane and run simulations with fewer lambdas then I will have two types of efficiency for AIM. Time per lambda, and number of lambdas. A reviewer may still want to see a comparison between the other expanded ensemble methods but I can explain that we wanted to compare against the current standard. Ensemble methods aren't the standard at the moment. Fixed lambda sims are the standard and there are other papers that compare expanded ensemble to fixed lambda.

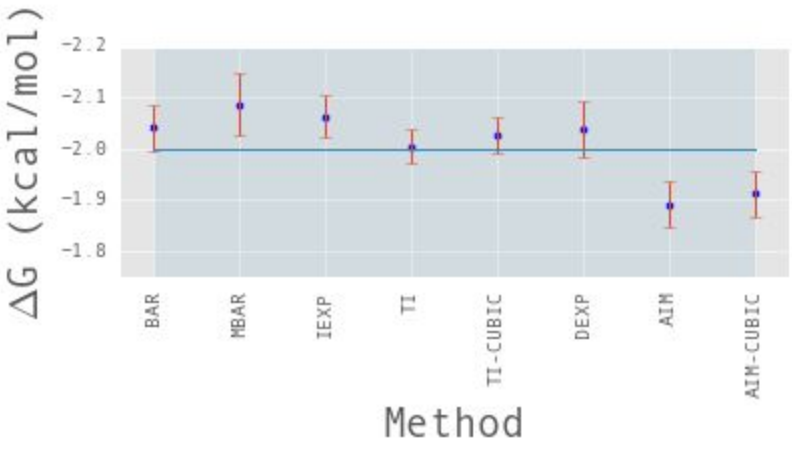
I'm reviewing my methods for the Methane analysis and making sure all of my results are standardized. For some of the Alchemical analysis results, my flags for the program were not consistent across all simulations. This won't change the results but everything needs to be consistent so I'm taking care of that.

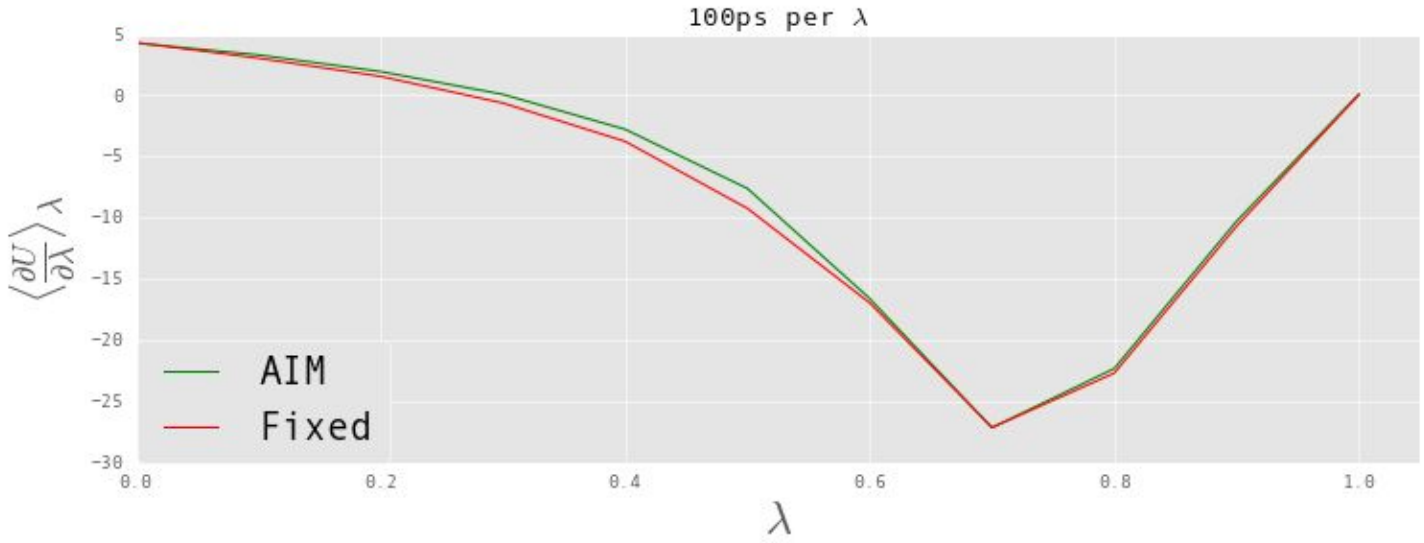
Plan moving forward:

1. Show that AIM is more efficient than standard methods in time per lambda and number of lambdas for a toy system (methane) in detail
    a. Compare 8 trials of one short simulation (100ps) each of the free energy breakdown between 11, 21 and 31 lambdas. This will show the process of finding the right lambda schedule and show that AIM is more efficient in areas of high variance.
    b. At 31 lambdas, compare 8 trials of 100ps, 250ps, 500ps, 750ps and 1ns per lambda to show that AIM converges sooner than the other methods.
2. Use configurations from the PMX mutation database to show that AIM works for a diverse set of benchmarks

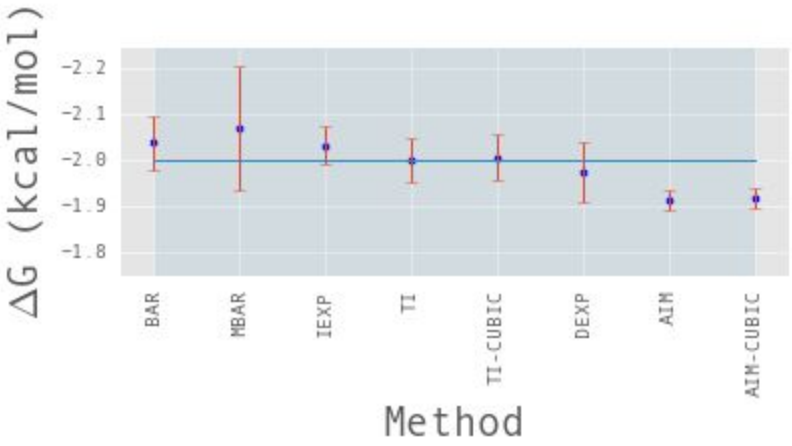
Update: What have you done?
Working on the above.

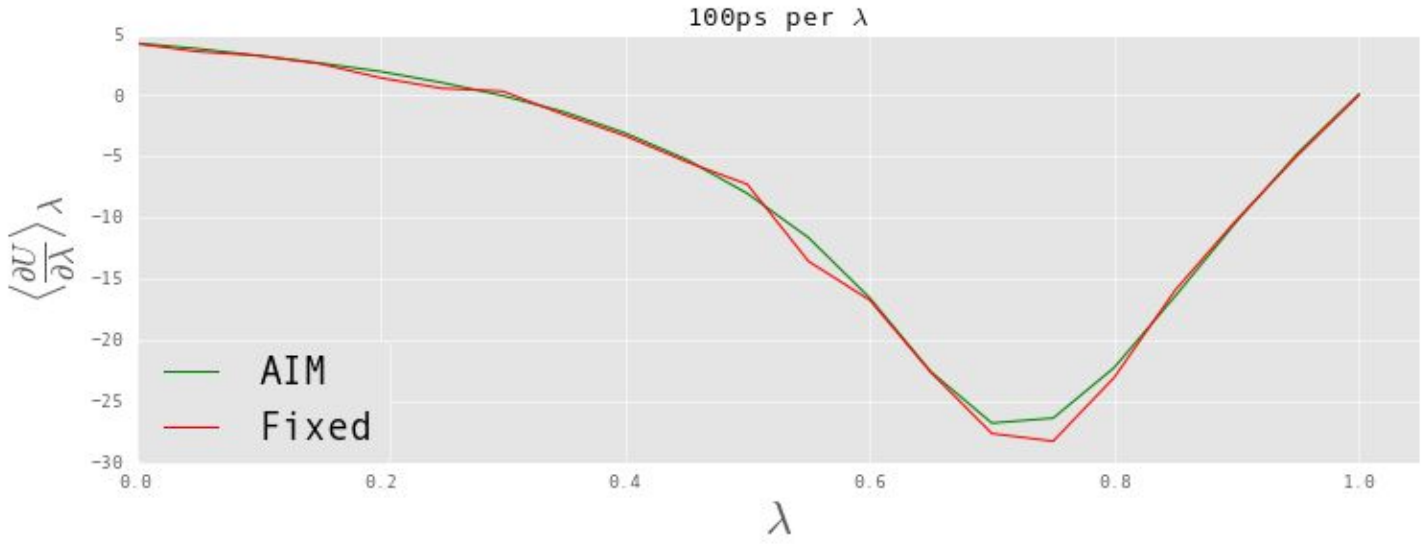
11 lambdas, 100ps per lambda, 8 trials



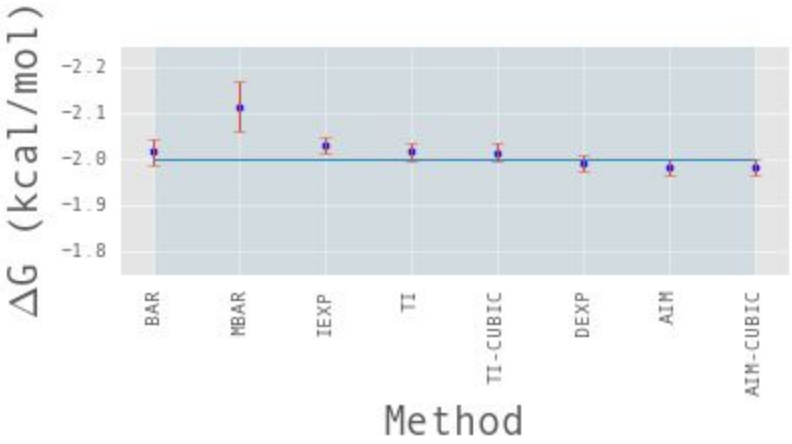


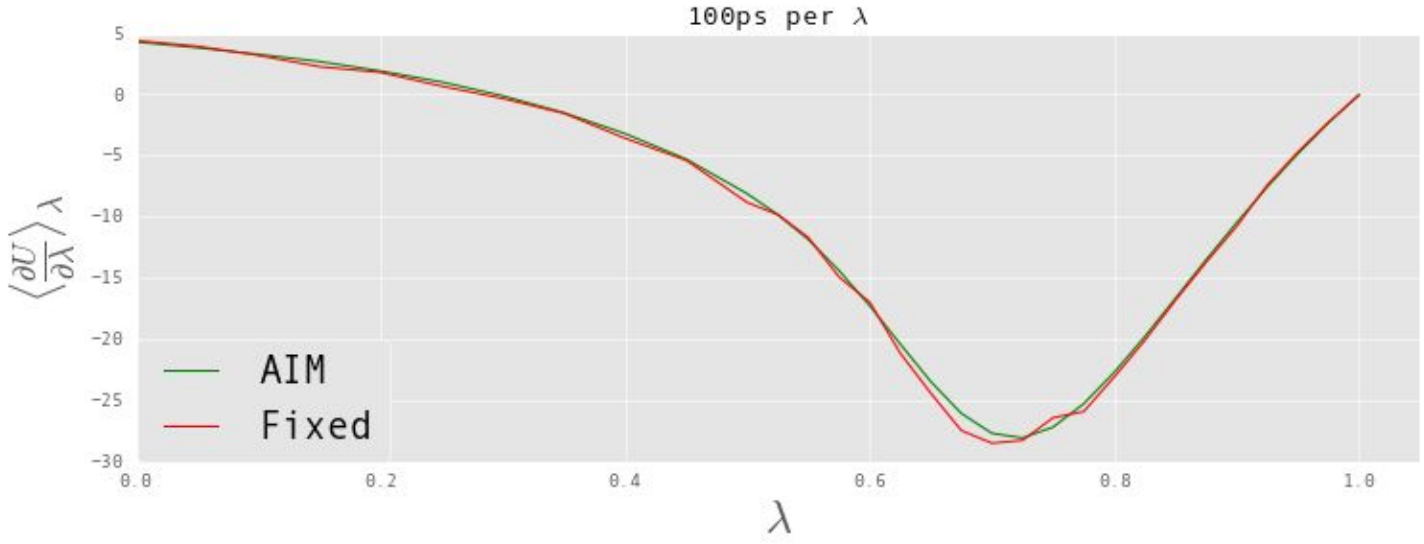
21 lambdas 100ps per lambda, 8 trials





31 lambdas 100ps per lambda, 8 trials

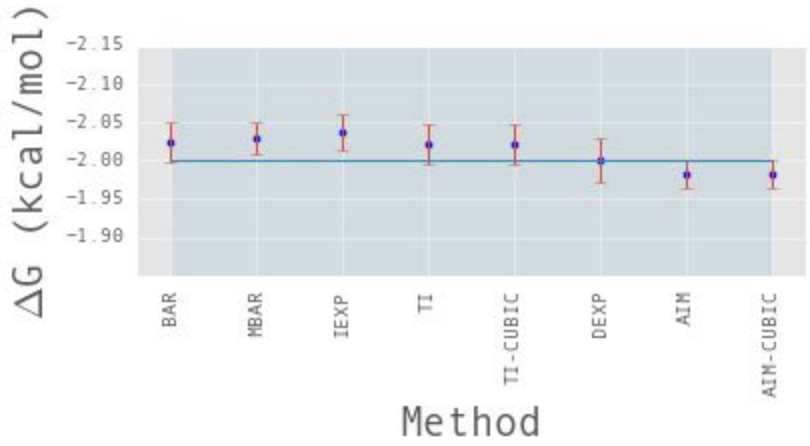




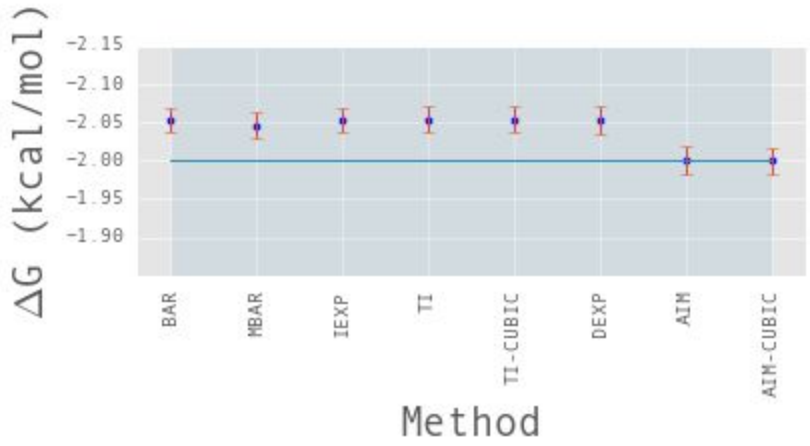
Current Problem: Where are you stuck?
Possible resolutions: What are you planning to do?

Date: 04/25/2018
Current Goal(s): What are you doing? Analysis on 8 trials of Methane at 100ps, 250ps, 500ps 750ps and 1ns
Update: What have you done? I have the results for 100ps, 250ps, 500, 750 and 1ns. It clearly shows AIM converging before the other methods.

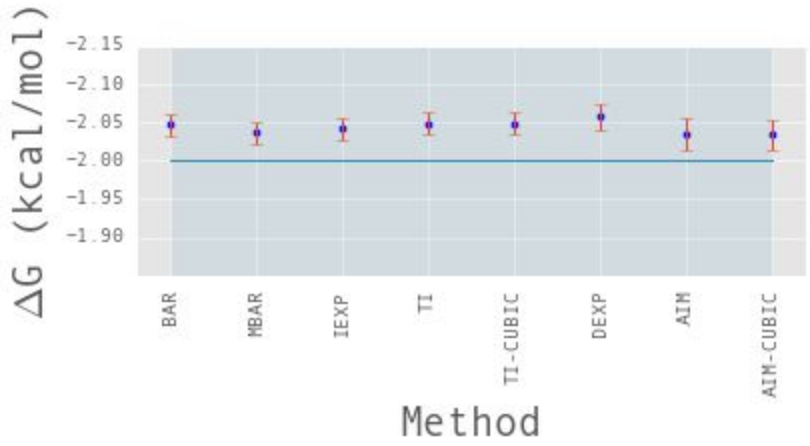
100ps per lambda



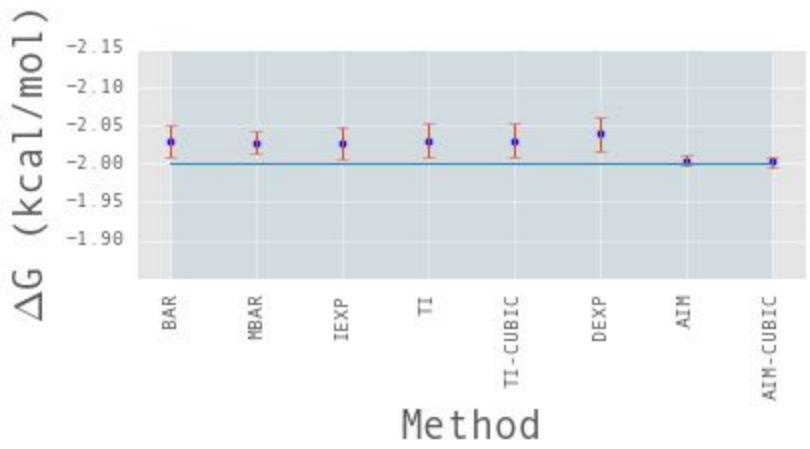
250ps per lambda



500ps per lambda

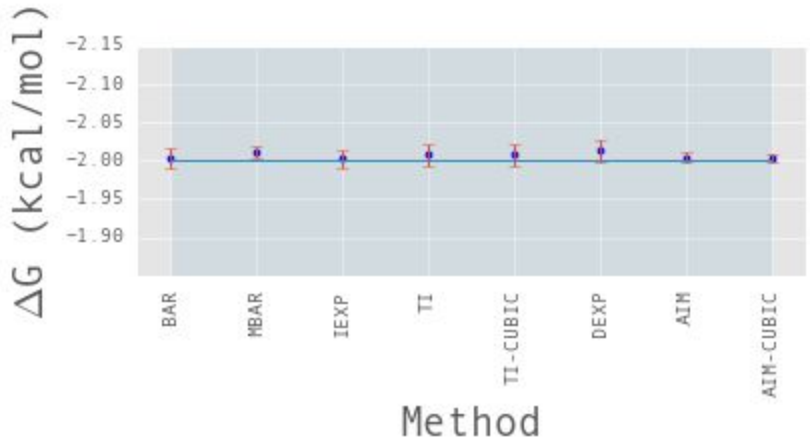


750ps per lambda

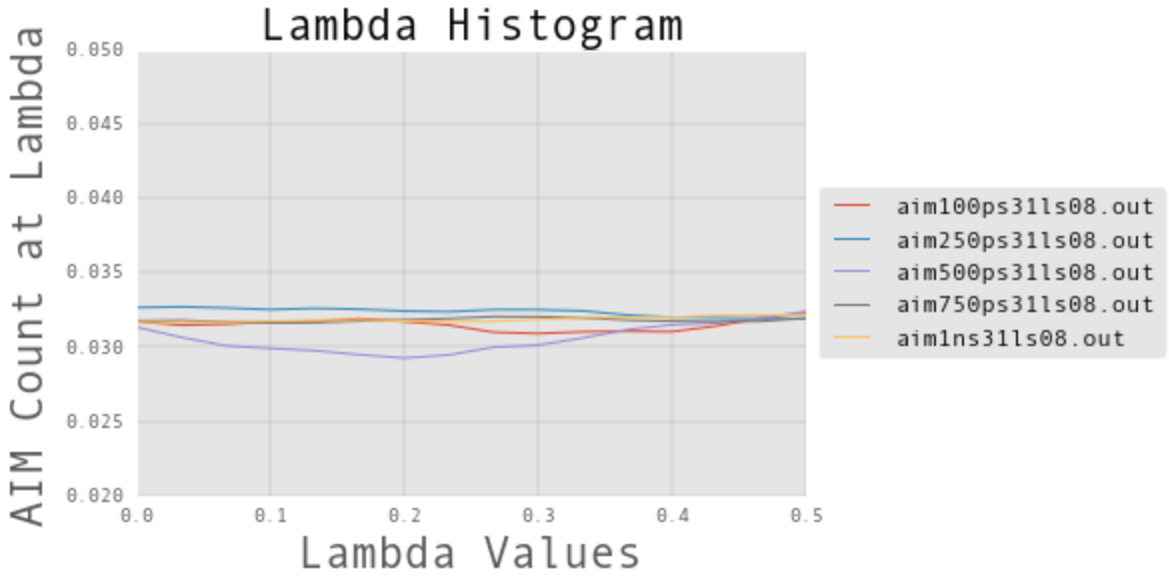


**Oddly, 500ps is more converged than 750ps. The reason is that averaging over 8 trials is not enough for the other methods.

1ns per lambda

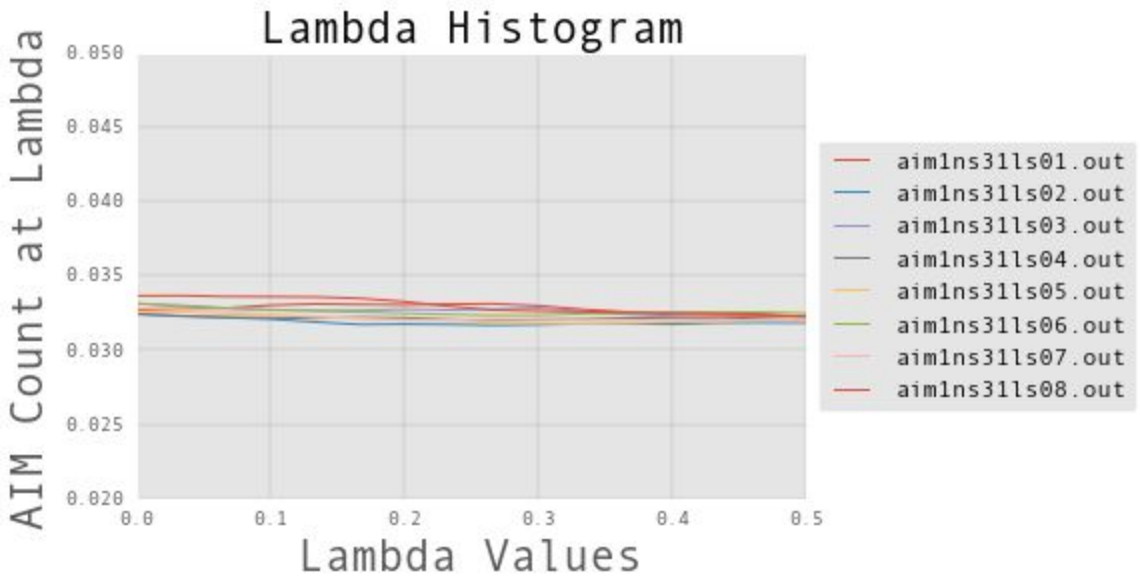


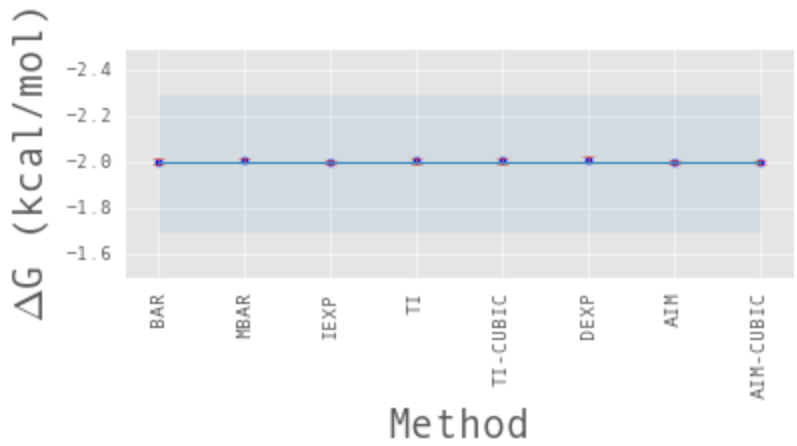
Histogram for last run, not the average.

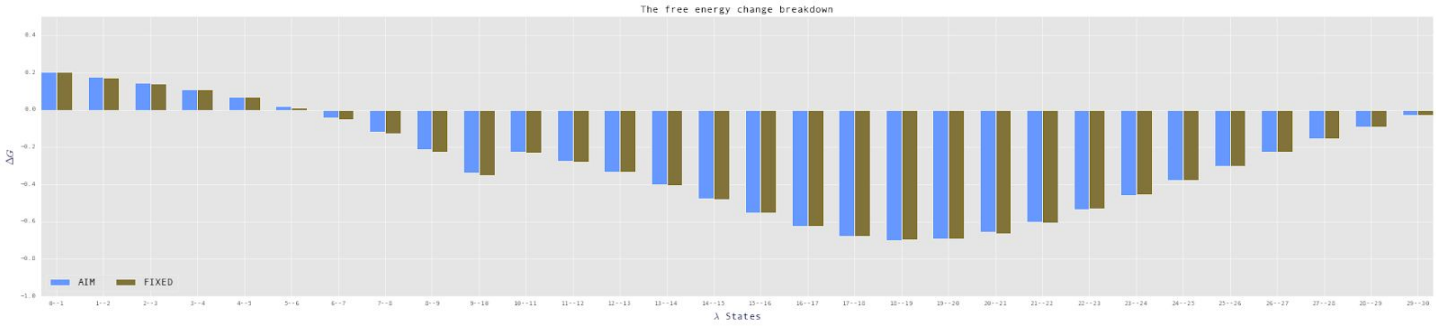


Current Problem: Where are you stuck? Do I need to do the same for 100ps and 250ps? I don't think they will be as dramatic. I have the sims finished and I have the results I just have not run the analysis. I also didn't print the free energy curves and the dhdl breakdown for these but I think this was all that was needed to see. The difference from 500ps to 750ps per lambda is the clincher for AIM being more efficient. Going from 750ps to 1ns we see that the other methods have converged within uncertainty to AIM.
Possible resolutions: What are you planning to do? It's 1 a.m. I should probably go to bed.

Date: 04/14/2018
Current Goal(s): What are you doing? Analysis on 8 trials of Methane at 1ns per lambda and 31 lambdas
Update: What have you done?
I have run 8 trials of Methane molecule at 1ns per lambda and 31 lambdas.







The Free Energy estimates:
BAR = -8.38402473273 +/- 0.0524982521725
MBAR = -8.41395321816 +/- 0.0322909454545
IEXP = -8.37777690146 +/- 0.0511271630074
TI = -8.39870807943 +/- 0.0556857632342
TI-CUBIC = -8.39938353628 +/- 0.0565393727374
DEXP = -8.4204782643 +/- 0.0619159480573
AIM = -8.38248464062 +/- 0.029278362544
AIM-CUBIC = -8.37896061677 +/- 0.0293107711887

Current Problem: Where are you stuck? Not stuck
Possible resolutions: What are you planning to do?
Per Marty: "... I would like to see estimates for 30 lambdas for different time points. Also, if you agree with me, I think it would be interesting to show some time points for 21 lambdas because it would appear that AIM is requiring fewer lambdas to get to the large-lambda limit compared to the other methods."

I don't think separate time points is necessary but different lambda counts could be. It's already been shown in literature (and the Ethanol sims) that increasing time per lambda doesn't improve agreement between methods in realistic time scales and between correlated samples. The easiest way to show agreement is with a reasonable time scale, sufficient number of lambdas, and multiple trials. It may be enough to show increased agreement with 1ns per lambda using 9 lambdas, 21 lambdas, and 31 lambdas. Increasing the number of lambdas and the number of trials is how I achieved convergence and agreement. For the Ethanol and Methane simulations, increasing time per lambda wasn't going to show convergence for 21 lambdas within the constraints of time and resources. But increasing the number of lambas for a single time point shows increasing agreement between all methods.

Originally you stated that I would need mutations as well. I have the pieces ready to run AIM against several mutations provided by the PMX server but I haven't had time to create the workflow. Soon I plan to transition my primary simulations away from what is interesting to focus on material necessary for the publications. I believe that I have more than enough to write a dissertation, but in order to produce two quality papers within the timeframe I have left, I need to enter the writing process very soon.

Plan:
8 trial simulations of Methane at 1ns per lambda and 31 lambdas (not 30, I miscounted before) took 24 hours.
8 trial sims of Methane at 500 ps per lambda, 31 lambdas, estimate 12 hours.
8 trial sims of Methane at 100 ps per lambda, 31 lambdas, estimate 2.5 hours

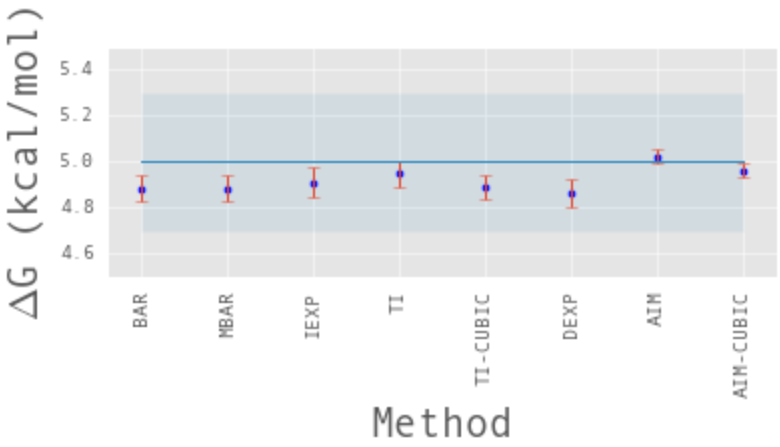
8 trial sims of Methane at 1ns per lambda and 21 lambdas, estimate 24 hours

8 trials sim of Methane at 500 ps per lambda and 21 lambdas, estimate 10 hours.
8 trails sim of Methane at 100 ps per lambda and 21 lambdas, estimate 2 hours.

I think this is what you're asking for which will take at least a week, maybe more. It depends on what's going on at work and the stability of the cluster.

Date: 04/08/2018
Current Goal(s): What are you doing?
Requested deliverables:

1. "Go back to comparing AIM to standard methods as a function of time (10 ps, 50 ps, 100 ps, ...)"
    a. Reasoning: This should tell us how efficient AIM is relative to the other methods.
    b. Resolves the statement: Identifying AIM as one of the most efficient free energy methods and proving that AIM improves sampling over other free energy methods.
    c. This has been done with Ethanol
        i. Since AIM is an expanded ensemble method, proving AIM improves sampling over fixed lambda simulations is a given. Comparing AIM to fixed lambda simulations is not comparing apples to apples. It has been shown that expanded ensemble methods spend more time in high-variance areas, and less in other areas, thus they are more statistically efficient than fixed lambda simulations.
            1. RE: Why AIM doesn't equal the other methods throughout the course of the entire simulation: When deciding on a series of intermediate states:
                a. "The variance shrinks very quickly as a function of state spacing. Until the free energy differences between intermediates are lowered to approximately 2–3 kBT, and if sufficient CPUs are available, it is better to use more states than fewer states. If limited by the number of CPUs available, fewer states can be used. However, it may end up being less statistically efficient, as more uncorrelated samples will be required from each simulation."["An Introduction to Best Practices in Free Energy Calculations"].
        ii. If we define convergence as decreased variance in the mean and we separate the ideas of accuracy and convergence, accuracy being dependent on the force field and convergence being independent of accuracy, and we also keep in mind that we have shown, given a sufficient number of lambda states, that AIM is equal to fixed lambda simulations, then we've already shown that this is true. We've shown that for the Ethanol molecule at 1ns per lambda for 21 lambdas that AIM converges to a single value sooner than the other methods.

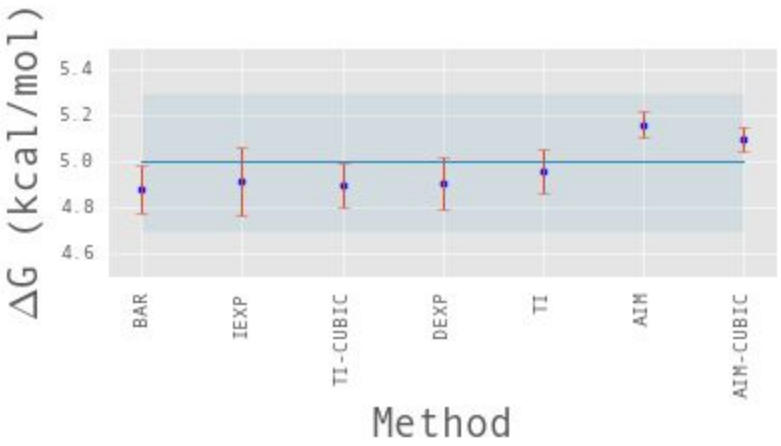


1.

iii. Simply by viewing the error bars (the variance from the mean) from the results of this simulation we show that AIM is more efficient using less lambdas and is statistically more efficient since expanded ensemble methods are able to escape certain kinetic traps. We see that AIM has nearly converged (the distance between the error bars is shorter) and the other methods have not. We also see that AIM is closer to the experimental value for the given molecule and mdp settings although accuracy should not be a factor in this comparison.

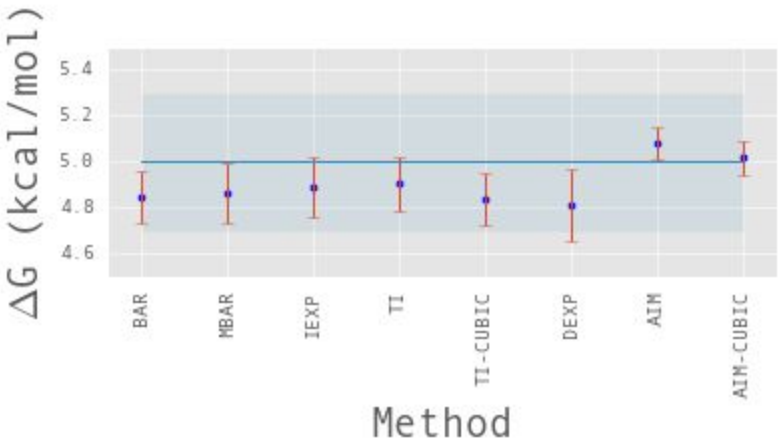
iv. We are only averaging over 5 samples because we are not interested in accurately predicting the mean but what we desire to show is this trend in increasing convergence for increasing time scales. Using the Ethanol molecule and 21 intermediate states we see that AIM converges sooner than the other methods:

1. 100ps per lambda



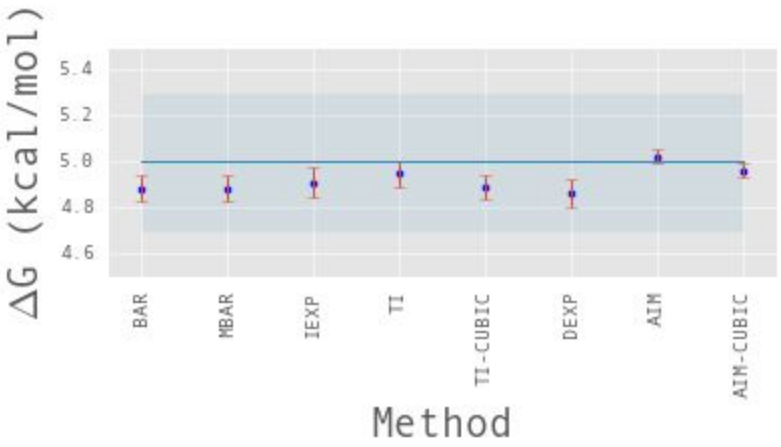
a.

2. 250ps per lambda



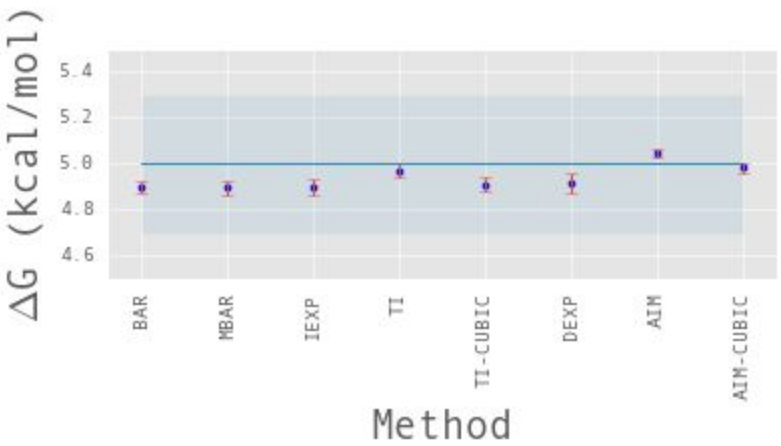
a.

3. 1ns per lambda



a.

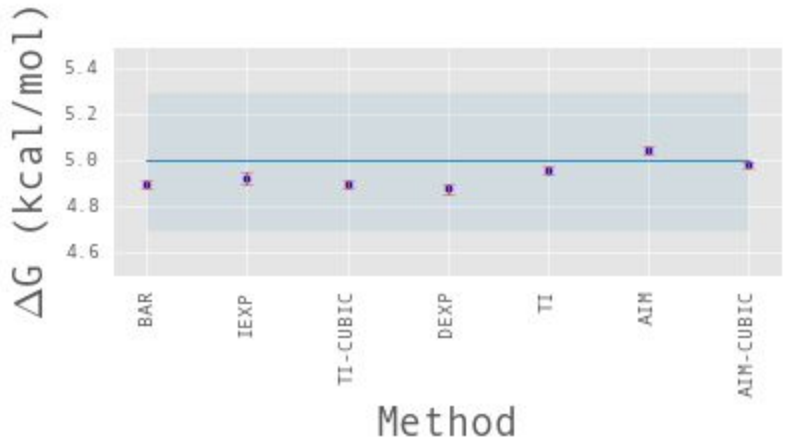
4. 5ns per lambda



a.

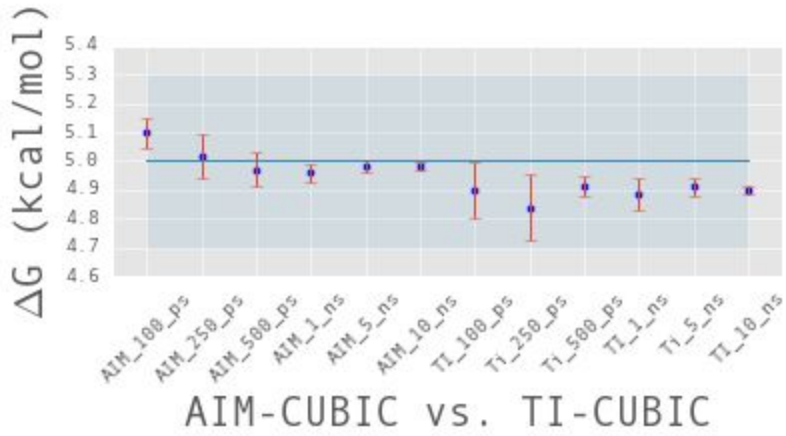
Update: What have you done? From the above I've shown that AIM is more efficient than fixed lambda simulations in that AIM converges sooner with less lambdas and less time per lambda than other methods. Showing this using the Ethanol molecule is more compelling since we have simulations up to 5ns per lambda to compare all methods and 10ns per lambda to compare all but MBAR.

10ns per lambda



The comparison of MBAR at 10ns per lambda is not possible without more RAM as the computation requires the difference between all lambda states at each lambda to be present in the XVG file.

We have also shown a condensed comparison of convergence between AIM-Cubic and TI-Cubic. We see at 10ns per lambda that AIM and TI are similarly converged as long as our definition of convergence is variance from the mean.



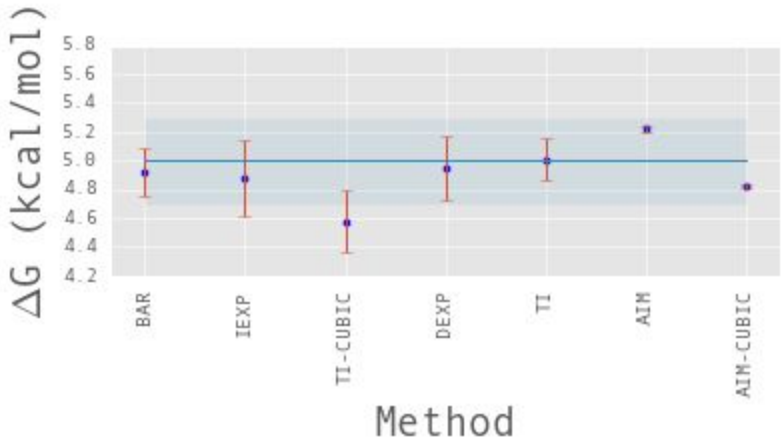
	AIM_100_ps	AIM_250_ps	AIM_500_ps	AIM_1_ns	AIM_5_ns	AIM_10_ns	TI_100_ps	TI_250_ps	TI_500_ps	TI_1_ns	TI_5_ns	TI_10_ns
count	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000
mean	5.097123	5.017147	4.968581	4.959230	4.981603	4.980696	4.900545	4.838080	4.914175	4.886008	4.909139	4.898823
std	0.060365	0.083832	0.066423	0.034197	0.022683	0.017195	0.107214	0.126750	0.038645	0.060509	0.035371	0.018996

The statistics table shows AIM and TI have similarly converged at 10ns per lambda by standard deviation (std) for the same number of intermediate states and sample size.

The next possible comparison would be convergence within statistical uncertainty which means comparing increasing time per lambda using 33 or more intermediate states and averaging over multiple samples. In my opinion, it would be more compelling to do this with Ethanol given everything above and previous updates below. Doing this comparison with a different molecule seems disconnected unless it is expected that all work

done against Ethanol is to be repeated against this different molecule. Since we have already seen agreement within statistical uncertainty at 100ps per lambda using 33 lambdas I think we should first look at 100ps per lambda with 5 samples of each. This may show AIM to converge sooner. This single simulation along with what is given above will be further proof that more lambda states are required for the fixed lambda simulations to converge.

For the Ethanol molecule using 4 samples of 100ps with 33 lambdas per lambda simulation (4 because the 5th is still running)



MBAR isn't shown because I forgot to change the mdp settings so I will need to run these again. There may be an argument that 5 samples isn't enough but this shows that AIM has already converged at 100ps per lambda for the given lambda schedule of 33 lambdas.

Current Problem: Where are you stuck?

Possible resolutions: What are you planning to do?

Date: 03/31/2018
Current Goal(s): What are you doing?
Requested deliverables:

1. "Why don't TI and AIM agree?"
    a. Fixed lambda free energy simulations use the equal time rule for intermediate state sampling
        i. The TI method underestimates variance
        ii. Its result is influenced by the integration method used and is determined by how the curvature varies along the pathway. See updates below this one for further reasoning
    b. Expanded ensemble methods are powerful tools for collecting uncorrelated data
        i. AIM aims at minimizing the overall variance of the free energy estimate by adjusting the distribution of the number of data points in each intermediate state based on the acceptance criteria (Note to me: This needs to be reworded but is the gist")
    c. I have shown that by increasing/decreasing the number of intermediate states that AIM and TI do agree within statistical uncertainty for Methane and Ethanol and I have provided possible explanations. See results below.
2. "Find a simple system where you can demonstrate that all methods converge to the same answer (within uncertainty)"

    a. I've chosen Methane although I have shown that it is possible for Ethanol and several other molecules.
        i. [A Benchmark Test Set for Alchemical Free Energy Transformations and Its Use to Quantify Error in Common Free Energy Methods]
    b. The paper "A Benchmark Test Set for Alchemical Free Energy Transformations and Its Use to Quantify Error in Common Free Energy Methods" makes a detailed argument to why this may not be possible due to hard limitations in computational resources.
3. "When testing a system, the first thing to do should be to generate a plot comparing slope values between AIM and TI."
    a. I've made this part of my analysis and have written code to do this on several levels.
4. "Go back to comparing AIM to standard methods as a function of time (10 ps, 50 ps, 100 ps, ...)"
    a. I believe that this means that I should run the different time scales using the same mdp settings as I used for the chosen molecule. This would include using the same lambda schedule. However, one does not simply run longer simulations with just any intermediate states so I am testing more lambda states as well. See "A Benchmark Test Set for Alchemical Free Energy Transformations and Its Use to Quantify Error in Common Free Energy Methods".
    b. I am not sure that this requirement will provide new or useful information or what requirement this set of simulations will deliver. The methods described in "A Benchmark Test Set for Alchemical Free Energy Transformations and Its Use to Quantify Error in Common Free Energy Methods" may be more appropriate if more evidence is required for number 1 above.

Update: What have you done?
I've compiled the cpu version for GROMACS so I can run more simultaneous simulations across the cluster. I was unable to start a job so I contacted Benji and he found one of the nodes were down. I'm now able to run simulations using the 'volatile' partition which gives me access to 19 nodes.

I've also spent time trying to figure out solvation steps for mutations. I want to use the PMX server to generate hybrid protein structure and topologies for simulations of mutations such as Alanine to Valine. I just need to put the steps in place to automate the solvation, equilibration and production runs for expanded ensemble simulations using AIM.

I've spent time trying to find the proper lambda schedule for methane and ethanol. I increased the number of intermediate values between 0.5 and 1.0 in order to allow the free energy to become independent of the number of states which should allow AIM and TI results to agree within statistical precision.

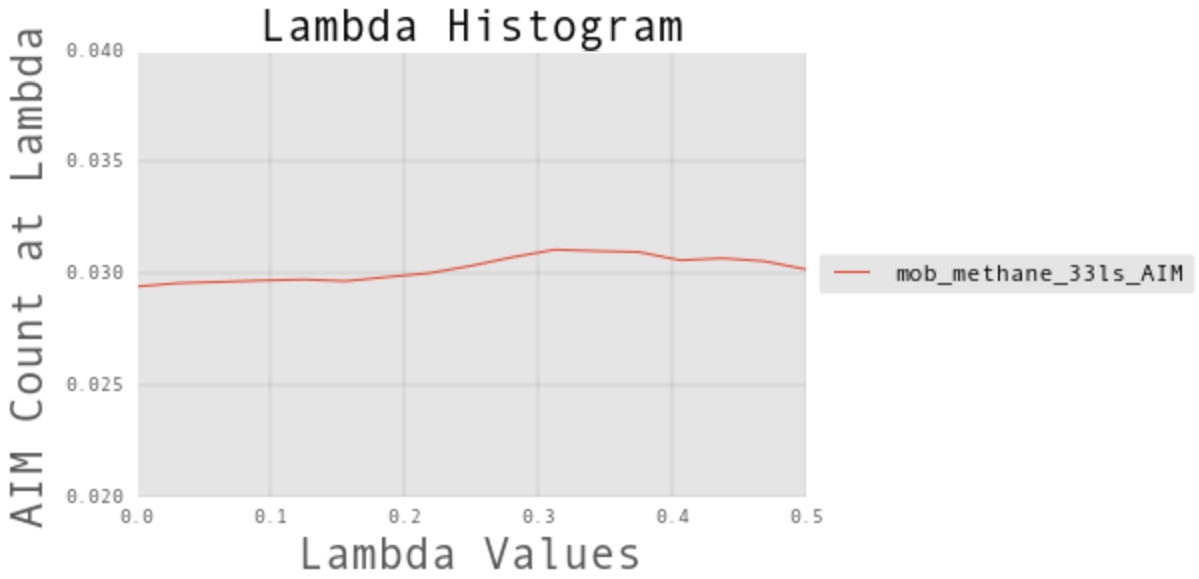
Methane
Using 33 lambdas and 100ps per lambda, for the methane molecule:

Using the alchemical analysis without removing 50% of the output for TI
    AIM-CUBIC = 10.830 +- 0.021 kJ/mol
    TI-CUBIC = 10.818 +- 0.022 kJ/mol

When removing 50% for TI
    TI-CUBIC = 10.751 +- 0.452

The two values are within statistical uncertainty of each other in both cases.

The histogram is flat (notice the y axis range is very small):



Ethanol
Using 33 lambdas and 100ps per lambda, for the ethanol molecule:
Using the alchemical analysis without removing 50% of the output for TI

AIM-CUBIC: 19.907 +- 0.049
TI-CUBIC: 19.854 +- 0.519

The two results are within statistical uncertainty of each other.

This lambda schedule isn't perfect but we see that, with enough intermediate states, AIM and TI are in agreement within statistical uncertainty and longer simulations aren't necessary except for further convergence.

Current Problem: Where are you stuck?
Possible resolutions: What are you planning to do?

Date: 03/26/2018
Current Goal(s): What are you doing? Comparing solvation free energy calculations using TI and AIM
Update: What have you done? I collected the FreeSolv https://github.com/MobleyLab/FreeSolv database of solvation free energies and grabbed a few of the simpler molecules and ran a 1ns per lambda simulation on each. The results are in the following table. I believe the results from the Mobley group are for 2.5 ns per lambda but I wanted to run quick sims to see AIM versus TI.

name	experimental value (kcal/mol)	experimental uncertainty (kcal/mol)	Mobley group calculated value (GAFF) (kcal/mol)	calculated uncertainty (kcal/mol)	AIM-CUBIC(kcal/mol)	TI-CUBIC(kcal/mol)

hydrogen sulfide	**-0.7**	**0.6**	**-1.14**	**0.01**	**-1.280593**	**-1.301890**
iodomethane	**-0.89**	**0.6**	**-0.64**	**0.02**	**-0.981956**	**-0.958517**
ammonia	**-4.29**	**0.6**	**-4.02**	**0.01**	**-4.258739**	**-4.081314**
methane*	**2**	**0.2**	**2.45**	**0.01**	**2.293890**	**2.297092**
ethanol**	**-5**	**0.6**	**-3.39**	**0.02**	**-4.981603**	**-4.886008**

*I have run a separate simulation for Methane using the topology and gro file from the Bevan labs tutorial and achieved a result even closer to the experimental value with AIM and TI.

AIM-CUBIC: 2.001566
TI-CUBIC: 1.969023

**The ethanol AIM and TI results are the averages of 5 simulations from my previous simulations. The ethanol file provided by Mobley failed with an error in the topology.

The difference between all of these results is going to be in the initial topology and lambda configuration.

I think this will be enough to prove the viability of AIM as a research tool. Also, from http://www.alchemistry.org/wiki/Thermodynamic_Integration , the reason TI doesn't match AIM is because:

"Although TI is one of the simplest free energy methods to analyze, it also suffers from some drawbacks that need to be carefully avoided. For instance, if the curvature of dU/dλ is large, the bias introduced by discrete λ states becomes significant. So when using TI it is very important that researchers verify that they have gathered data from sufficient numbers of states, such that the free energy becomes independent of the number of states to within statistical precision."

I have been trying to show this for some time because the difference in AIM vs. TI has been bothering me. See Hypothesis in update 03/17-18/2018 and conversations in Slack from February 25. AIM reduces this bias by spending more time on critical points in the curvature of the hamiltonian.

Current Problem: Where are you stuck? If we choose quantity over quality, I can run a few more of the molecules from FreeSolv and compare AIM's results instead of choosing one molecule and attempting to run longer simulations which may not improve the comparisons if I don't have the right lambda schedule.
Possible resolutions: What are you planning to do? I can do the same for mutations taken from the PMX web server for simple mutations. They already have the topology files created.

Date: 03/17-18/2018
Current Goal(s): What are you doing? Analyzing fixed lambda simulations
Update: What have you done? I don't think running longer sims is the solution in and of itself because running longer sims pushes the fixed lambda results further from AIM.

The problem relates to how large sample sizes can be problematic due to propagation of error.

i.e. one sample or independent sample t-tests:

n = sample population
$t = (\bar{x} - \mu 0) / (s / \sqrt{n})$

Rewriting this: $t = \sqrt{n} * (\bar{x} - \mu 0) / s$

Therefore, as n --> inf, *t* will also go to inf. This means a statistically significant difference is assured.

We can also express this in terms of variation in sum of squares (where effect size is exaggerated as sample size increases) anytime we are means testing.

Hypothesis
Averaging over longer simulations (increasing the sample size) in a state space that isn't dense enough (non-representative) to fully describe the state function propagates the error (bias). What I see is that by not increasing the density of states (the lambda density) we aren't able to achieve the required 'smoothness' of the function in regions of high variation. Averaging over longer sims of an insufficient state space simply propagates the error. Thus, the t-score for TI simulations will increase as time per lambda increases.

Test
The t score is a ratio between the difference between two groups and the difference within the groups. The larger the t score, the more difference there is between groups. The smaller the t score, the more similarity there is between groups.

Comparing AIM-CUBIC to TI-CUBIC for 3 simulations with 21 lambdas, for a given time per lambda, we look at the t-score as time per lambda increases:

TI-CUBIC: This shows that TI's t-score is increasing as time per lambda increases.

('500ps-1ns',(t-score=0.32806155017039801, pvalue=0.7594911215031066))
('500ps-5ns',(t-score=0.13875971913798063, pvalue=0.89658204453584545))
('500ps-10ns',(t-score=0.82993703522413664, pvalue=0.47040290795652273))

AIM-CUBIC: This shows that AIM's t-score is decreasing as time per lambda increases.

('500ps-1ns',(t-score=-1.6025963691391512, pvalue=0.18624695496962021))
('500ps-5ns',(t-score=-2.2282725006244481, pvalue=0.090005308578132548))
('500ps-10ns',(t-score=-2.7573395848079336, pvalue=0.098109910595888389))

Therefore, propagation of error in fixed lambda simulations is increasing as time per lambda increases. This means a statistically significant difference is assured as time per lambda increases for fixed lambda simulations. This should be verified by someone other than me.

Further:
Comparing T-Test of AIM of 21 lambdas versus TI of 21 lambdas
(t-score=2.7955717097498898, pvalue=0.051834834053082791)

Comparing T-Test of AIM of 21 lambdas versus TI of 28 lambdas
t-score=0.89416653767795773, pvalue=0.4245192831337285

The decrease in t-score of AIM 21 versus TI 28 suggests that TI 28 is more similar to AIM 21 than TI 21.

This update and the next are attempts to find the proper lambda schedule that allows fixed lambda simulations (TI) to converge to the same value as AIM.

Focused on:

- Efficiency and convergence of free energy methods, rather than accuracy
    - Accuracy is a function of the force field
- If the methods are biased or incorrect, or the simulations are simply not converged, then any conclusion about the underlying models will likely be wrong
    - https://static1.squarespace.com/static/530562f9e4b06fd3c041e221/t/57056d1be707eb64fd5cc05d/1459973403785/alchemical+free+energy+calculations-+ready+for+prime+time.pdf
- Identifying AIM as one of the most efficient free energy methods and proving that AIM improves sampling over other free energy methods

Hypothesis:
If running longer simulations of the same lambda schedule were the solution, then the uncertainties would significantly decrease for subsequently longer simulations.

We do see significant decrease (a negative increase) in uncertainty for the method TI-CUBIC and the 21 lambda schedule:

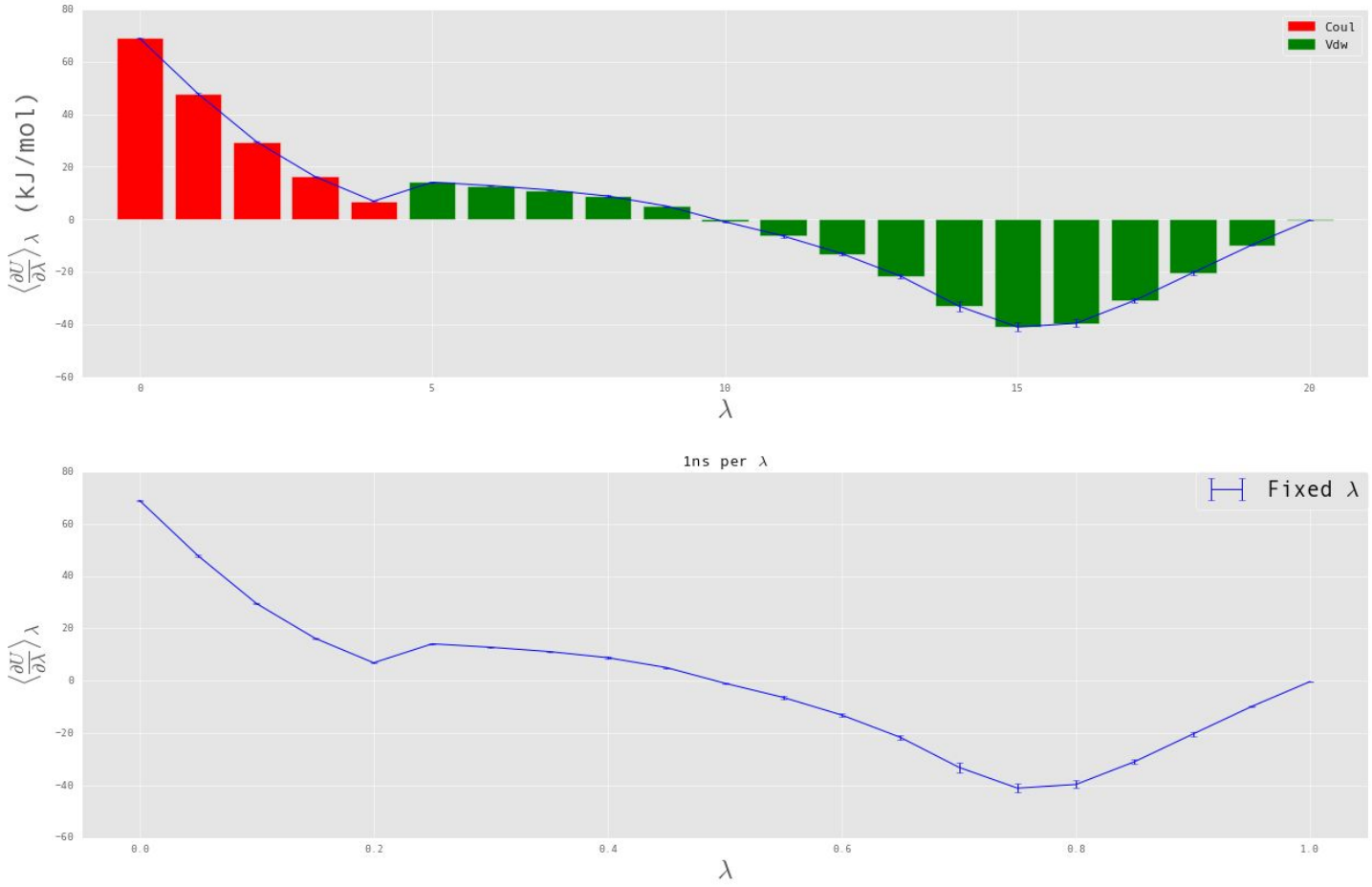
Time per lambda, Average, uncertainty
('100ps', 20.470497457338247, 0.44103733608714168)
('250ps', 20.255424864477664, 0.52998781398489969)
('500ps', 20.512197912369601, 0.11197017093953696)
('1ns', 20.540307550515063, 0.14187867118126288)
('5ns', 20.495999232422793, 0.13790076725511063)
('10ns', 20.477098851121251, 0.060530140192755211)

% Increase in uncertainty between times.
100ps - 250ps 20.1684688845 %increase
250ps - 500ps -78.8730668923 %increase
500ps - 1ns 26.7111320727 %increase
1ns - 5ns -2.80373638478 %increase
5ns - 10ns -56.1060163786 %increase

There is a significant decrease in the uncertainty (-increase) from 5ns per lambda to 10ns per lambda and greater than 5% decrease in the average from 5ns per lamba to 10ns per lambda, -0.0922149785780785% increase. It seems that more simulations are necessary. There are three routes that I could take:

1. Run longer simulations, i.e. increase the time per lambda
2. Run more simulations of the same time per lambda, i.e. average over more runs
3. Break up the lambda schedule to increase the density of lambdas in high variance areas

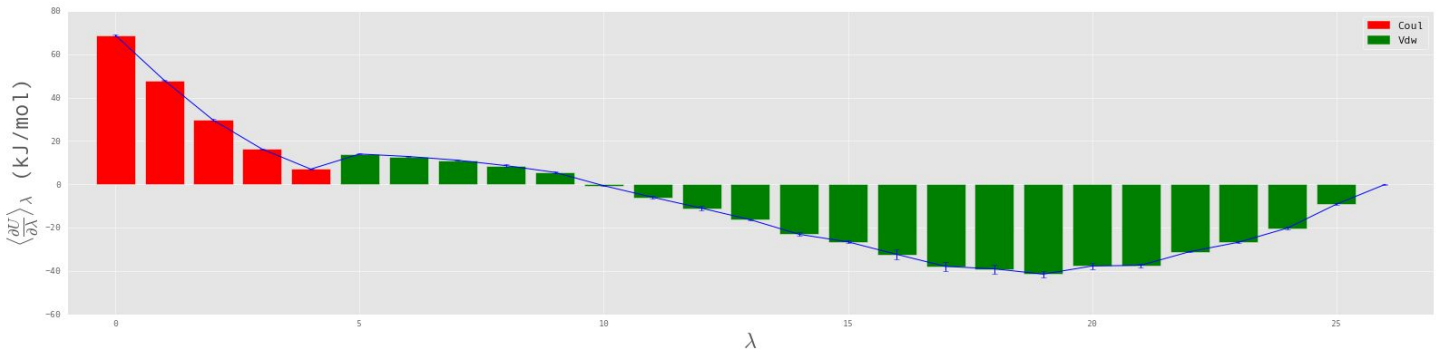
The paper “Guidelines for the analysis of free energy calculations” suggests a plot of the delta G components to help determine the next best strategy.

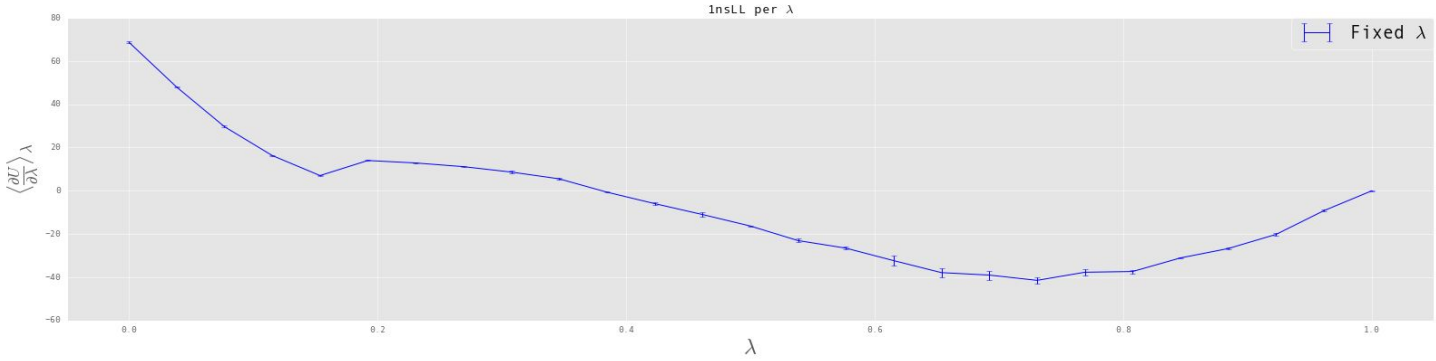


Fixed lambda simulation with 1ns per lambda and 21 lambdas averaged over 5 runs. Different delta G components are shown in distinct colors: red is the electrostatic DG component (k indices 0 – 5), while green is the van der Waals DG component (k indices 5–21). Alternate interpolation schemes disagree most around k points where the slope of <dh/dl> changes suddenly, so these are regions where a more dense lambda spacing is desirable. Error bars in the line indicate uncertainty over the average.

This also suggests that I need to increase the lambda density.

28 Lambdas, 1ns per lambda for testing, average over 3 runs
coulLambdas =
[0.0,0.2,0.4,0.6,0.8,1.0,1.0,1.0,1.0,1.0,1.0,1.00,1.0,1.000,1.00,1.000,1.0,1.000,1.00,1.000,1.0,1.000,1.00,1.000,1.00,1.00,1.00]
vdwLambdas =
[0.0,0.0,0.0,0.0,0.0,0.0,0.1,0.2,0.3,0.4,0.5,0.55,0.6,0.625,0.65,0.675,0.7,0.725,0.75,0.775,0.8,0.825,0.85,0.875,0.90,0.95,1.00]





1ns per lambda simulation with 28 lambdas. The coverage seems better but still need to increase lambda density between 0.6 and 0.8.

Current Problem: Where are you stuck? Not stuck.
Possible resolutions: What are you planning to do? Running a 1ns per lambda sim with greater lambda density

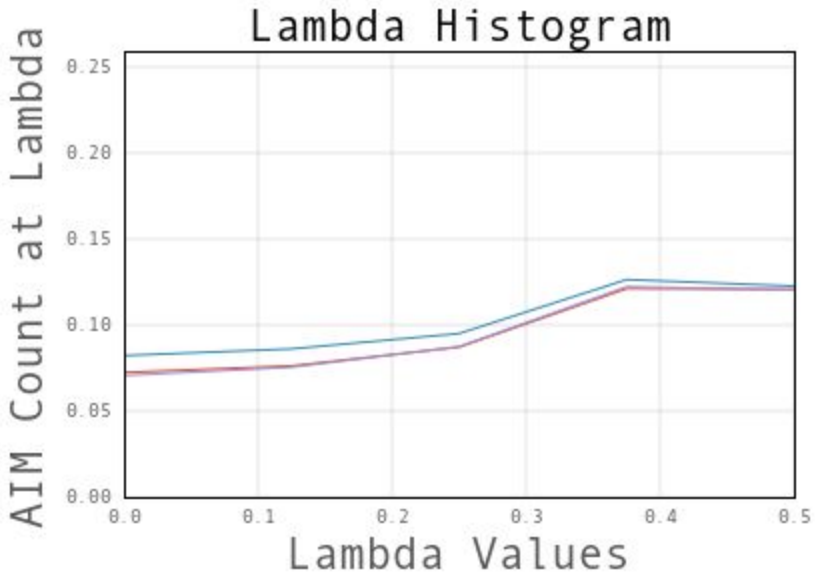
---

Date: 03/13/2018
Current Goal(s): What are you doing? Testing AIM vs TI
Update: What have you done? I went back to the original schedule of 9 lambdas and ran 3 fixed lambda simulations and 3 expanded ensemble simulations at 1ns per lambda.

Method, Average, Standard Deviation(Uncertainty)
'TI-CUBIC', 20.490866484752704, +/- 0.094319286699329
'AIM-CUBIC', 20.415690947804116, +/- 0.1029387261864715

20.490866484752704 - 20.415690947804116 = 0.07517553694858847

TI and AIM are within statistical uncertainty of each other.



Current Problem: Where are you stuck? The histogram suggests longer simulations.
Possible resolutions: What are you planning to do? Run longer simulations of 10ns per lambda
---
Date: 02/11/2018
Current Goal(s): What are you doing? Convincing you that my code is correct.
Update: What have you done? I've compared my code to alchemical_analysis which uses pymbar to calculate free energies. I did not write the alchemical_analysis script. It is open source on github and related to the pymbar project. I also found out why my code wasn't working in GROMACS 5.0.6: https://redmine.gromacs.org/issues/2264 ← was corrected in GROMACS 2016 which is what I'm using now. This implies that we need to upgrade to GROMACS 2016 if doing any expanded ensemble simulations.

The code that I use to calculate AIM is the code that pymbar uses to calculate TI after collecting the averages so let's compare the outputs for TI.

alchemical_analysis collects the xvg files and calculates the averages of each lambda, storing them in a unitless 2d vector called ave_dhdl.

As input for my calculations, I don't use the xvg files. I use averages that I collect with awk in bash because the files for 5ns are too large for alchemical_analysis.

The results presented here are for 100ps simulations where 50% of the xvg file was thrown away and using the alchemical_analysis flag of -i 0 which turns off time series analysis.

For a 100ps per lambda simulation:

alchemical_analysis gives these results for the 5 runs:
0 20.637173
1 20.503738
2 21.419590
3 20.222947

4 20.961346

average = 20.748959

And my code calculates TI as:
0 20.634195
1 20.500763
2 21.416488
3 20.220040
4 20.958295

average = 20.745956

All of the above applies to TI-CUBIC as well.
alchemical_analysis TI-CUBIC
0 20.463942
1 20.212222
2 21.184393
3 20.021433
4 20.654712

average = 20.507340

My code TI-CUBIC
0 20.460990
1 20.209285
2 21.181324
3 20.018556
4 20.651705

average = 20.504372

For a single run of 100ps using AIM, I "injected" the AIM coulomb dhdl averages and vdw dhdl averages into alchemical_analysis without changing anything else and used it's internal TI machinery to calculate the free energy.

alchemical_analysis returned:
non-cubic: 21.536
cubic: 21.251

The values from my python code for the exact same dhdl averages:
AIM AIM-CUBIC
0 21.536354 21.250975

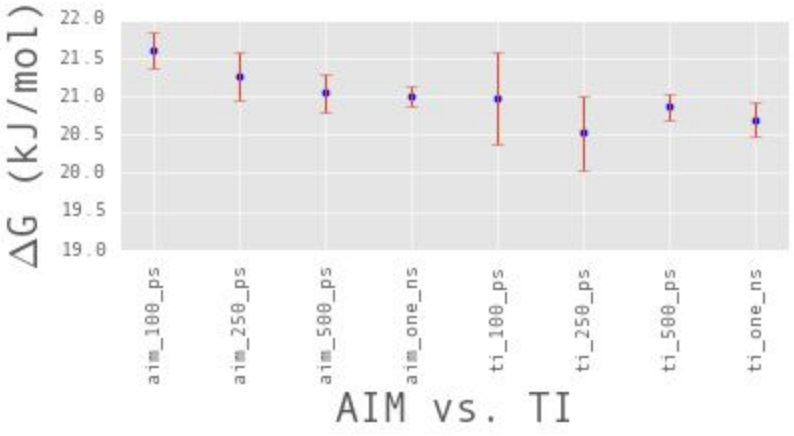
Current Problem: Where are you stuck?
Possible resolutions: What are you planning to do? I have to run 10ns per lambda simulations.

Date: 02/05/2018

Current Goal(s): What are you doing?
Update: What have you done?

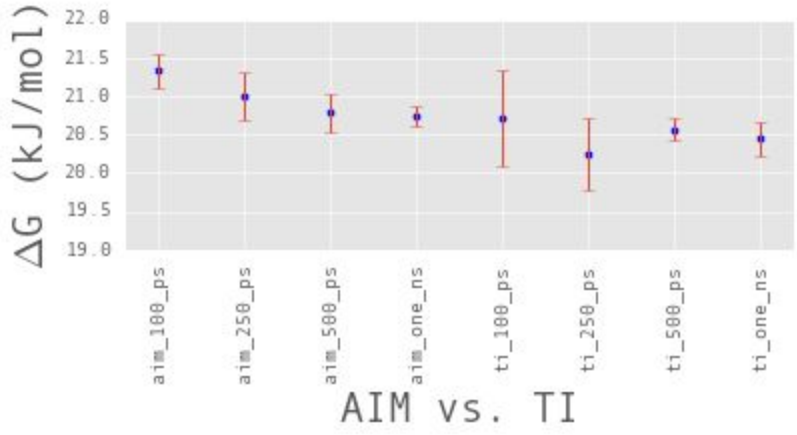
Using pymbar with flag -i 0 for doing the calculations.

AIM vs TI using pymbar for TI non-cubic approximation



	aim_100_ps	aim_250_ps	aim_500_ps	aim_one_ns	ti_100_ps	ti_250_ps	ti_500_ps	ti_one_ns
count	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000
mean	21.596337	21.257890	21.049397	21.009589	20.973714	20.519048	20.861742	20.693462
std	0.256998	0.339839	0.276483	0.143479	0.656423	0.544755	0.187188	0.248293
min	21.235398	20.818240	20.820069	20.878792	20.491421	19.892542	20.615230	20.333498
25%	21.522641	21.180809	20.928179	20.898714	20.500261	20.393348	20.754098	20.552469
50%	21.536354	21.244168	20.937078	20.951040	20.549199	20.411147	20.889837	20.770803
75%	21.807864	21.276285	21.037159	21.112527	21.404976	20.503465	20.941539	20.878460
max	21.879429	21.769945	21.524498	21.206870	21.922715	21.394739	21.108004	20.932078

AIM vs TI using pymbar for calculation cubic approximation

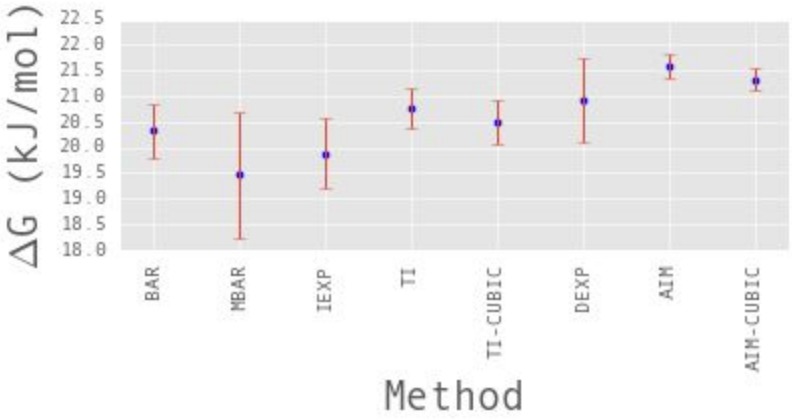


	aim_100_ps	aim_250_ps	aim_500_ps	aim_one_ns	ti_100_ps	ti_250_ps	ti_500_ps	ti_one_ns
count	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000
mean	21.326875	20.992248	20.789040	20.749918	20.707478	20.245945	20.564376	20.446514
std	0.252575	0.350761	0.277922	0.143086	0.700488	0.530412	0.161711	0.253210
min	20.988321	20.534885	20.556692	20.622507	20.125827	19.668092	20.367030	20.071342
25%	21.239581	20.910350	20.666305	20.638577	20.202830	20.080900	20.443113	20.308906
50%	21.250975	20.976791	20.678563	20.691132	20.305917	20.157859	20.593339	20.546672
75%	21.552236	21.022475	20.777551	20.846041	21.225989	20.208027	20.645310	20.624450
max	21.603259	21.516739	21.266090	20.951334	21.676828	21.114849	20.773091	20.681202

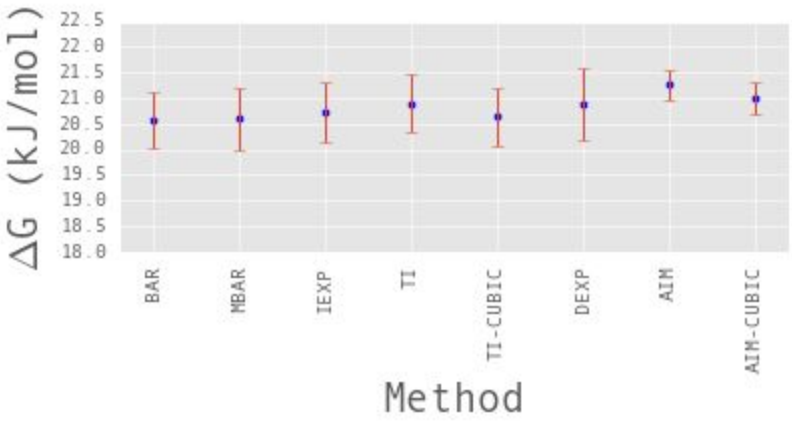
Current Problem: Where are you stuck? 5ns per lambda files are too large for memory because I used nstdhdl=1.
Possible resolutions: What are you planning to do?

Date: 02/05/2018
Current Goal(s): What are you doing? Finishing TI simulations for Ethanol in water
Update: What have you done? Compared the results for 5 runs of expanded ensemble with AIM versus fixed lambda simulations.

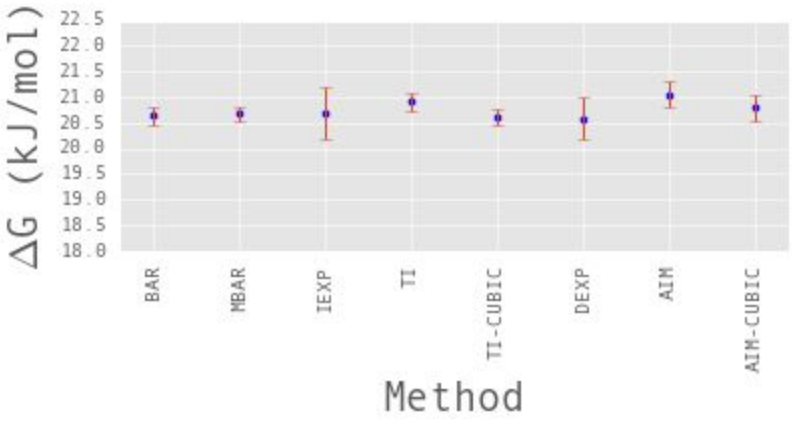
100ps per lambda



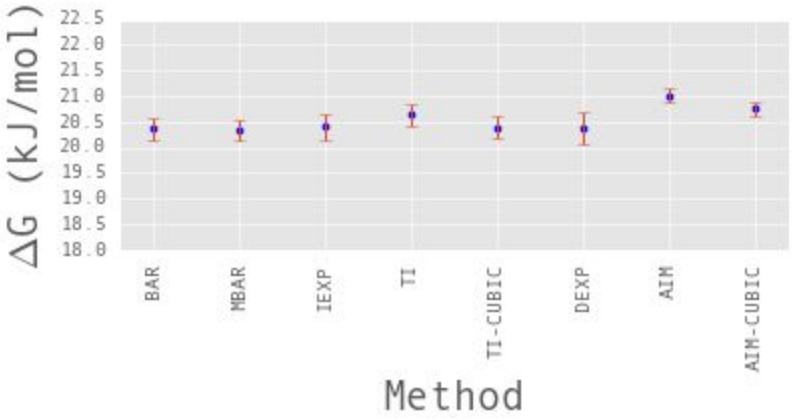
250ps per lambda



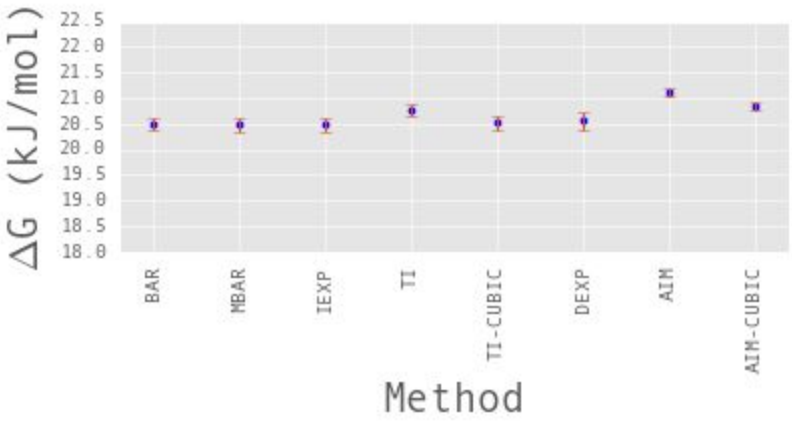
500ps per lambda



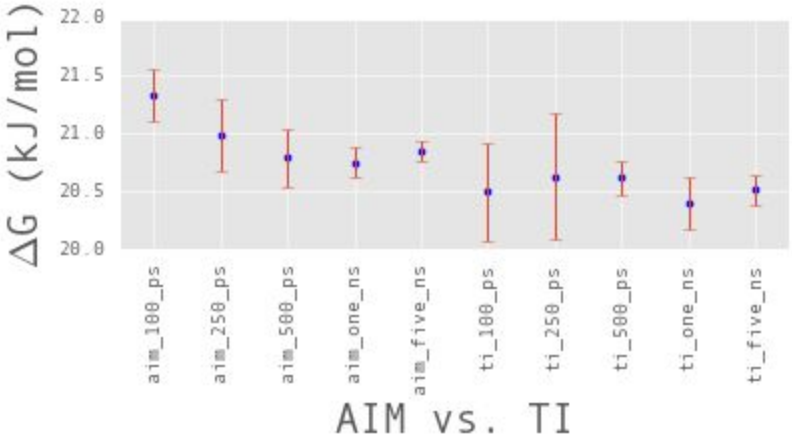
1ns per lambda



5ns per lambda



AIM-cubic vs. TI-cubic

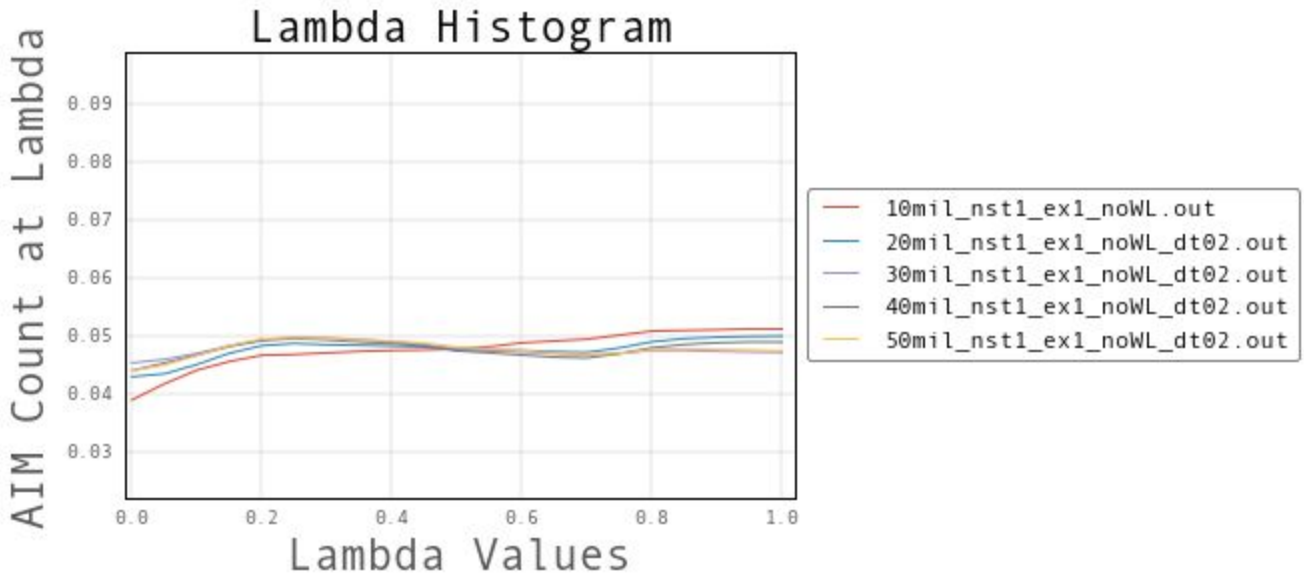


The experimental value for Ethanol solvation, given by http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.505.5752&rep=rep1&type=pdf , is -5.0 kcal/mol or -20.9 kJ/mol.

The value I obtained at 5ns per lambda is positive (opposed to negative) because the process I conducted for this simulation was the decoupling of ethanol, the reverse process (the introduction of uncharged ethanol into

water, thus the actual hydration energy of the process) corresponds to a negative ΔG. Assuming reversibility, the value due to AIM-cubic is in good agreement with the value obtained by the above referenced paper. (Assuming the referenced paper's reference is good)

Date: 01/16/2018
Current Goal(s): What are you doing? Testing for convergence
Update: What have you done? I've run simulations for AIM at 10mil, 20mil, 30mil, 40mil and 50mil steps.



If I print out the "flatness" between each simulation:

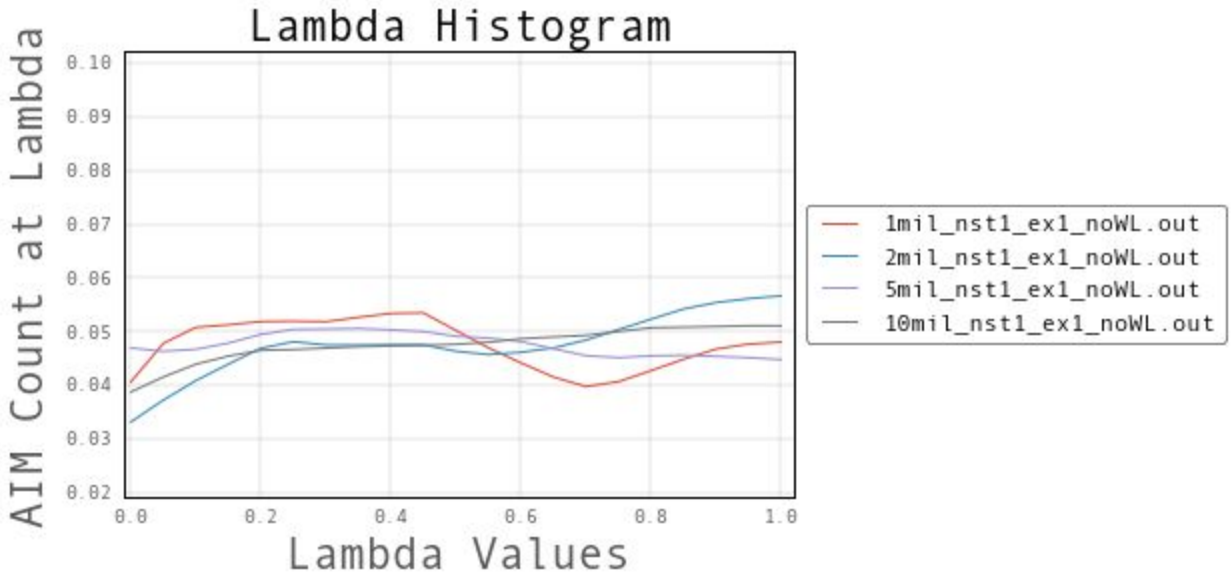
('10mil_nst1_ex1_noWL.out', ' ', "It's flat", 81.570930000000004)
('20mil_nst1_ex1_noWL_dt02.out', ' ', "It's flat", 89.980064999999996)
('30mil_nst1_ex1_noWL_dt02.out', ' ', "It's flat", 95.022970000000001)
('40mil_nst1_ex1_noWL_dt02.out', ' ', "It's flat", 92.201707499999998)
('50mil_nst1_ex1_noWL_dt02.out', ' ', "It's flat", 92.140607999999986)

the number on the far right is the ratio of the minimum count over the average of counts for each value of steps.

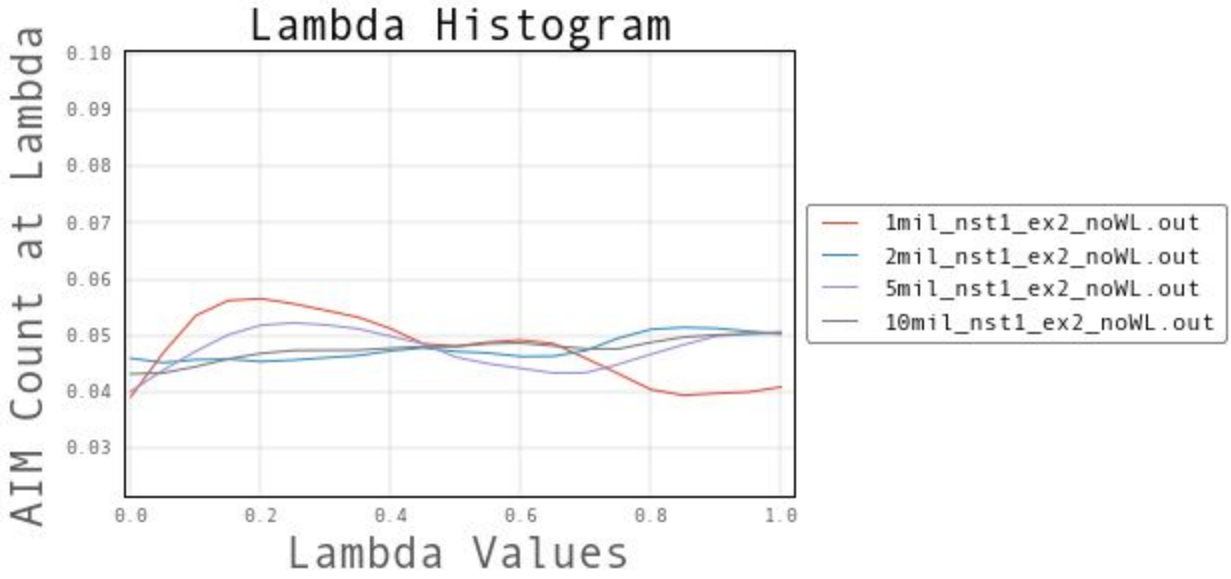
Current Problem: Where are you stuck?
Possible resolutions: What are you planning to do?

Date: 12/11/2018
Current Goal(s): What are you doing?
Update: What have you done?

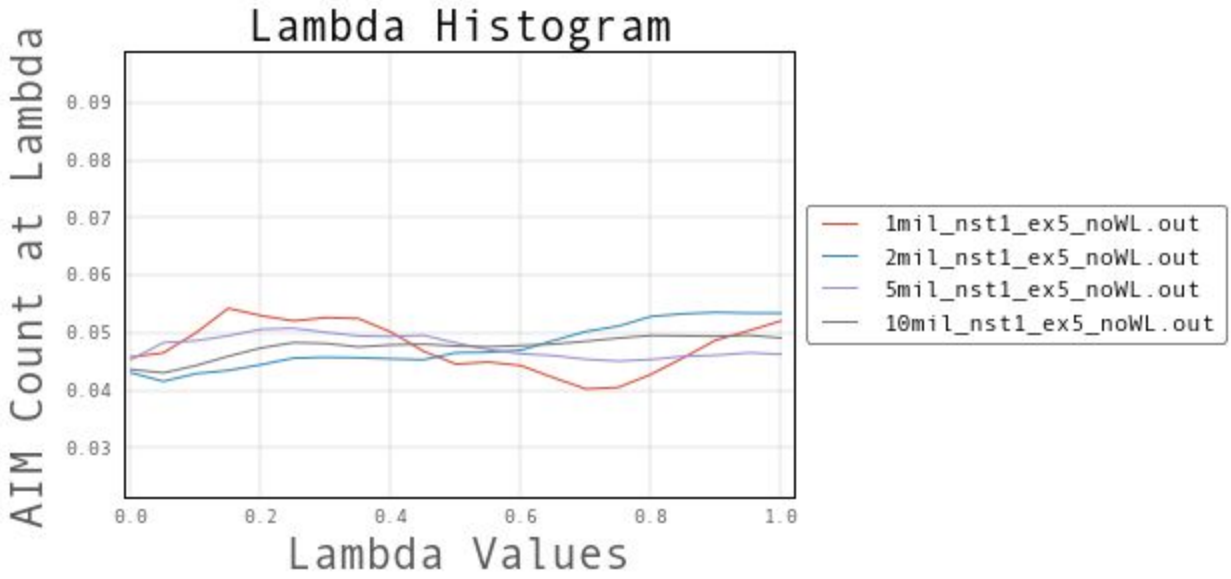
('1mil_nst1_ex1_noWL.out', ' ', "It's flat")
('2mil_nst1_ex1_noWL.out', ' ', 'Not flat')
('5mil_nst1_ex1_noWL.out', ' ', "It's flat")
('10mil_nst1_ex1_noWL.out', ' ', "It's flat")



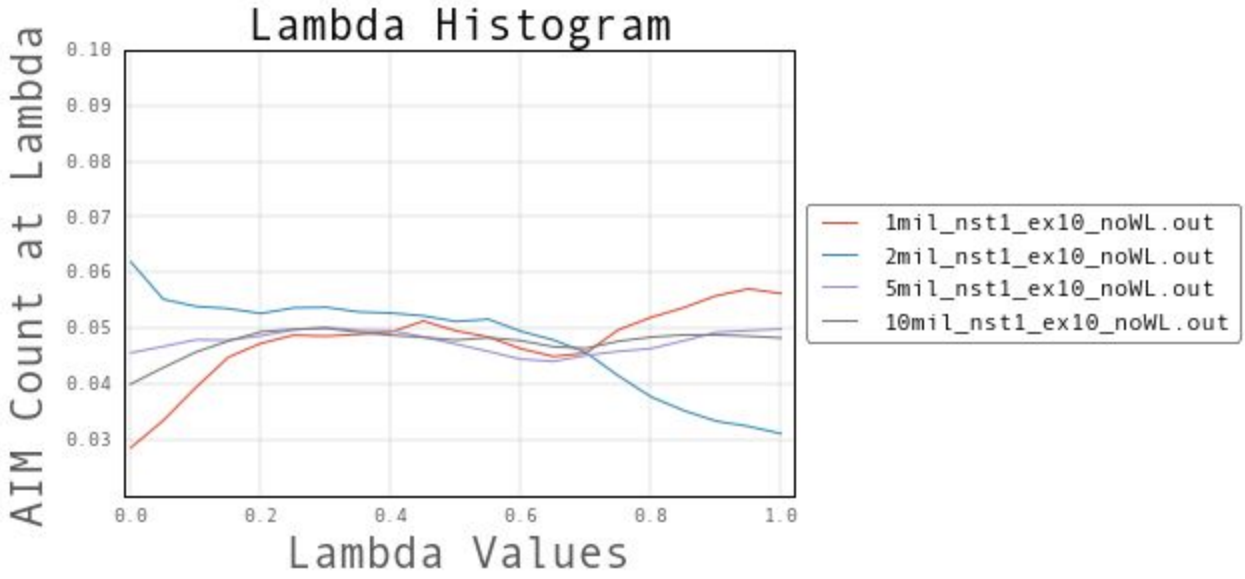
('1mil_nst1_ex2_noWL.out', ' ', "It's flat")
('2mil_nst1_ex2_noWL.out', ' ', "It's flat")
('5mil_nst1_ex2_noWL.out', ' ', "It's flat")
('10mil_nst1_ex2_noWL.out', ' ', "It's flat")



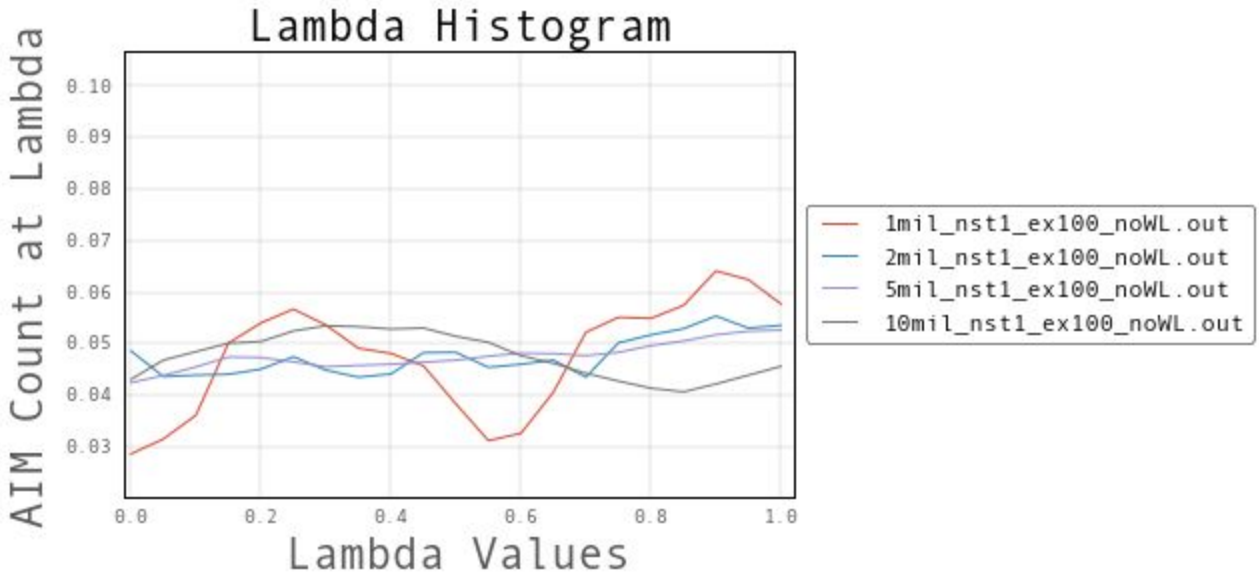
('1mil_nst1_ex5_noWL.out', ' ', "It's flat")
('2mil_nst1_ex5_noWL.out', ' ', "It's flat")
('5mil_nst1_ex5_noWL.out', ' ', "It's flat")
('10mil_nst1_ex5_noWL.out', ' ', "It's flat")



('1mil_nst1_ex10_noWL.out', ' ', 'Not flat')
('2mil_nst1_ex10_noWL.out', ' ', 'Not flat')
('5mil_nst1_ex10_noWL.out', ' ', "It's flat")
('10mil_nst1_ex10_noWL.out', ' ', "It's flat")



('1mil_nst1_ex100_noWL.out', ' ', 'Not flat')
('2mil_nst1_ex100_noWL.out', ' ', "It's flat")
('5mil_nst1_ex100_noWL.out', ' ', "It's flat")
('10mil_nst1_ex100_noWL.out', ' ', "It's flat")

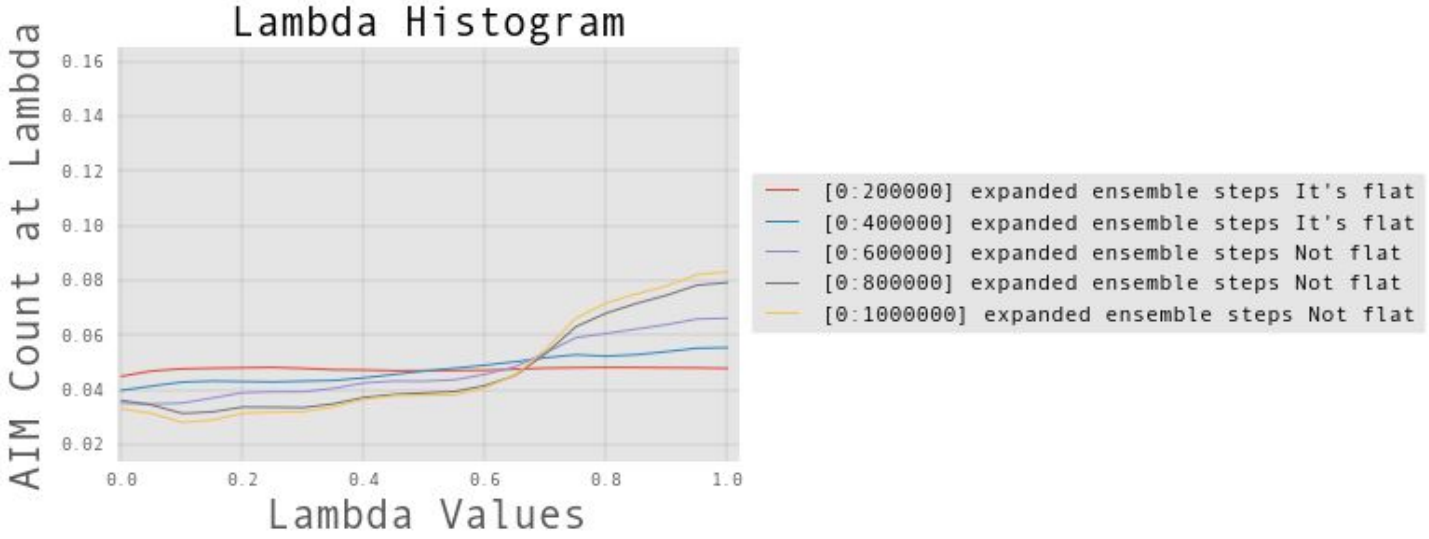


Current Problem: Where are you stuck?
Possible resolutions: What are you planning to do?

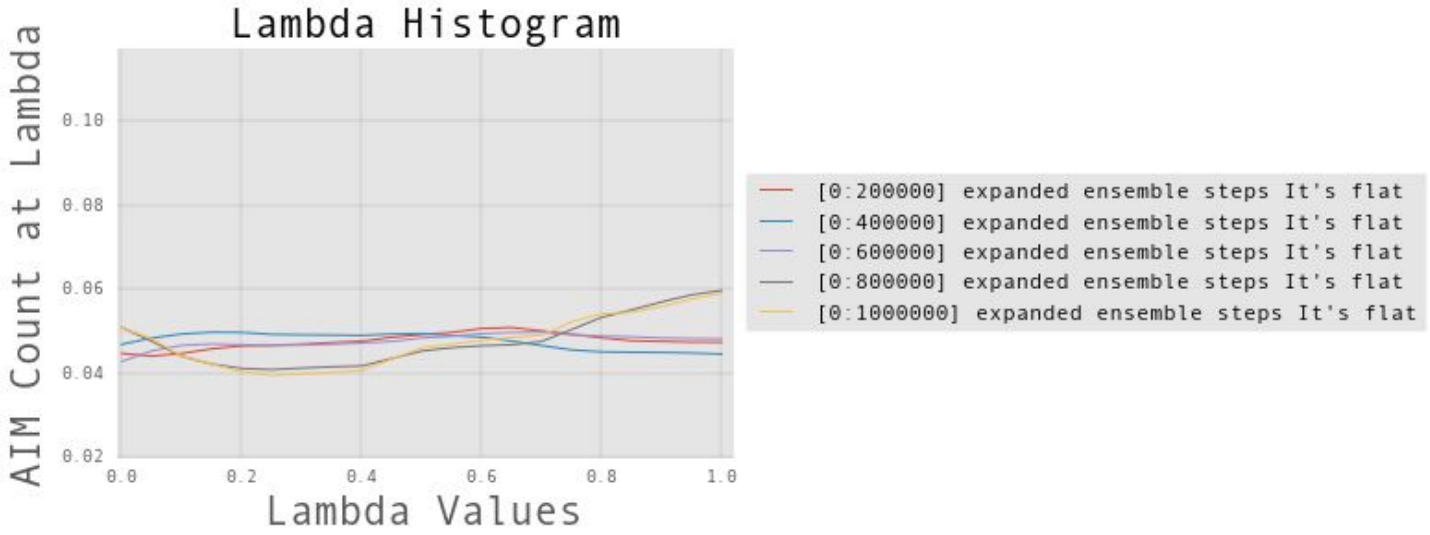
Date: 12/11/2017
Current Goal(s): What are you doing? Running simulations to test the outcomes of using different values for nstcalcenergy, nstdhdl, and nstexpanded.

Update: What have you done? I've run sims where nstcalcenergy = 1, nstdhdl=1 and nstexpanded = 2, 100

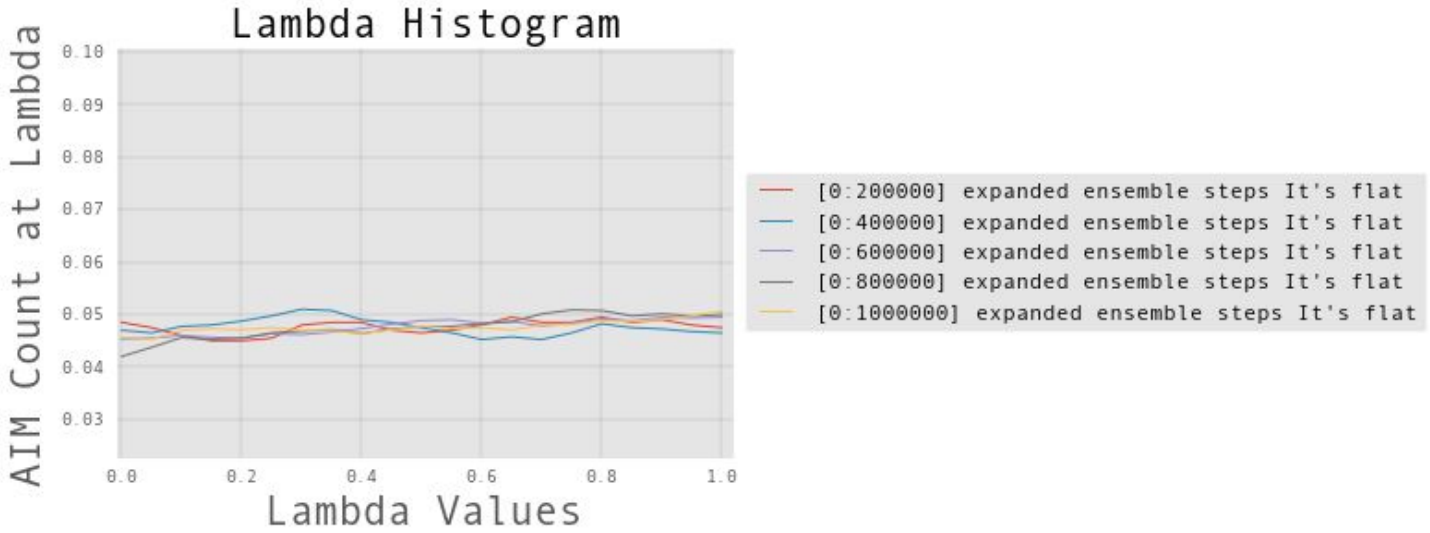
For comparison:
Nstexpanded = 1, nstdhdl=1, nstcalcenergy=1



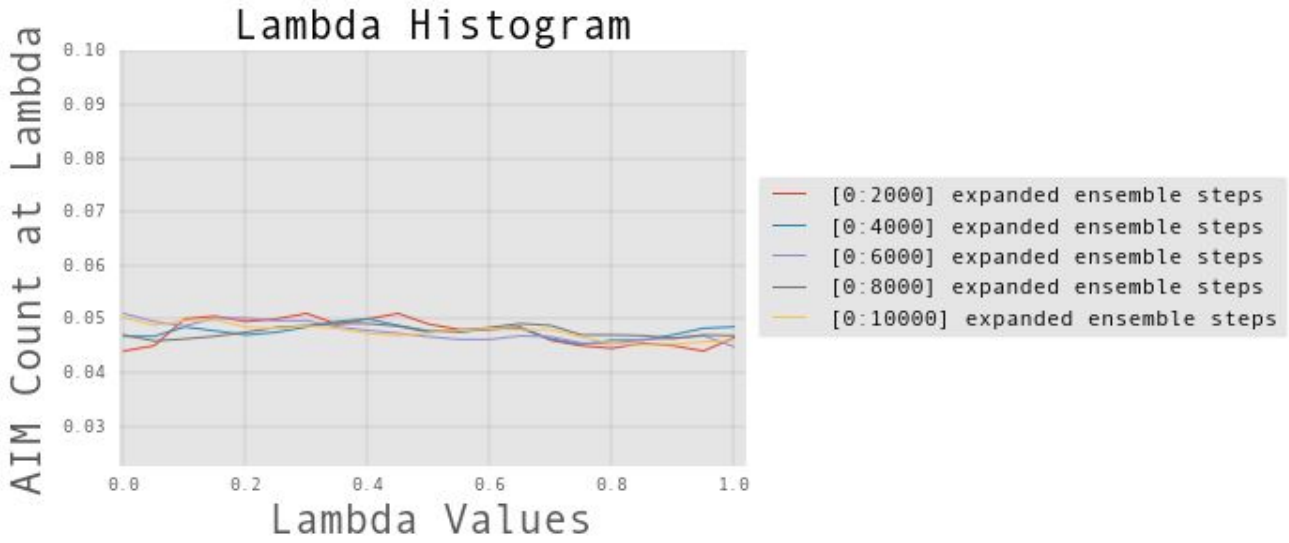
Nstexpanded = 2, nstdhdl=1, nstcalcenergy=1



Nstexpanded = 100, nstdhdl=1, nstcalcenergy=1



Nstexpanded = 100, nstdhdl=100, nstcalcenergy=100



Current Problem: Where are you stuck?
Possible resolutions: What are you planning to do?

Date: 12/03/2017
Current Goal(s): What are you doing?

Currently running a 100 ns simulation with nstexpanded = 100. The idea is that a 1 million step simulation with nstexpanded = 1 is not the same as a 1mil step sim with nstexpanded = 10 is not the same as a 1mil step sim with nstexpanded = 100 since there is time for equilibration between expanded ensemble steps when nstexpanded is greater than 1. I want to make a distinction between AIM and the simulation at large. I want to be completely sure that there is no question as to whether or not the code is correct.

Update: What have you done?

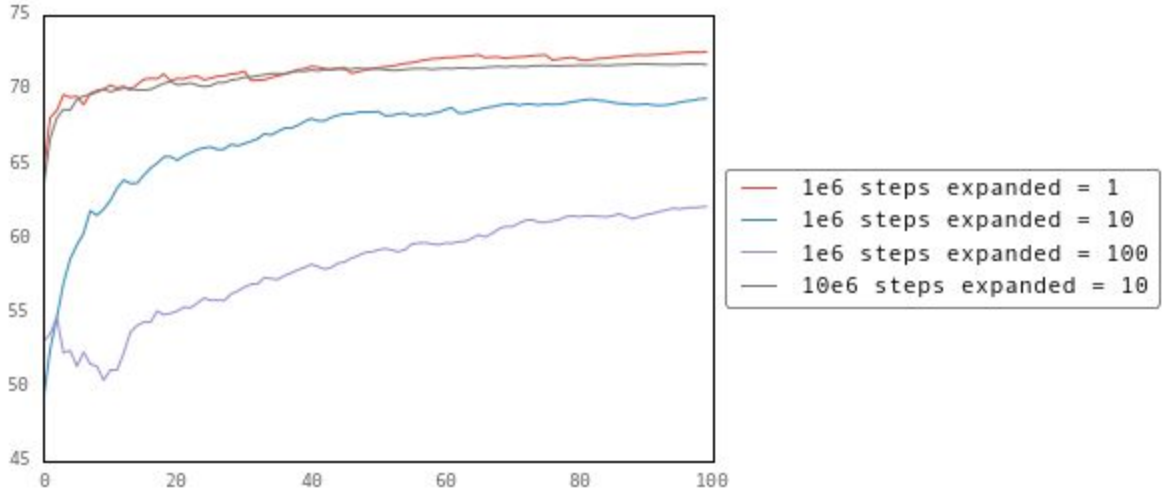
I have run 5 simulations.

1. 1 millions steps, nstexpanded = 1
2. 1 million steps, nstexpanded = 10
3. 1 million steps, nstexpanded = 100
4. 10 million steps nstexpanded = 10
5. 100 million steps nstexpanded = 100 (still running)

Note: nstexpanded is the number of integration steps bewteen attempted moves changing the system Hamiltonian in expanded ensemble simulations. This value must be a multiple of **nstcalcenergy**, but can be greater or less than **nstdhdl**. For this reason nstexpanded = nstdhdl = nstcalcenergy for these simulations.

Analysis: Histograms and Acceptance ratio.
We wanted to make sure the histogram is getting flatter and the acceptance ratio is increasing as the simulation goes longer. As the simulation goes on, the flatness and acceptance should increase because it should get easier to accept lambda trial. Because deltaF - deltaE should be getting closer to 0.

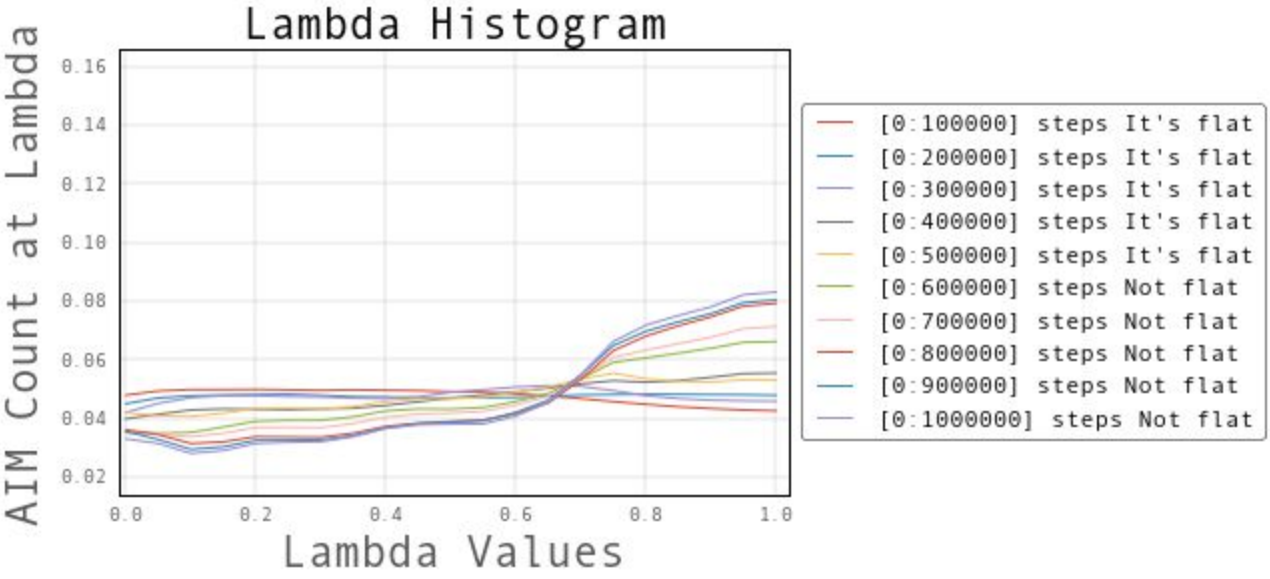
Acceptance ratio is increasing:



The x-axis can be thought of as a window length. Percent of total timescale works as well. The total steps are broken into “chunks”. The first chunk goes from 0:10000 for 1 million steps, then 0:20000, 0:30000… For 10 million steps is goes from 0:100000, 0:200000…

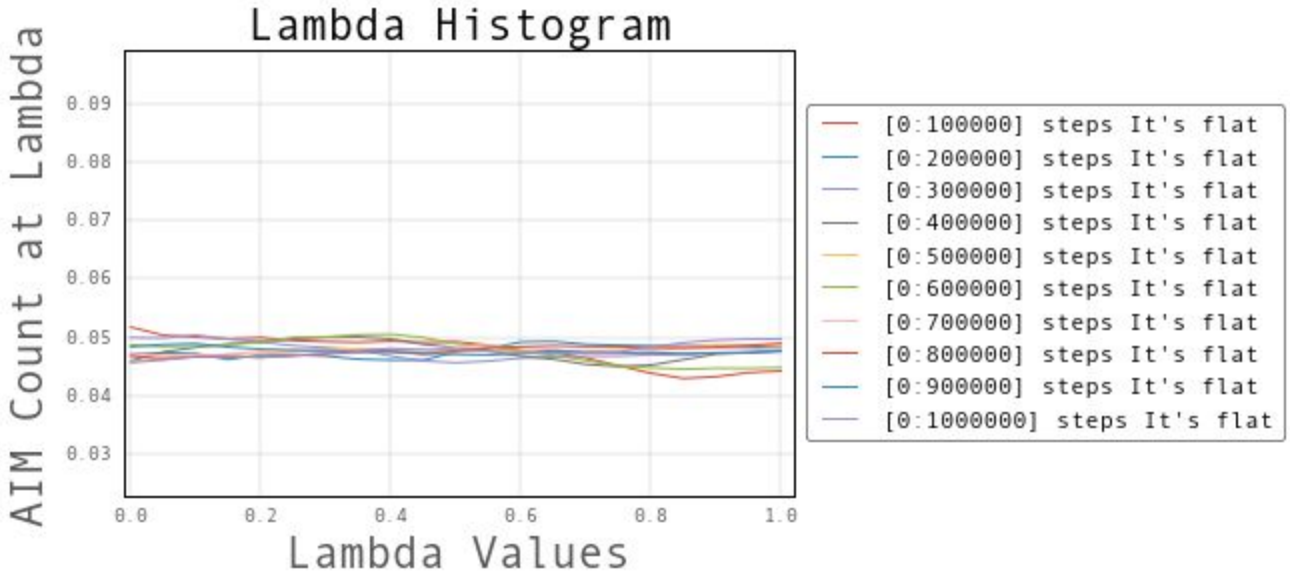
The acceptance ratio is calculated by incrementing a count whenever a lambda move is accepted and divided by the total number of steps. We multiply the result by one hundred for readability and to indicate that it is a probability score or liklihood ratio.

Flatness decreases if nstexpanded = 1 for 1 million steps:
Note: These are plots of increasing window size as indicated by the legend on the RHS.

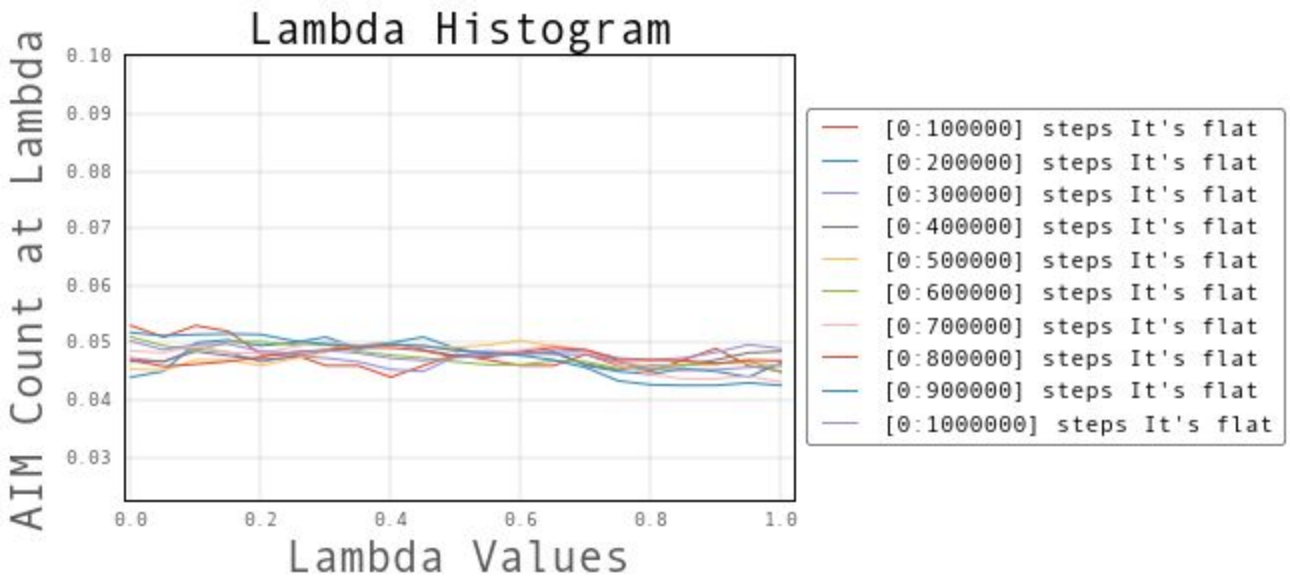


For 1 million steps, if nstexpanded is greater than 1, the histograms are “flat enough”
Note: These are plots of increasing window size as indicated by the legend on the RHS.

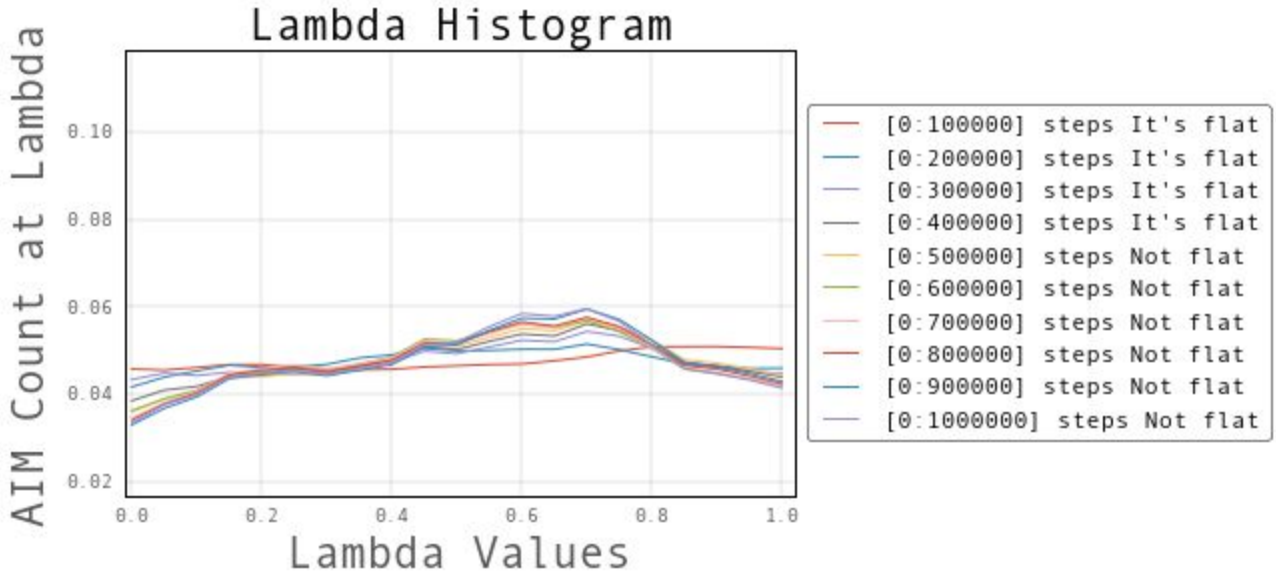
Nstexpanded = 10, 1 million steps

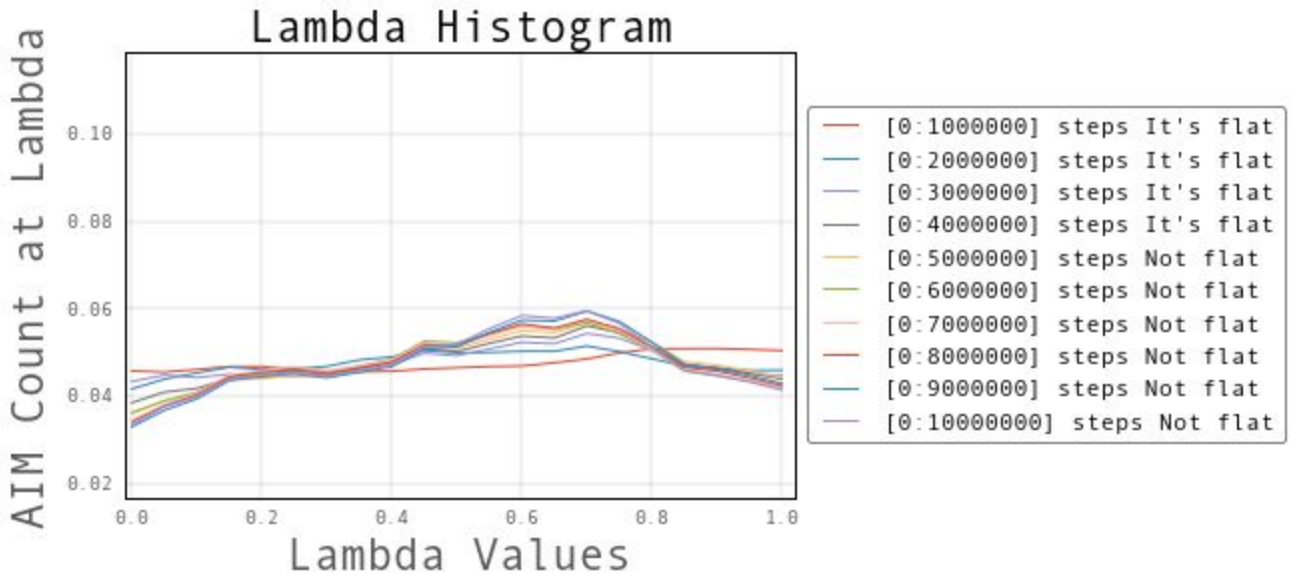


Nstexpanded = 100, 1 million steps

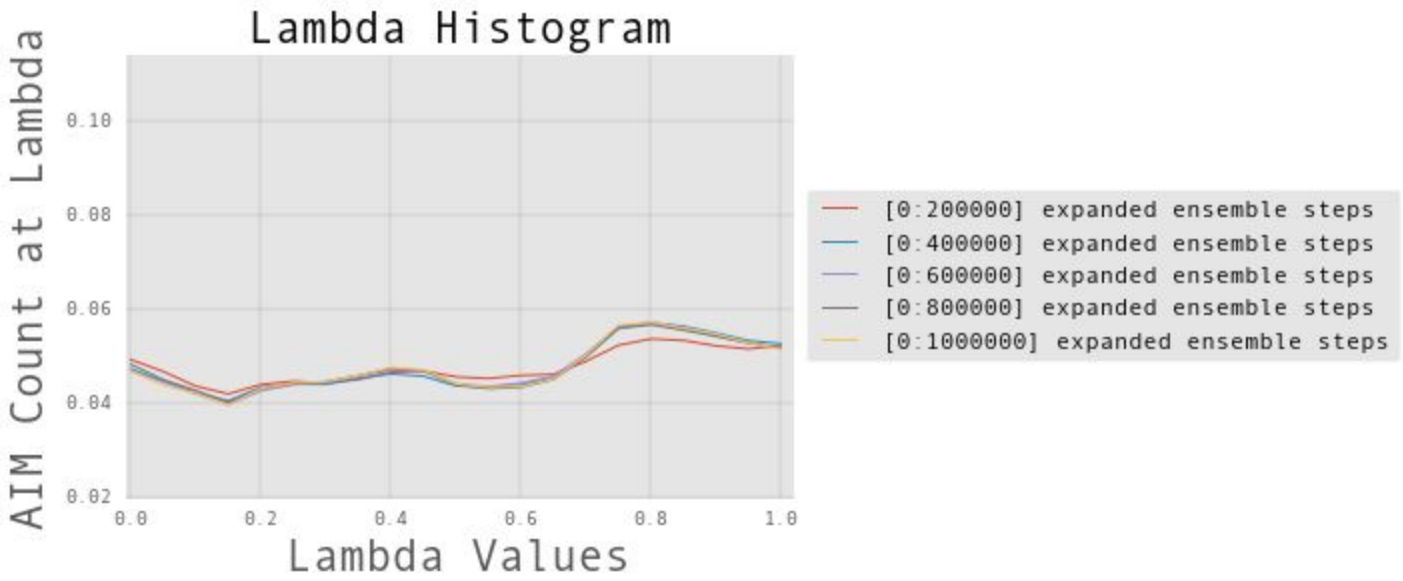


Flatness decreases if nstexpanded = 10 for 10 million steps:
Nstexpanded = 10, 10 million steps

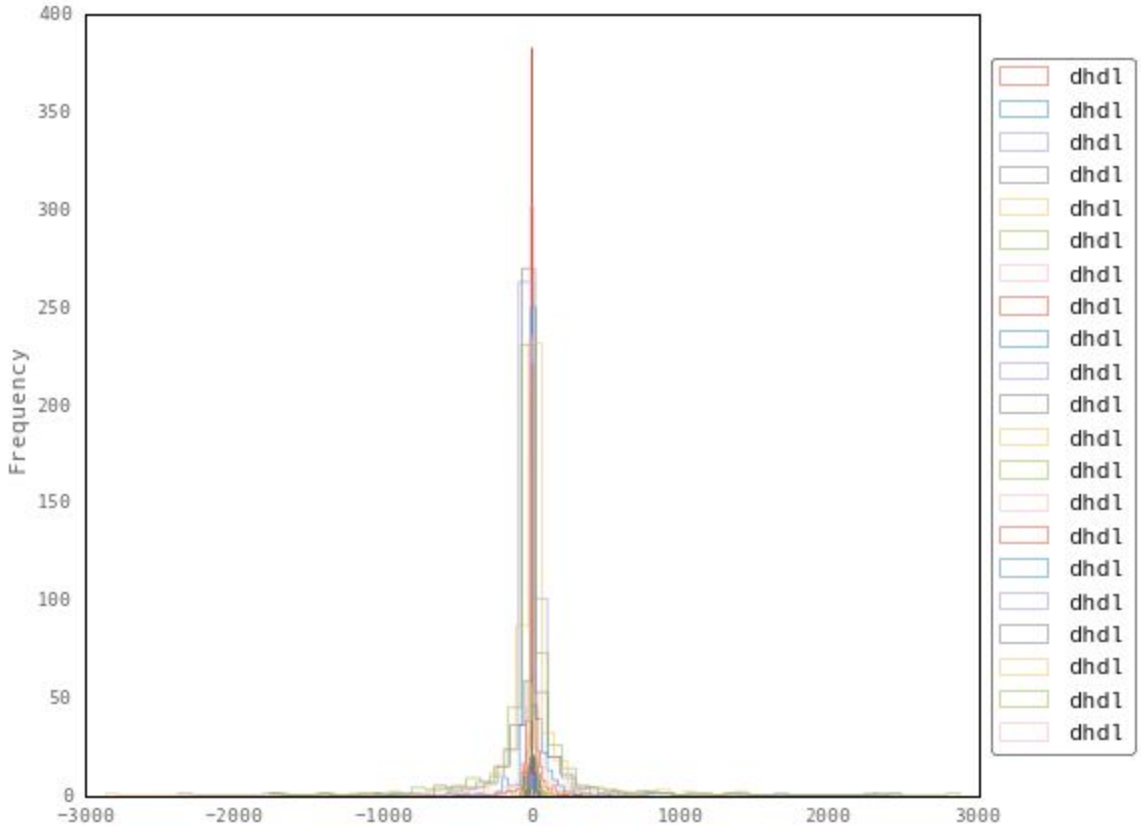




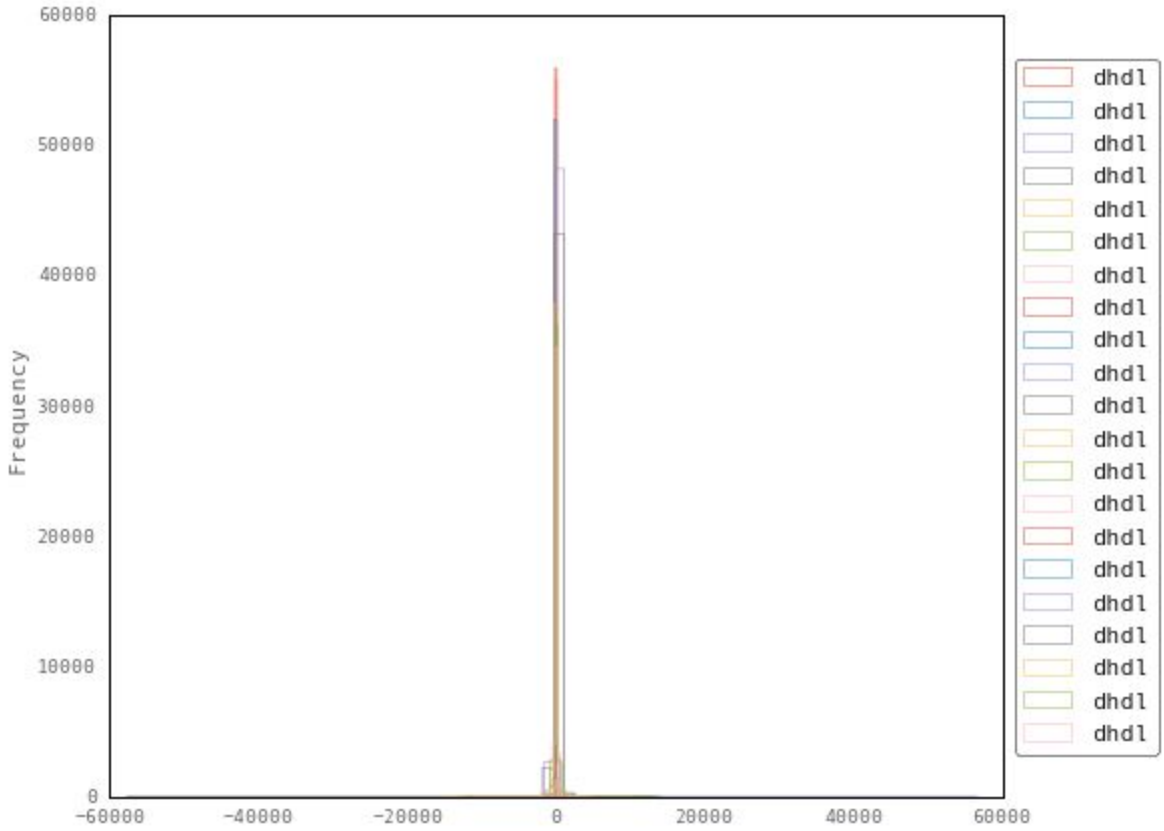
Nstexpanded = 100, 100 million steps



Additionally, for the 1 million steps with nstexpanded = 100, I plotted the distribution of dh/d(lambda) for each value of lambda. The legend didn't work but it should be each lambda in sequential order so we can see the overlap. The x axis has units of kJ/mol.



Same for 10 million steps with nstexpanded = 10



So what does this mean? In general, histograms from MD simulations are used to indicate some kind of overlap in energies, positions, etc.. Here, in order to obtain a reliable estimate of ΔG, we need adequate sampling within each window (based on ∂H/∂λ distributions). Is this something we want to include in the analysis? Does the x axis seem correct?

```
# for what value of lambda is dhdl large?

# 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1.0

myDeltaG.lamb[myDeltaG.dhdl > 400].unique()

array([17, 16, 15, 13, 18, 19, 14, 20, 12, 11])
```

Thus, when lambda is 0.55 to 1.0, the value of dhdl is large.

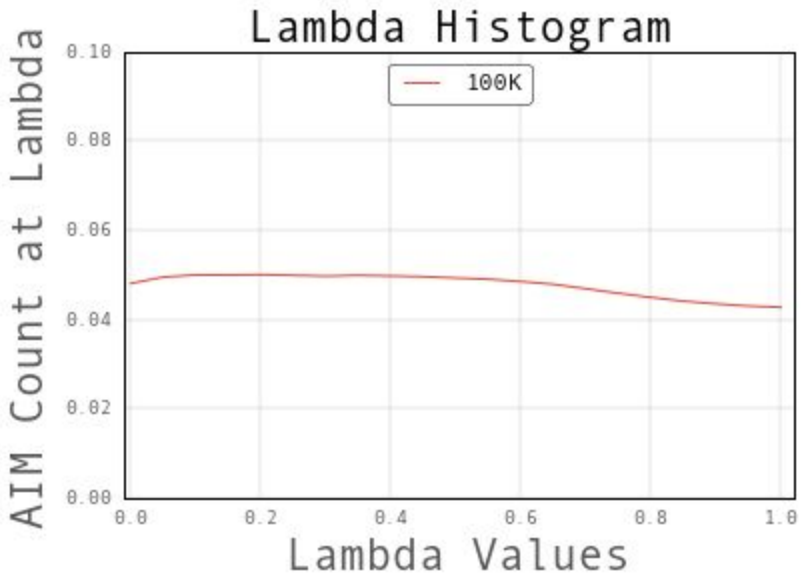
Current Problem: Where are you stuck?

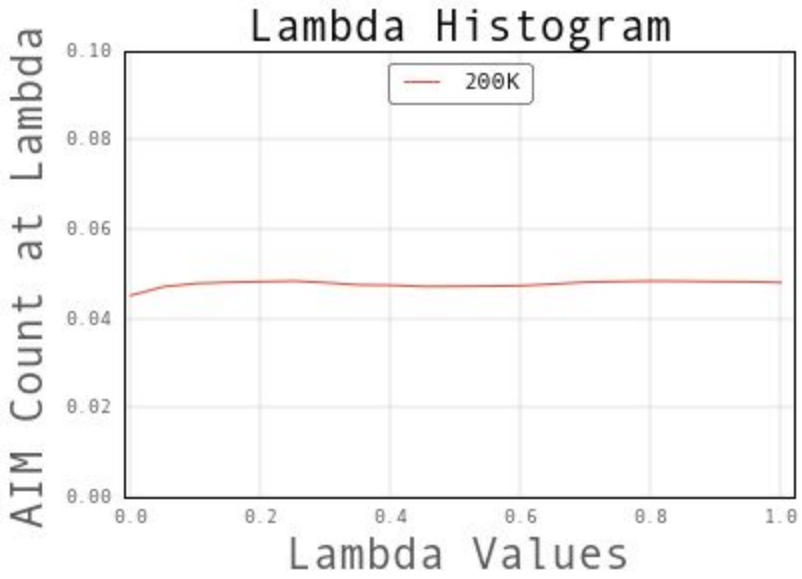
Can we figure out what's going on in the first plot? This is what we know:

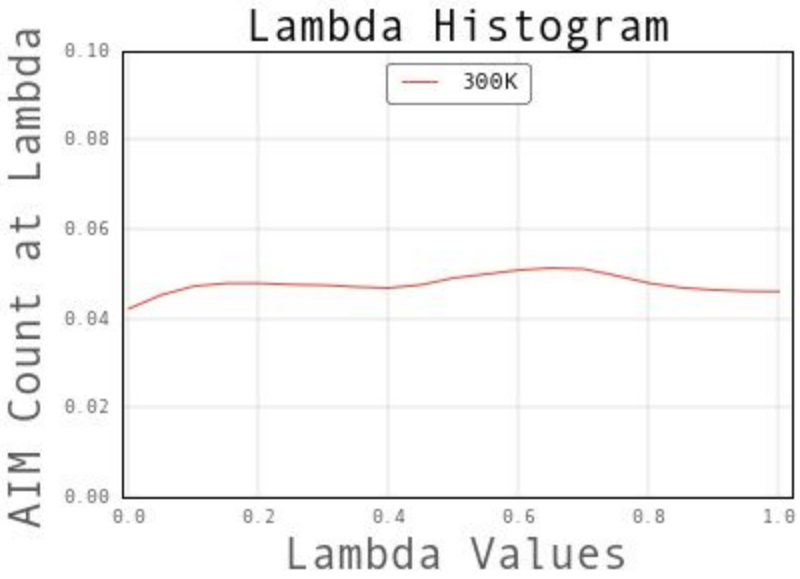
- Flatness is a function of the number of steps/iterations spent at each lambda.
- This is determined by the acceptance criteria in the AIM function.
- A decrease in flatness indicates that the algorithm spends more iterations on higher target regions.
- A change in lambda incurs a large change in energy.
- The lambda steps from 0.5 to 1.0 are van Der Waals
- The steps are smaller from 0.5 to 1.0 because we are trying to round out the curves estimate in that region.

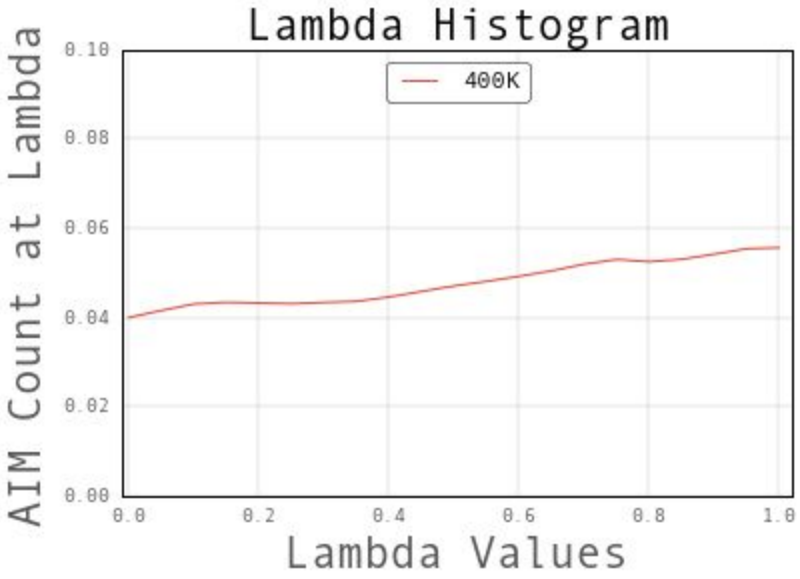
Possible resolutions: What are you planning to do? Wait for the 100 million step sim to finish.

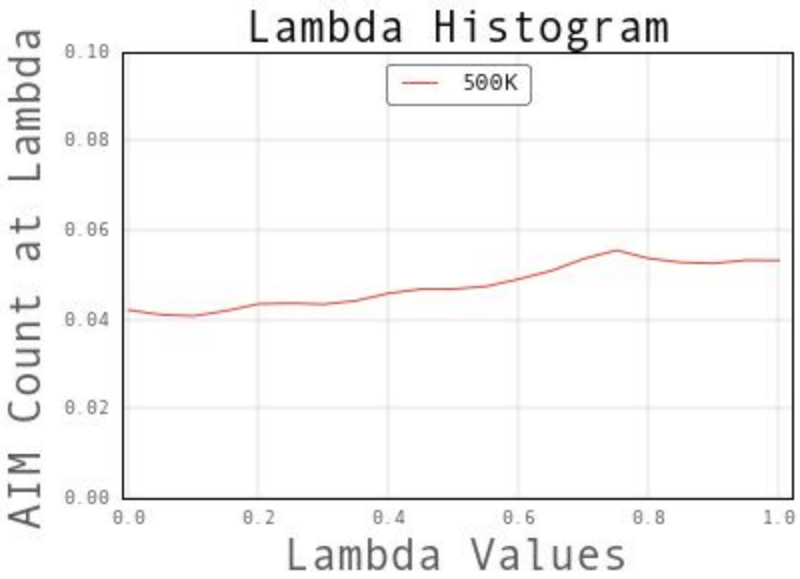
Date: 11/20/2017
Current Goal(s): What are you doing? I ran a 1 million step simulation where nstexpanded is applied at every step.
Update: What have you done? I have created histograms for every 100k steps.

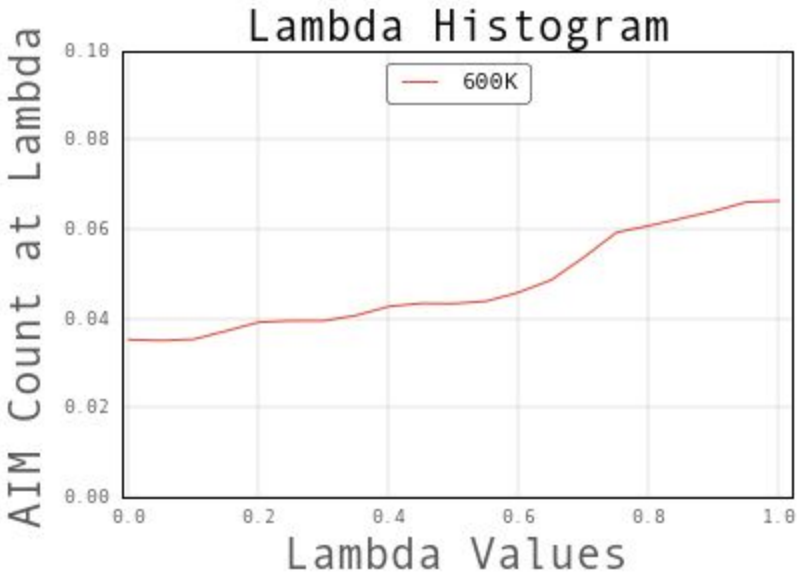


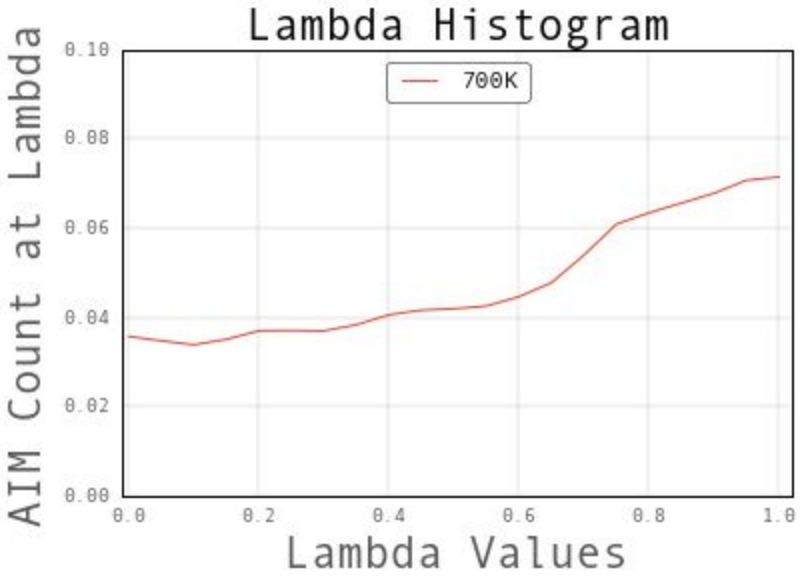


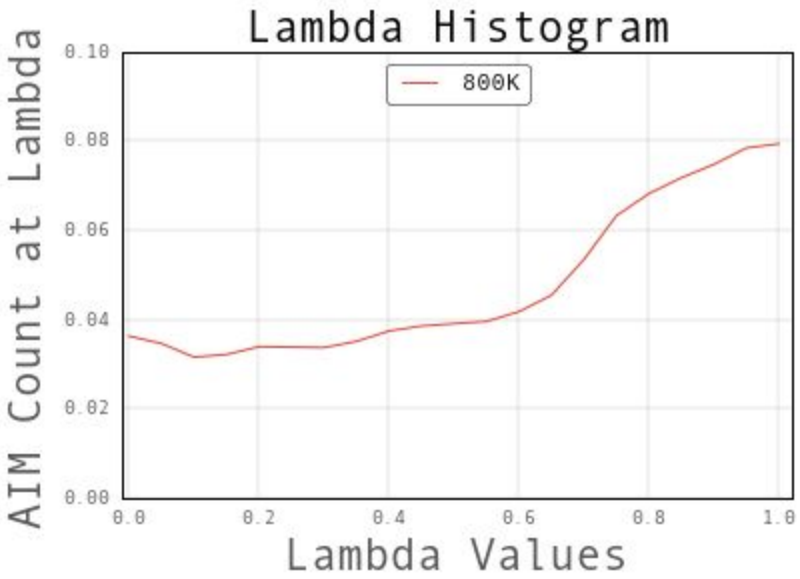

Lambda Histogram
300K
AIM Count at Lambda
Lambda Values
0.00
0.02
0.04
0.06
0.08
0.10
0.0
0.2
0.4
0.6
0.8
1.0

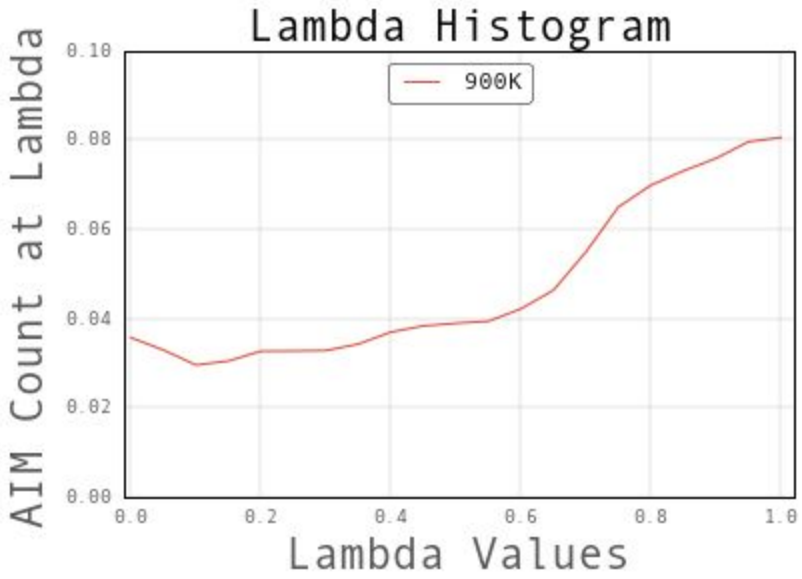

Lambda Histogram
400K
AIM Count at Lambda
Lambda Values
0.00
0.02
0.04
0.06
0.08
0.10
0.0
0.2
0.4
0.6
0.8
1.0

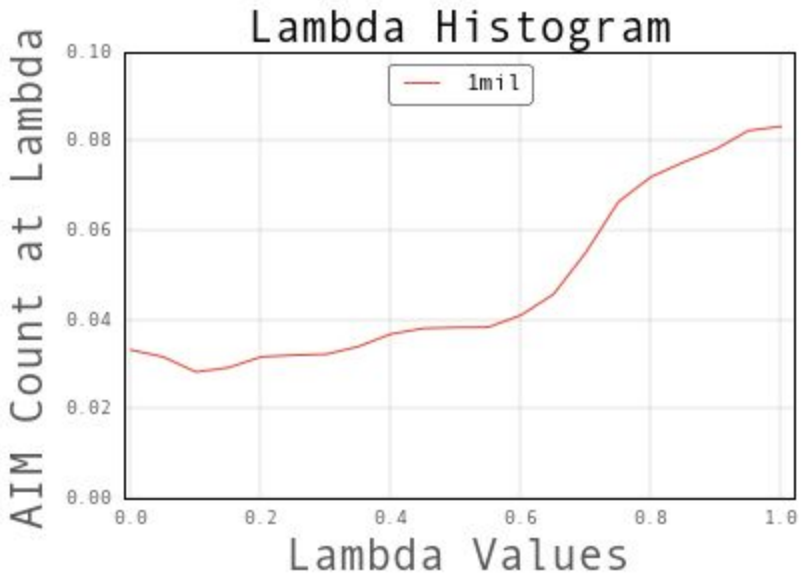

Lambda Histogram
500K
AIM Count at Lambda
Lambda Values
0.00
0.02
0.04
0.06
0.08
0.10
0.0
0.2
0.4
0.6
0.8
1.0


Lambda Histogram
600K
AIM Count at Lambda
Lambda Values
0.00
0.02
0.04
0.06
0.08
0.10
0.0
0.2
0.4
0.6
0.8
1.0


Lambda Histogram
700K
AIM Count at Lambda
Lambda Values
0.00
0.02
0.04
0.06
0.08
0.10
0.0
0.2
0.4
0.6
0.8
1.0


Lambda Histogram
800K
AIM Count at Lambda
Lambda Values
0.00
0.02
0.04
0.06
0.08
0.10
0.0
0.2
0.4
0.6
0.8
1.0



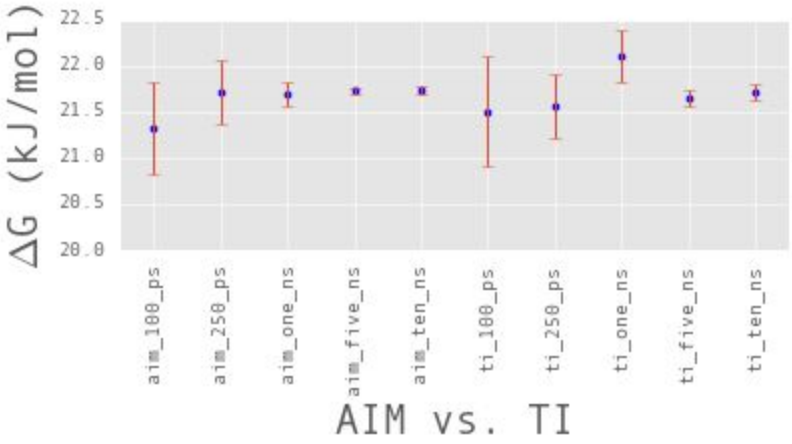


Current Problem: Where are you stuck? As you can see, the “flatness” of the lambda histogram is going away with more steps. I think this might be because AIM is running at every step and there is no equilibration time between steps.
Possible resolutions: What are you planning to do? I should probably do a 10 million step sim where AIM runs every 10 steps and see if anything changes.

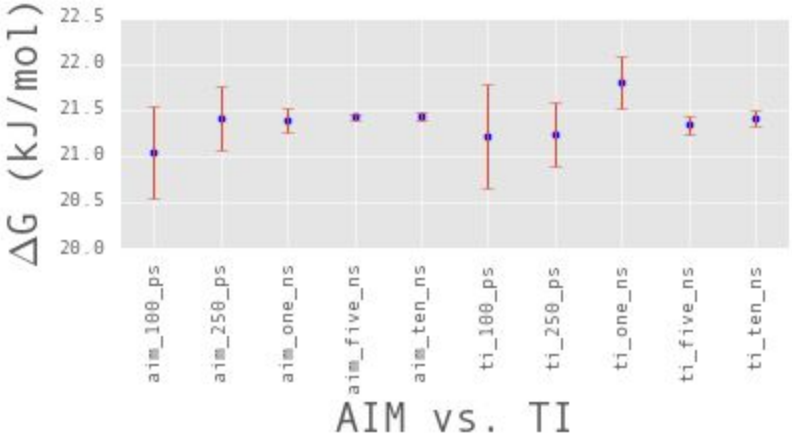
Date: 11/11/2017
Current Goal(s): What are you doing? Running simulations
Update: What have you done? I’ve run sims of 100ps, 250ps and 500ps using AIM. Now I’m running the same for TI.

Comparing AIM to TI for 100ps, 250ps, 1ns, 5ns and 10ns. TI 500 ps per lambda was started at 1:20 a.m. 11/14.

Trapezoidal Rule



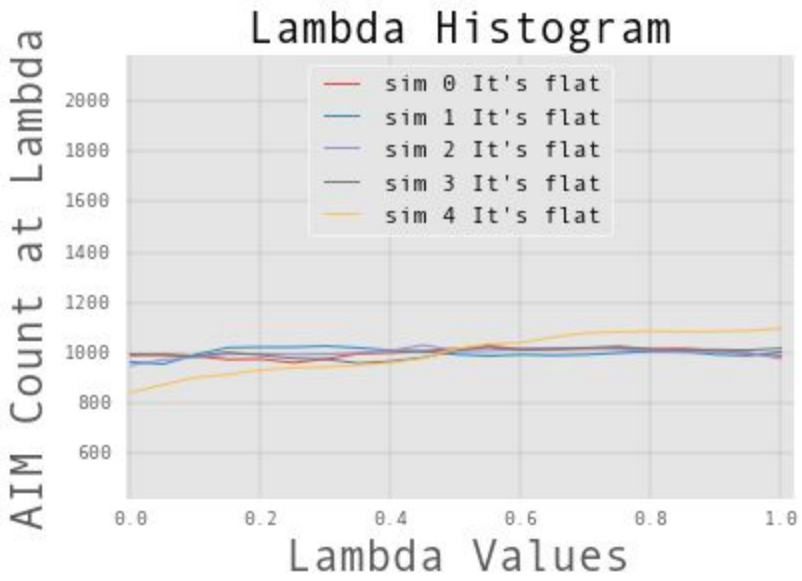
Cubic-spline



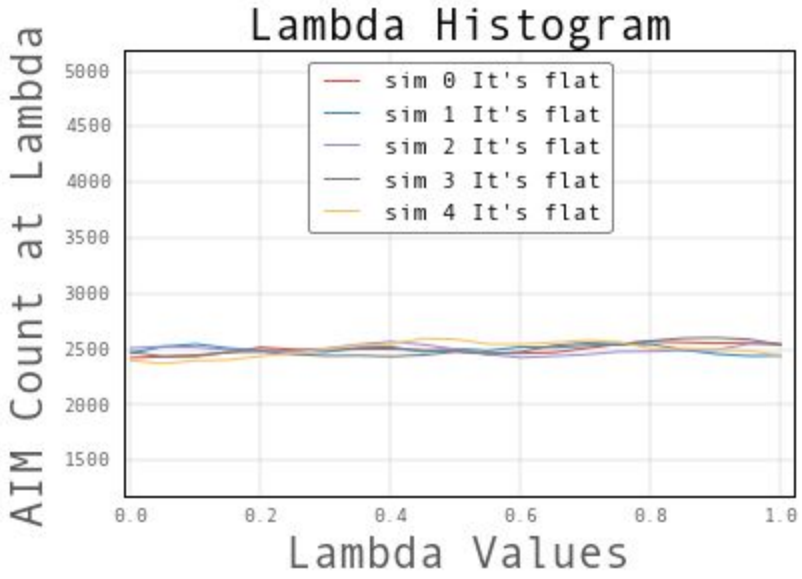
It's really cool that you can see AIM zeroing in and TI still isn't sure.

I also added the histograms here to show an interesting trend that I don't know how to explain yet. Each increase in lambda window, the histogram becomes less flat. The bin size is mentioned in the original paper so I should go back and read that. I think this is actually expected due to the use of the derivative of the free energy.

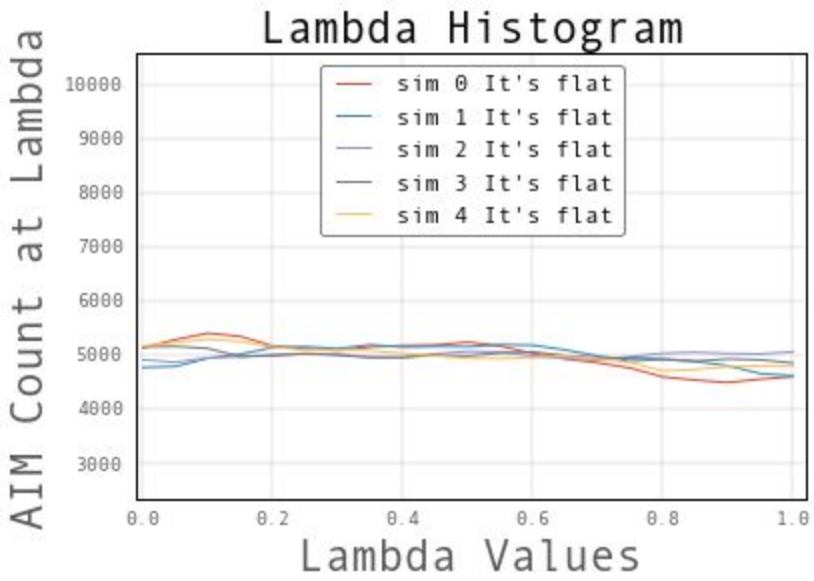
100ps


Lambda Histogram
sim 0 It's flat
sim 1 It's flat
sim 2 It's flat
sim 3 It's flat
sim 4 It's flat
AIM Count at Lambda
2000
1800
1600
1400
1200
1000
800
600
0.0
0.2
0.4
0.6
0.8
1.0
Lambda Values

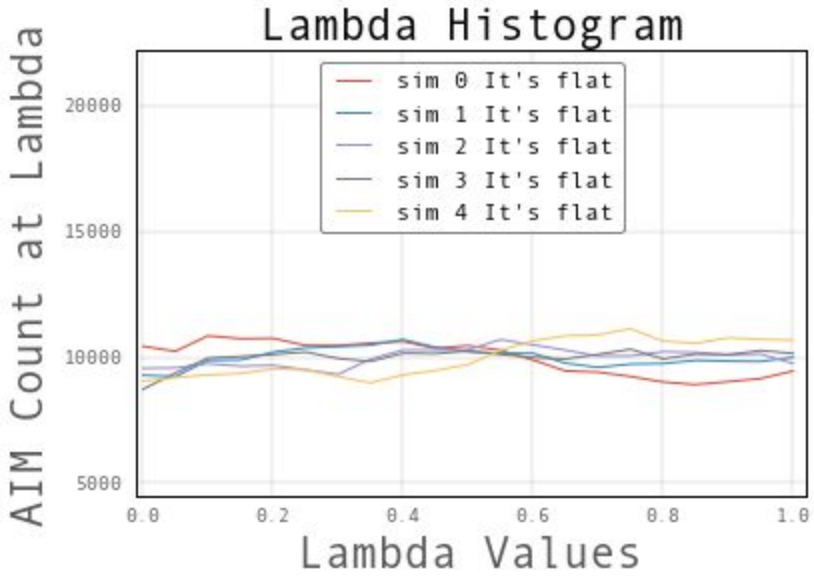
250ps


Lambda Histogram
sim 0 It's flat
sim 1 It's flat
sim 2 It's flat
sim 3 It's flat
sim 4 It's flat
AIM Count at Lambda
5000
4500
4000
3500
3000
2500
2000
1500
0.0
0.2
0.4
0.6
0.8
1.0
Lambda Values

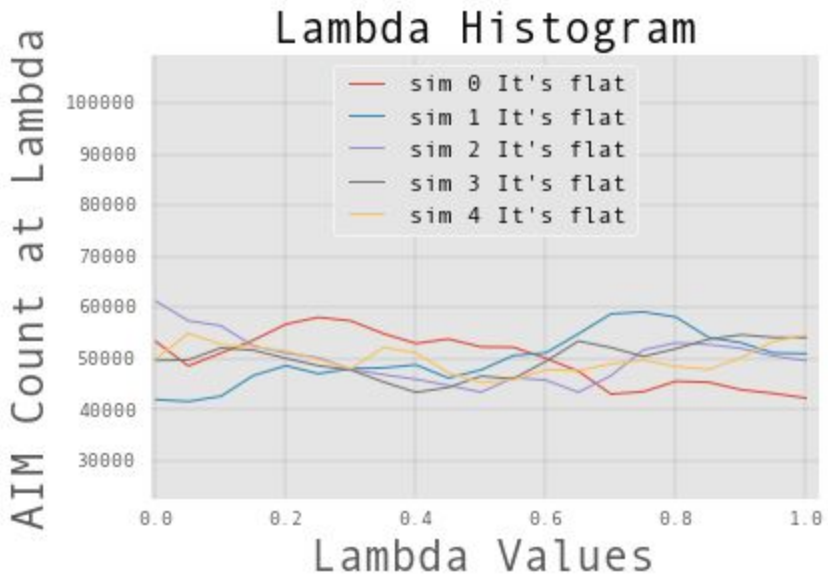
500ps


Lambda Histogram
sim 0 It's flat
sim 1 It's flat
sim 2 It's flat
sim 3 It's flat
sim 4 It's flat
AIM Count at Lambda
Lambda Values
10000
9000
8000
7000
6000
5000
4000
3000
0.0
0.2
0.4
0.6
0.8
1.0

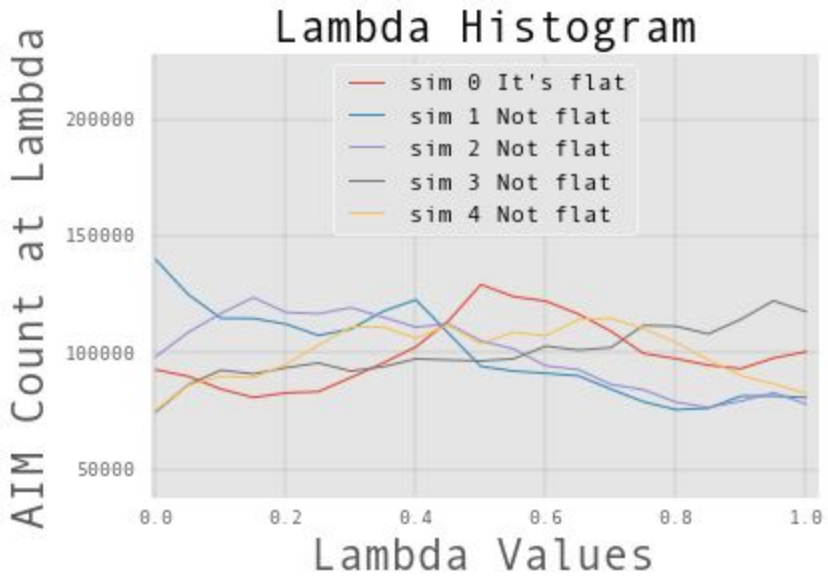
1ns


Lambda Histogram
sim 0 It's flat
sim 1 It's flat
sim 2 It's flat
sim 3 It's flat
sim 4 It's flat
AIM Count at Lambda
Lambda Values
20000
15000
10000
5000
0.0
0.2
0.4
0.6
0.8
1.0

5ns



10ns



Extra:

I keep finding myself having to figure out nsteps based on number of lambdas (expanded ensemble) and how long I want each lambda to run so I finally wrote this module (and saved it). You can run it in a Jupyter Notebook cell (Kyle) to calculate how many steps you need based on dt and how long you want to run the sim in nanoseconds. It returns number of steps. If you find an error please let me know.

```
############################################################
def calc_nsteps(dt,num_lambdas,ns_perLambda):
    # steps = ?
    # 1000 ps = 1 ns
    factor = 1000
    # want the answer in number of steps
    return num_lambdas*ns_perLambda*factor/dt
"""
```

Reference:
If dt = 0.001,
if you want 1ns per lambda
and you have 21 lambdas,
then nsteps = 21000000 ; 1 ns per lambda

If you are just running a normal sim then
num_lambdas = 1
"""
# time step CHANGE THIS BASED ON YOUR OWN SIMS
dt = 0.001
# how many lambdas?
# 1 if running normal sims
num_lambdas = 21
# how many ns per lambda? CHANGE THIS FOR YOUR OWN SIM
# 500 ps = 0.5 ns, 250 ps = 0.25 ns
# unit is ns
ns_per_lambda = 0.1   # 100ps

print(calc_nsteps(dt, num_lambdas, ns_per_lambda))
##

Current Problem: Where are you stuck? Not stuck. Running sims.
Possible resolutions: What are you planning to do? I need to runs sims on a simple mutation.

Date: 11/06/2017
Current Goal(s): What are you doing? Changing the output of the log for AIM counts and values. Testing 250ps per lambda sims and 500ps per lambda sims.
Update: What have you done? Changed the output in the log to look like this

N	CoulL	VdwL	Count	G(in kT)	dG(in kT)	AIMCount	dGd[CoulL](in kT)	dGd[VdwL](in kT)
1	0.000	0.000	22	0.00000	0.87103	301	68.98895	-38.01369
2	0.200	0.000	12	0.87103	-0.77842	294	48.83764	-13.46589
3	0.400	0.000	22	0.09261	0.29889	304	29.93038	3.14268
4	0.600	0.000	19	0.39150	0.23285	307	16.02307	11.55537
5	0.800	0.000	11	0.62435	-0.15185	303	6.98440	13.94858
6	1.000	0.000	15	0.47250	0.07024	311	1.06133	15.34400
7	1.000	0.100	13	0.54273	0.45181	312	0.35143	11.73460 <<

…..

The column header is a little long but it has both counts, AIM and standard, and it has the output values used to calculate AIM at the end.
Current Problem: Where are you stuck? Just running more sims.
Possible resolutions: What are you planning to do? Rest. I had a long weekend but didn't get anything done for research.

Date: 10/30/2017
Current Goal(s): What are you doing? Creating lambda histograms based on the log files

Update: What have you done?
I used the output from the log files to collect all of the lambda counts and sum them together based on some simple logic:

In the log file you'll see the lambda count go up to something like 25 and then the WL criteria resets the count to zero. Something like this

Step 1000
N Coul VDW Count
1 0.2 0.0 25 ...
.
.

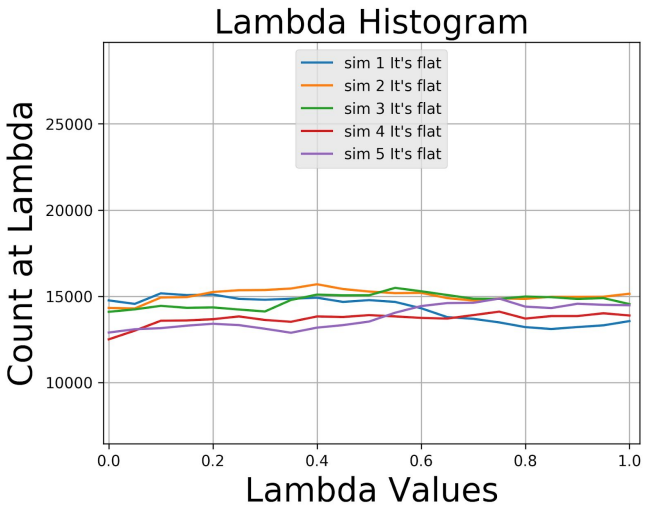
Step 1001
N Coul VDW Count
1 0.2 0.0 0 ...
.
.

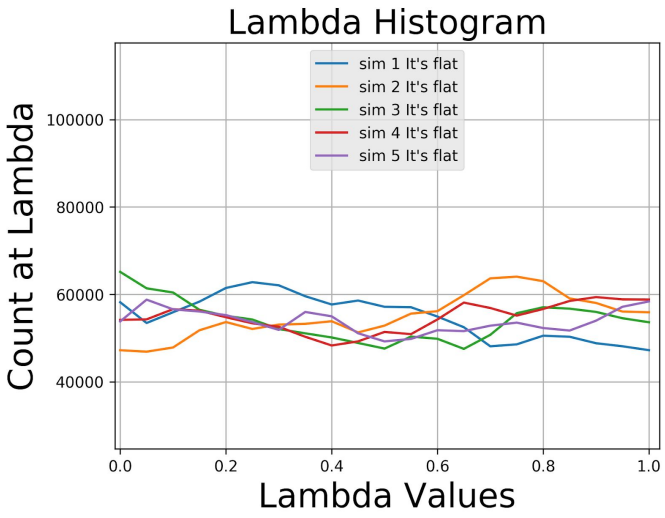
So I wrote a loop that goes through the log file and grabs all lambda equals 1 lines (separate files for each lambda). So for lambda 1 out file it looks like:
23
23
24
25
0
0
.
.
.
The code then looks for the situation when i < i+1, where i is the index. There is also the situation where you could have:
384
4900
5000

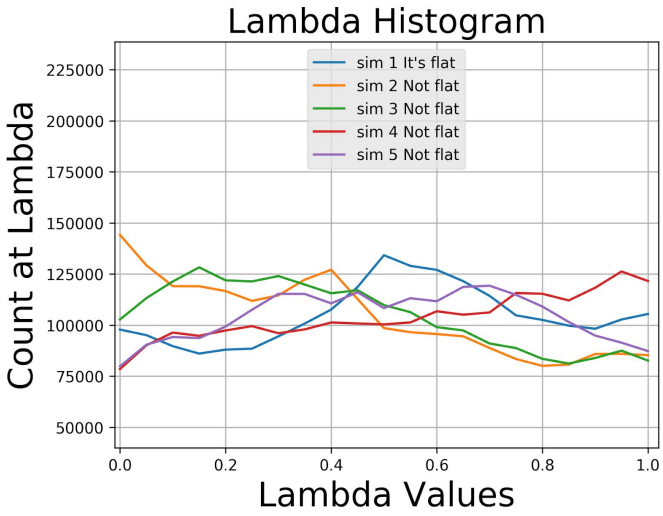
This is where WL equilibration has happened. So I added a clause
if (i > i+1) or ((i+1 - i) < 100)

Basically, I'm collecting the totals before the reset and the total at the end and summing them.

The graphs below correspond to 1ns per lambda, 5ns and 10ns.







Current Problem: Where are you stuck? Most of the 10ns per lambda are not flat enough but the results are similar to the histograms where I only used the WL count. I need to run more sims to confirm this, but I think it means we can trust the flatness due to the WL criteria without having to output the actual counts.The WL criteria resets the counts only when the histogram hits the “flat enough” criteria of 80% flatness (or whatever you set it to in your own simulations.)

Flatness = ((max - min)/max)*100

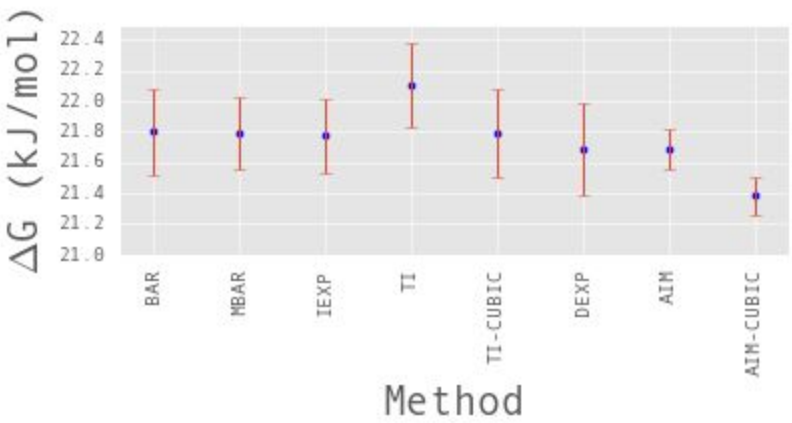
Possible resolutions: What are you planning to do? This last weekend wasn’t a good use of my time. I need to run some shorter sims to see what those look like and see if there is a way to tell when AIM starts to converge better than the other expanded ensemble movers.

Date: 10/21/2017
Current Goal(s): What are you doing? Interpolating Cubic Splines
Update: What have you done? I used Pymbar’s method for calculating Cubic Splines and this is the result:

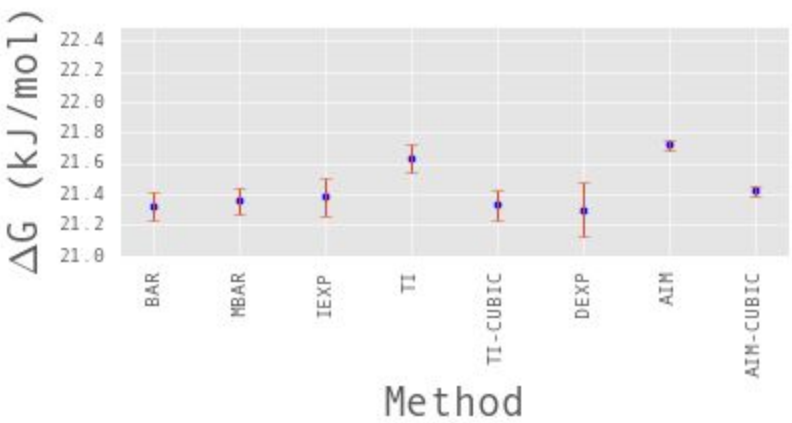
These plots are comparing averages of “direct sims” to averages of expanded ensemble with AIM.

I made sure to maintain the same scale on the y-axis for comparisons. Otherwise, it looked like more simulations didn't make a difference.
5 runs each were used to get the averages.

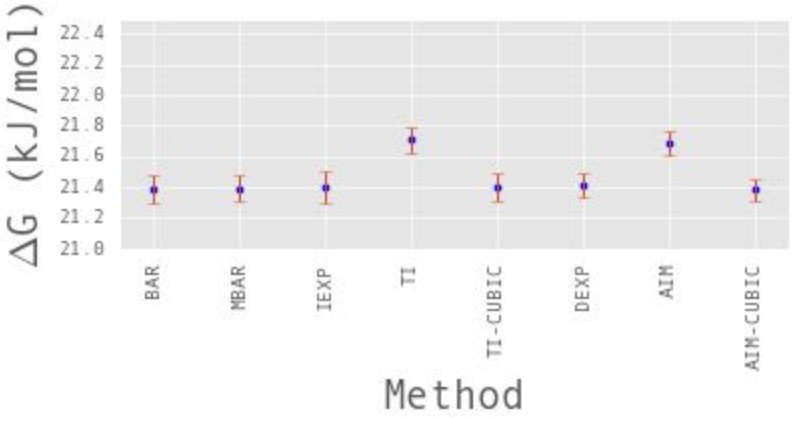
1ns per lambda



5ns per lambda



10ns per lambda



Current Problem: Where are you stuck? I don't know why the 10ns AIM sims have larger error. I double checked and all of the data is correct.
Possible resolutions: What are you planning to do? I'm rerunning the 10ns AIM sims just to double check.

Date: 10/16/2017
Current Goal(s): What are you doing? Running longer sims and attempting to calculate AIM using Cubic Spline interploation
Update: What have you done? The 10ns AIM sims are still running. They need about 5 million more steps before they are finished.

My first crack at cubic splines doesn't make much of a difference in the values. It's not easy trying to figure this out. Not as easy as I had hoped. The code I wrote was not part of Pymbar. Pymbar was hard to follow.

AIM sim 1 = 21.72948675
AIM_cubic sim 1 = 21.7015535566
AIM sim 2 = 21.71768375
AIM_cubic sim 2 = 21.6896912782
AIM sim 3 = 21.759188
AIM_cubic sim 3 = 21.7309794838
AIM sim 4 = 21.729912
AIM_cubic sim 4 = 21.7022294069
AIM sim 5 = 21.66862425
AIM_cubic sim 5 = 21.6407675951
The average is 21.72097895
The cubic average is 21.6930442641

Marty will want to see the code for the approximation so this is here mostly for him:

```
# The file_in is the copied over output from the log file
# coulLambdas and vdwLambdas are the lambda vectors
def quad_AIM(file_in, coulLambdas, vdwLambdas):
    # import the copied averages from the AIM sim
    location = file_in
    myDeltaG = pd.read_csv(
        location,
        delim_whitespace=True)

    # only need the averages from the copied info
    dGdcoul = myDeltaG.dGCoulL
    dGdvdw = myDeltaG.dGVdwL

    # we only need the delta lambdas
    dlamCoul = np.diff(coulLambdas, axis=0)
    dlamVdw = np.diff(vdwLambdas, axis=0)

    # lv are the lambda vectors
    lv = np.array([coulLambdas, vdwLambdas])
```

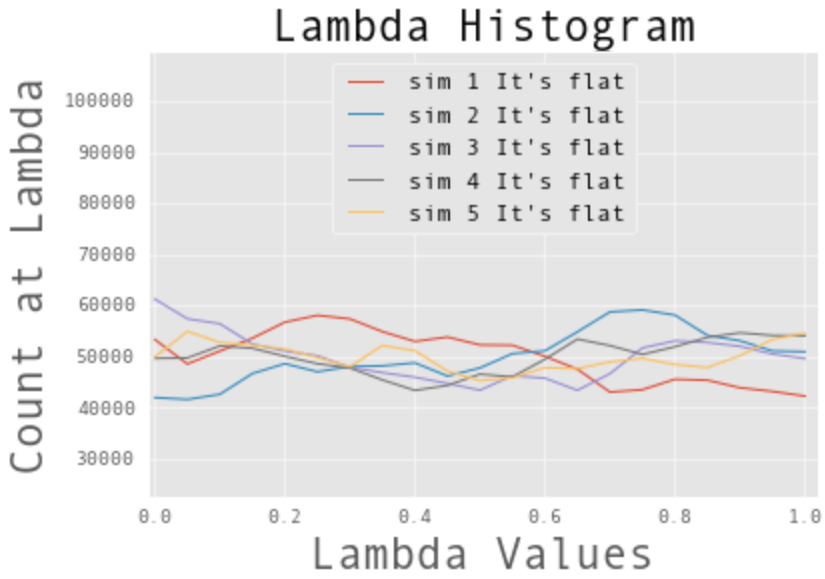
```
# K in this case would be each array of averages (coul,vdw)
# n_components is just the count
K, n_components = lv.shape

# 2d arrays to match Pymbar calculations
dlam = np.array([dlamCoul,dlamVdw])
ave_dhdl = np.array([dGdcoul,dGdvdw])

# initial values
aim = 0.0
aim_cubic = 0.0

for k in range(K):
    for j in range(n_components-1):
        if dlam[k,j] > 0.0: # saves computation time
            # regular trapezoidal rule
            aim += 0.5*np.dot(dlam[k,j],(ave_dhdl[k,j]+ave_dhdl[k,j+1]))
            # Cubic Spline coefficients
            # the curvatures function was taken from numerical methods book
            # it returns the coefficients of the cubic spline interpolation
            coefs = curvatures(np.array(range(len(ave_dhdl[k]))),ave_dhdl[k])
            # Cubic spline approximation
            # the approximation was found online
            aim_cubic += 0.5*np.dot(dlam[k,j],(ave_dhdl[k,j]+ave_dhdl[k,j+1])) \
                    - (1.0/12.0)*np.dot(dlam[k,j]**3,(coefs[j]+coefs[j+1]))
```

Current Problem: Where are you stuck? The values aren't as low as I expected.
Possible resolutions: What are you planning to do?I am going to have to go back through the Pymbar code and create print statements to see how they composed their cubic spline function.

Date: 10/01/2017
Current Goal(s): What are you doing? Checking the output of AIM
Update: What have you done? I ran 5 simulations for 105 ns each.

AIM sim 1 = 21.72948675
AIM sim 2 = 21.71768375
AIM sim 3 = 21.759188
AIM sim 4 = 21.729912
AIM sim 5 = 21.66862425

The average is 21.72097895



In case you have questions, you can see the Python code that generated the above results here:

https://github.com/bioSandMan/gmx514/blob/master/Gromacs514AIM.ipynb

Current Problem: Where are you stuck? I think I need to run more simulations now? Do I need to run 1ns,2ns,5ns, 10ns… per lambda?
Possible resolutions: What are you planning to do?

Date:09/25/2017
Current Goal(s): What are you doing? started writing code in gromacs 5.1.4 on forty four.
Update: What have you done? Finished TI and expanded ensemble sims.

TI = [21.968,

21.456,

21.819,

21.933,

21.674]

Expanded = [21.628,

21.566,

21.738,

21.736,

21.722]

Averages:

np.average(Expanded)

21.678000000000004

np.average(TI)

21.77

Standard deviation:
np.std(Expanded)

0.069201156059708085

np.std(TI)

0.18760916821946644

Current Problem: Where are you stuck? Getting a segmentation fault in the code but I think I know what it is.
Possible resolutions: What are you planning to do? Debugging, finishing the code and running AIM sims.

Date: 09/11/2017
Current Goal(s): What are you doing? Running 5ns per lambda simulations
Update: What have you done? Started the TI simulations over the weekend
Current Problem: Where are you stuck? Waiting… always waiting...
Possible resolutions: What are you planning to do? Wait until they are done and run the expanded ensemble version and wait some more.

Date: 09/04/2017
Current Goal(s): What are you doing? Testing Gromacs 5.1.4.
Update: What have you done?Same as last week but this time with multiple simulations runs.

Expanded Ensemble
alchemical_analysis -t 300 -m TI -p our01 -x -v -i 0
Run01: 21.663 +- 0.116
Run02: 21.353 +- 0.113
Run03: 21.653 +- 0.112

Run04: 21.525 +- 0.114
Run05: 21.276 +- 0.113

TI:
alchemical_analysis -t 300 -m TI -p prod01 -v -s 250 -i 0
Run01: 22.341 +- 0.159
Run02: 22.446 +- 0.160
Run03: 22.022 +- 0.159
Run04: 21.461 +- 0.159
Run05: 21.634 +- 0.161

Current Problem: Where are you stuck? Not stuck.
Possible resolutions: What are you planning to do? These simulations aren't run long enough. I need to run them longer.

Date: 08/31/2017
Current Goal(s): What are you doing?Testing direct sims (standard TI) in Gromacs 5.1.4 with the same settings as the expanded ensemble sims.

Update: What have you done? I've run the TI simulation and the results look good and are similar to the expanded ensemble results.

Standard TI:
TOTAL: 21.359 +- 0.113 21.006 +- 0.116 21.176 +- 0.297 21.057 +- 0.167 21.026 +- 0.083 21.156 +- 0.064

Expanded Ensemble:
TOTAL: 21.772 +- 0.114 21.470 +- 0.116 21.516 +- 0.180 21.604 +- 0.161 21.454 +- 0.083 21.522 +- 0.063

This is the diff between the two mdp files. The sims were run on the same day.

```
[chrism@coan 514expanded]$ diff expanded.mdp ../514/out.mdp
10c10
< nsteps               = 10500000
---
> nsteps               = 500000
95c95
< free-energy          = expanded
---
> free-energy          = yes
104c104
< init-lambda-state    = 0
---
> init-lambda-state    = ILS
129,153d128
<
```

< ; expanded ensemble Parameters.
< ; every 100 steps, we try switch between the intermediate states.
< nstexpanded = 100
< ; Wang-Landau algorithm to determine the free energies 'weights' of the states
< lmc-stats = wang-landau
< ; Metropolized gibbs algorithm to move between states
< lmc-move = metropolized-gibbs
< ; we stop equilibrating when the wang-landau scaling term gets as low as 0.001
< lmc-weights-equil = wl-delta
< weight-equil-wl-delta = 0.001
<
< ; Seed for Monte Carlo in lambda space
< ; We scale our wang landau weight by 0.7, whenever the smallest state
< ; and largest state have ratio of 0.8. The initial wang-landau weight
< ; increment delta is 1 kbT, and when this delta<1/N, where N is the
< ; number of attempted switches in state space, we use 1/N as the delta,
< ; which is less prone to saturation (stopping at the wrong value because
< ; the weight schedule lowered too quickly).
< ;
< ;
< wl-scale = 0.7
< wl-ratio = 0.8
< init-wl-delta = 1.0
< wl-oneovert = yes

Current Problem: Where are you stuck? At work mostly…. But I am having fun and learning a lot about Machine Learning and AI.

Possible resolutions: What are you planning to do? This means that I'm ready to run AIM. I've already started writing the code into 5.1.4 and hope to run a simulation this weekend. Or 5? maybe ?

Date: 08/20/2017
Current Goal(s): What are you doing?Testing TI and expanded ensemble with new settings from Gromacs 2016.
Update: What have you done? I have finally figured out how to run the simulations and for TI (direct sims) I get:

TI
TOTAL: 21.548 +- 0.331 21.119 +- 0.340 21.507 +- 0.705 21.788 +- 0.473 21.060 +- 0.259 21.238 +- 0.205

And for expanded ensemble I get nearly the same. So everything looks good again.

Current Problem: Where are you stuck? I'm not sure which setting did the trick so I am testing soft core settings and Electrostatics/VDW settings.

Update: Using the previous sc settings the simulation ran and the results are good:

TOTAL: 21.280 +- 0.410 21.014 +- 0.418 20.721 +- 0.627 21.439 +- 0.511 21.000 +- 0.303 21.356 +- 0.235

This makes me think it was the electrostatics/vdw settings.

Possible resolutions: What are you planning to do? Once I understand the settings a little better, or even before that, I am ready to import AIM code into Gromacs 5.1.4.

Date: 08/12/2017
Current Goal(s): What are you doing? Trying to figure out the correct way to run expanded ensemble simulations in Gromacs 5.1.4.

Update: What have you done? The simulations that I've run have a very large van der waals value for TI. However, Bar, IEXP and Mbar methods all have very good results.

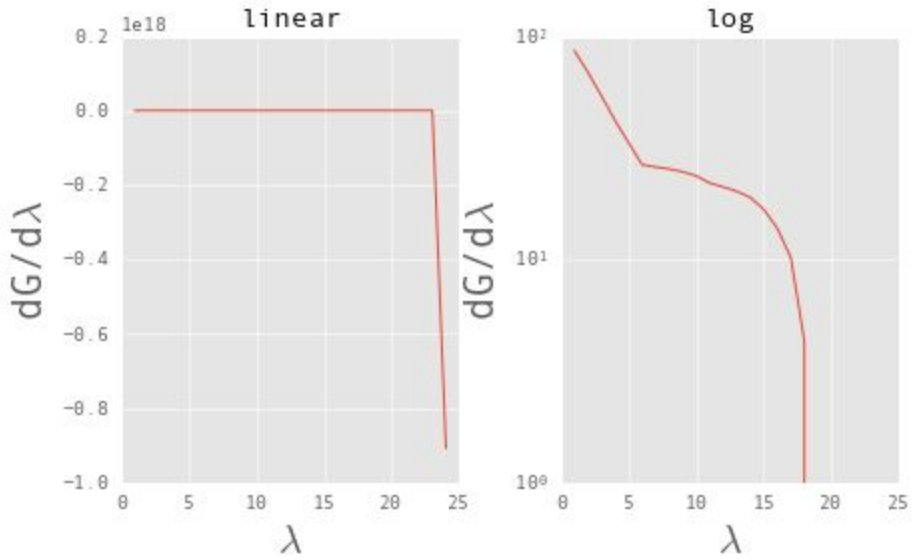
If I look at the individual sums of each lambda VDW we see that lambda = 23 is 0, lambda = 24 is very large

(0, 59141.988895417999)
(1, 87427.156876926005)
(2, 103225.75215248999)
(3, 110079.53639719999)
(4, 112752.9205597)
(5, 113880.28403350001)
(6, 111274.59909999999)
(7, 108429.17927851)
(8, 104352.32568139999)
(9, 100467.85776785)
(10, 93449.416253410003)
(11, 88828.469827403998)
(12, 84487.320095989999)
(13, 78673.295528292001)
(14, 70880.334065444011)
(15, 59097.530547081304)
(16, 43893.436408171401)
(17, 17981.825856822801)
(18, -35954.431055093301)
(19, -87470.2613370209)
(20, -189662.61365320501)
(21, -376724.55367421603)
(22, -593977.61834872025)
(23, 0)
(24, -2.7903401438685902e+21)

I thought if I removed the last two lambdas from the equation that it might help but only partially.

Total = 44.016814

I plotted dG/dlambda as a function of lambda but I'm not sure how it helps:



In [29]:

Current Problem: Where are you stuck? I feel that I need to understand how these methods calculate the free energy to understand what I'm doing incorrectly however it could be the way pymbar is calculating TI.

Possible resolutions: What are you planning to do? I'm also considering just coding AIM into 5.1.4 and running it to see what happens.

Date: 08/08/2017
Current Goal(s): What are you doing? I've managed to get simulations to run but the value for TI explodes in the van der waals term.
Update: What have you done? Tried different settings for the mdp file
Current Problem: Where are you stuck? I don't know why the VDW term is getting large.
Possible resolutions: What are you planning to do? Porting AIM code to 5.1.4 and continue testing mdp options to see if I can make it work.

Date: 08/01/2017
Current Goal(s): What are you doing? Running expanded ensemble simulations using Gromacs 5.1.4.
Update: What have you done? I tried a metropolized-gibbs simulation and got an error:

"Fatal error:
Something wrong in choosing new lambda state with a Gibbs move probably underflow in weight determination"

In trying to find out more about this error I found this thread:
http://thread.gmane.org/gmane.science.biology.gromacs.user/63021

If you read down you'll find this explanation:

> The basic problem is that for this particular configuration, the
> current state is the only state with nonzero weight. Note that the
> state with the second highest weight has weight 10^-7. When it tries
> to compare weights in single precision, it has a numerical overflow
> and fails.
>
> A few things:
>
> 1. This really should be more robust, so that it will realize it's
> supposed to stay in the most likely state, since that's the only state
> with nonzero weight. I have a fix that I've been working on for
> exactly this problem, but it's not quite ready yet. Hopefully in the
> next couple of days.
>
> 2. This problem is very unlikely to occur in double precision, if you
> can afford the performance hit in the meantime.
>
> 3. If this is a typical average difference, exchanges will be very
> unlikely. You should probably choose your lambda intervals to be a
> bit closer together at the end range.
>
> Hopefully this will give you enough information to move forward for
> the time being until a better fix is implemented.

The only thing I can try is the 3rd option but before doing that I decided to try a metropolis run. I didn't get the error with the metropolis mover so I ran the results into pymbar:

States	TI (kJ/mol)	TI-CUBIC (kJ/mol)	DEXP (kJ/mol)	IEXP (kJ/mol)	BAR (kJ/mol)	MBAR (kJ/mol)
TOTAL:	-83555626246573.438 +- 50270158147937.641		-65898429518720.594 +- 39646934891390.086			
	35.361 +- 0.530	25.025 +- 1.791	21.897 +- 0.815	21.836 +- 0.815		

So, BAR and MBAR look really good but TI is huge. In looking further into the log I found that the last state had not been visited very much so it seems that it's the same problem as with metropolized-gibbs.

Current Problem: Where are you stuck?
Possible resolutions: What are you planning to do? I'm going to run simulations with closer lambdas and further lambdas to see what makes a difference.

Date: 07/24/2017
Current Goal(s): What are you doing? Trying to run the tutorials from alchemistry.org in gromacs 5.1.4
Update: What have you done? Working through all of the (insert expletive) errors to turn a fixed lambda sim into an expanded ensemble simulations in 5.1.4

error 1 [file ligand.mdp]:
Can only use expanded ensemble with md-vv (for now)

error 2 [file ligand.mdp]:
  for md-vv and md-vv-avek, can only use Berendsen and
  Martyna-Tuckerman-Tobias-Klein (MTTK) equations for pressure control;
  MTTK is equivalent to Parrinello-Rahman.

error 3 [file ligand.top, line 205]:
  MTTK not compatible with lincs -- use shake instead.

error 4 [file ligand.top, line 205]:
  Constraints are not implemented with MTTK pressure control.

Current Problem: Where are you stuck? I worked through the errors except for the last one.
Possible resolutions: What are you planning to do? I emailed the guy that made the tutorials and asked if he knew what was going on.

My email to him:

Hello,

I've learned a lot and am very grateful for the alchemistry.org website. I have a question. I noticed that I'm getting different results than you (lower results) using the supplied files for the expanded ensemble tutorial in GROMACS 5.0. However, in GROMACS 5.0, the ethanol salvation tutorial, I get nearly the same results. I thought there might be a bug in 5.0 expanded ensemble so I moved to 5.1. In 5.1 I am getting an error that constraints are not supported with MTTK pressure settings. Could you tell me what needs to be done to upgrade the tutorial to 5.1?

Thanks,

-ChrisM

His response:

Hi Chris,

Mmm that's interesting - firstly however I would try running the expanded ensemble approach using the same mdp options used in the absolute binding free energy tutorial. This because those might be the cause of the systematic difference you see. E.g. I used LJ-PME, which is available only in the more recent GMX versions, and this would always return larger binding free energies than having a cut-off, since the long-range part of the LJ interactions is always attractive.

I must also say that I did not write the expanded ensemble tutorial (I believe Michael Shirts did) as I am not very familiar with it.

Regarding more specifically the error you get with Gromacs 5.1, it is probably just because that combination of inputs is not supported. So the only way around it with that version of Gromacs would be to switch to Parrinello-Rahman for pressure coupling, and possibly also to the md integrator if you're using the md-vv one. The other option is not to use constraints (or maybe constraints other than LINCS, like the SHAKE ones), but then you'd end up having to run double the amount of integration steps. It is strange you didn't get this error with Gromacs 5.0 though, so imagine in that case you didn't use MTTK? In fact it seems that in the ethanol solvation tutorial the sd integrator is used with Parrinello-Rahman for pressure coupling. This is actually probably the safer combination at the moment for free energy calculations. I believe Michal Shirts have been experimenting/implementing other integrators and coupling schemes, and that's probably why his tutorial uses those, but I think they are more recent additions to Gromacs so they are more likely not to work with certain other options...

Hope this helps somewhat.

Best,
Matteo

I've tried what he recommended and I get the same errors above. I think I have to start over from scratch and remake the topology and gro files.

Date: 07/14/2017
Current Goal(s): What are you doing? Reorienting to research
Update: What have you done? Ran some simulations
Current Problem: Where are you stuck?
Possible resolutions: What are you planning to do?

Date: 05/30/2017
Update:
We are in the middle of moving to Spokane. We finally found a place to rent and we have until mid June to completely move. I don't typically share personal things with the group but the move, the new job plus the stress of my daughter's illness are all wearing me out. I haven't looked at AIM for weeks and I don't think I will be able to until we have fully moved to Spokane.

Date: XX/YY/ZZZZ
Current Goal(s): What are you doing? Debugging AIM
Update: What have you done? I emailed the GROMACS dev group about the correct de term and got this response:

"So, the updated data can be seen most clearly in mdebin.c, in the function upd_mdebin, right after:

```
if ((md->fp_dhdl || md->dhc) && bDoDHDL)
```

The components for dH/dL are in enerd->term[F_DVDL+i]), where i=0,1,2,3,4, and you can see by reading the dhdl output what the order is.

The potential energy differences are calculated from: from enerd->enerpart_lambda[i+1]-enerd->enerpart_lambda[0]

If you are only interested in the neighboring differences, the can be calculated from enerd->enerpart_lambda[i+1]-enerd->enerpart_lambda[i] (as the [0] component cancels out).

So the question is where in the code are they "in step" with the timestep?

When one gets to ExpandedEnsembleDynamics in md.c, then they are all updated. However, if you are using velocity verlet, then they are not finished updating until after sum_dhdl(enerd, state->lambda, ir->fepvals) is called. If you only acess these arrays within the ExpandedEnsembleDynamics (which it sounds like you would be doing, since AIM deals with transitions between states), then you should be fine."

Current Problem: Where are you stuck? Every de that I try, the result is the same, the answer is around 15.x. I've tried this in both 5.0.4 and 5.0.7. Using this value of de is exact and it's what I was originally using. The way I am calculating df matches the solution of Pymbar's TI which also means that I'm storing the same value for the average that Pymbar uses to calculate TI. I've used the same system throughout all of my simulations. The "direct" simulations have the result much higher, so does the literature.

Possible resolutions: What are you planning to do?

Date: 04/18/2017
Current Goal(s): What are you doing? Debugging AIM
Update: What have you done?

I found this is the replica exchange code

```
  if (re->type == ereLAMBDA || re->type == ereTL)
  {
      bDLambda = TRUE;
      /* lambda differences. */
      /* de[i][j] is the energy of the jth simulation in the ith Hamiltonian
         minus the energy of the jth simulation in the jth Hamiltonian */
      for (i = 0; i < re->nrepl; i++)
      {
          for (j = 0; j < re->nrepl; j++)
          {
              re->de[i][j] = 0;
          }
      }
      for (i = 0; i < re->nrepl; i++)
      {
          re->de[i][re->repl] = (enerd->enerpart_lambda[(int)re->q[ereLAMBDA][i]+1]-enerd->enerpart_lambda[0]);
      }
  }
```

[5:30]
read the comments about de[i][j]

Current Problem: Where are you stuck? How do I use this? The value for de is what I've been using. It gets updated every step. What I don't understand is whether or not this value is the difference between previous steps or current steps. I think it's current steps.
Possible resolutions: What are you planning to do?

Date: 04/11/2017
Current Goal(s): What are you doing? Debugging AIM
Update: What have you done?

I've tried different versions of the energy difference (de) and different versions of the acceptance criteria. Assuming my value for df is correct, based on Pymbar, the problem must lie with de and/or the acceptance criteria of exp(de+df).

I found the energy differences are printed to the xvg file. The xvg file is written by methods inside of mdebin.c. I found the energy differences being calculated by:

md->dE[i] = enerd->enerpart_lambda[i+1]-enerd->enerpart_lambda[0];

I confirmed that this is the value being printed to the xvg file.

Xvg output:

0.0000 0 73.743774 -11.946036 1.5258789e-05 14.748753 29.497392 44.246176 58.994952 73.743711 73.215749 73.736161 74.990772 76.768417 78.923009 80.108059 81.351008 82.643290 83.977579 85.347550 86.747758 88.173467 89.620583 91.085537 92.565217 0.56809938

Only the values in red are energy differences. The first term is the time step, second and third terms are dvdl terms. The very last term is the pV term.

Output from md->dE
md->dE[0] 0.00002
md->dE[1] 14.74875
md->dE[2] 29.49739
md->dE[3] 44.24618
md->dE[4] 58.99495
md->dE[5] 73.74371
md->dE[6] 73.21575
md->dE[7] 73.73616
md->dE[8] 74.99077
md->dE[9] 76.76842
md->dE[10] 78.92301
md->dE[11] 80.10806
md->dE[12] 81.35101
md->dE[13] 82.64329
md->dE[14] 83.97758
md->dE[15] 85.34755
md->dE[16] 86.74776
md->dE[17] 88.17347
md->dE[18] 89.62058
md->dE[19] 91.08554
md->dE[20] 92.56522

After a few more tests I found that the output of md->dE wasn't always what I expected.

	Lambda	**de_00**	**de_01**	**de_02**	**de_03**	**de_04**	**de_05**	**de_06**	**de_07**
0	0.0	-0.000015	14.748746	29.497392	44.246176	58.994939	73.743694	73.215732	73.736149

5	1.0	-14.077061	0.000008	14.077092	28.154237	42.231387	56.308531	55.301698	55.508467

9	2.0	-27.521731	-13.760879	0.000008	13.760913	27.521798	41.282693	40.339056	40.560564

The values contain the energy difference between the “native” lambda and the “foreign” lambdas. Notice that when Lambda = 0.0, de_00 is essentially zero. The same with Lambda = 1.0 and de_01. It looks like these are the energy differences between the “current” lambda and all other lambdas. This also implies that enerd->enerpart_lambda[0] is the energy of the current or “native” lambda. So what is the origin of the value? I actually can’t figure that out. When I print out the 0th term, it doesn’t match any of the energy terms printed out in the log file.

Armed with this “new” value, I performed some experiments with different values of de and the acceptance criteria.

Relevant mdp settings:

```
nsteps               = 10500000    ; 1 ns per lambda
nstcalcenergy            = 100
nstdhdl                = 100
nstexpanded              = 100
separate-dhdl-file       = no
dhdl-derivatives         = yes
cutoff-scheme            = Verlet
```

Lmc-stats (wang-landau equilibration) was disabled unless otherwise noted.

In all experiments, df is calculated as;

```
for (j=0; j < efptNR; j++)
{
    if (fepvals->separate_dvdl[j])
    {
        df += 0.5
          *(fepvals->all_lambda[j][lamtrial] - fepvals->all_lambda[j][fep_state])
          *(dfhist->sum_dvdl[j][lamtrial] + dfhist->sum_dvdl[j][fep_state])
          *(1.0/(expand->mc_temp*BOLTZ));
    }
}
```

The calculation for df matches that of Pymbar/TI.

Experiment 00: Sanity Check. Am I going insane(Maybe)? Was the solution positive or negative? What were the units again?

- Look at one of the old “direct” sims with 1ns per lambda and using the “new” lambda schedule.
    - Check the input files
        - The gro file and top file match what I’ve been using for AIM. The major settings in the mdp file are the same. Nstdhdl was 100.
    - Estimate the free energy change with Pymbar/TI

$alchemical_analysis -t 300 -x -v -p prod -m TI -a Gromacs -i 0

```
Coulomb:     27.205  +-  0.080
vdWaals:     -8.532  +-  0.085
  TOTAL:     18.672  +-  0.116
```

The units were (kJ/mol). The units for my result are kJ/mol.

The expected value from literature is 5 kcal/mol which is 20.92 kJ/mol.

Experiment 01: Gromacs 5.0.7. How does the "new" dE change the results?

- Result: 16.86 This result is closer than what I've been getting, but still not right.
- dy = (enerd->enerpart_lambda[lamtrial+1] - enerd->enerpart_lambda[0])*(1.0/(expand->mc_temp*BOLTZ));
- trialprob = exp(-dy+df);

N	CoulL	VdwL	Count	G(in kT)	dG(in kT)	dGd[CoulL]	dGd[VdwL]
1	0.000	0.000	3443	0.00000	0.00000	49.19154	-11.50584
2	0.200	0.000	4918	0.00000	0.00000	47.30220	-10.44317
3	0.400	0.000	5249	0.00000	0.00000	29.94618	3.23328
4	0.600	0.000	5541	0.00000	0.00000	16.18690	10.28211
5	0.800	0.000	5651	0.00000	0.00000	7.23301	12.49816
6	1.000	0.000	5702	0.00000	0.00000	0.01656	13.64898
7	1.000	0.100	5753	0.00000	0.00000	0.15832	11.62215
8	1.000	0.200	5670	0.00000	0.00000	-0.15711	9.97982
9	1.000	0.300	5491	0.00000	0.00000	-0.16025	7.56658
10	1.000	0.400	5349	0.00000	0.00000	-0.02990	3.76502
11	1.000	0.500	5204	0.00000	0.00000	0.23218	-3.55155
12	1.000	0.550	5041	0.00000	0.00000	0.03186	-8.66030
13	1.000	0.600	4975	0.00000	0.00000	0.22889	-15.13137
14	1.000	0.650	5055	0.00000	0.00000	2.53643	-24.01683
15	1.000	0.700	4841	0.00000	0.00000	0.19183	-37.56108 <<
16	1.000	0.750	4632	0.00000	0.00000	1.31382	-45.99666
17	1.000	0.800	4534	0.00000	0.00000	-0.82487	-43.24601
18	1.000	0.850	4358	0.00000	0.00000	-10.38982	-32.90984
19	1.000	0.900	4453	0.00000	0.00000	-6.84242	-20.84859
20	1.000	0.950	4574	0.00000	0.00000	8.36856	-9.62420
21	1.000	1.000	4566	0.00000	0.00000	0.63592	-0.00852

Experiment 02: Gromacs 5.0.7. What happens if I change sign of dy?

- Result: 26.44. This result is too high and the histogram isn't flat.
- dy = (enerd->enerpart_lambda[lamtrial+1] - enerd->enerpart_lambda[0])*(1.0/(expand->mc_temp*BOLTZ));
- trialprob = exp(dy+df);

N	CoulL	VdwL	Count	G(in kT)	dG(in kT)	dGd[CoulL]	dGd[VdwL]
1	0.000	0.000	1	0.00000	0.00000	72.61532	-33.47468

2	0.200	0.000	2	0.00000	0.00000	69.33799	-9.89875
3	0.400	0.000	44	0.00000	0.00000	28.98283	3.08159
4	0.600	0.000	560	0.00000	0.00000	14.78297	10.91952
5	0.800	0.000	1804	0.00000	0.00000	6.17231	12.41496
6	1.000	0.000	2358	0.00000	0.00000	0.16430	12.69547
7	1.000	0.100	4633	0.00000	0.00000	-0.07037	11.17811
8	1.000	0.200	8168	0.00000	0.00000	0.03455	9.06883
9	1.000	0.300	12418	0.00000	0.00000	0.11598	6.00781
10	1.000	0.400	15895	0.00000	0.00000	-0.10007	1.84585
11	1.000	0.500	16587	0.00000	0.00000	0.18892	-4.04453
12	1.000	0.550	14727	0.00000	0.00000	0.45545	-9.00623
13	1.000	0.600	11382	0.00000	0.00000	-0.36320	-15.00583
14	1.000	0.650	7398	0.00000	0.00000	-1.23796	-20.54082
15	1.000	0.700	4134	0.00000	0.00000	-3.32914	-24.06239 <<
16	1.000	0.750	2125	0.00000	0.00000	-2.66366	-25.37313
17	1.000	0.800	1106	0.00000	0.00000	4.01280	-24.33015
18	1.000	0.850	615	0.00000	0.00000	-11.12083	-19.41537
19	1.000	0.900	398	0.00000	0.00000	0.04600	-13.62844
20	1.000	0.950	320	0.00000	0.00000	-10.73088	-6.27050
21	1.000	1.000	325	0.00000	0.00000	9.97968	1.80753

Experiment 03: Gromacs 5.0.7. What happens if I change the sign using original value for de?

- Result: 23.00781. This is closer but the histogram is not flat. Changing sign was wrong.
- de = weighted_lamee[lamtrial] - weighted_lamee[fep_state];
- trialprob = exp(-de+df);

N	CoulL	VdwL	Count	G(in kT)	dG(in kT)	dGd[CoulL]	dGd[VdwL]
1	0.000	0.000	2	0.00000	0.00000	74.34986	-29.41245
2	0.200	0.000	1	0.00000	0.00000	53.84623	21.89536
3	0.400	0.000	32	0.00000	0.00000	27.31845	4.92563
4	0.600	0.000	589	0.00000	0.00000	14.70018	10.70391
5	0.800	0.000	1841	0.00000	0.00000	6.57370	12.11096
6	1.000	0.000	2420	0.00000	0.00000	0.30822	12.06662
7	1.000	0.100	4652	0.00000	0.00000	0.02763	10.58458
8	1.000	0.200	8287	0.00000	0.00000	-0.01343	8.94490
9	1.000	0.300	12514	0.00000	0.00000	-0.13300	6.14538
10	1.000	0.400	15943	0.00000	0.00000	-0.00005	1.54704
11	1.000	0.500	16707	0.00000	0.00000	-0.03653	-3.86623
12	1.000	0.550	14726	0.00000	0.00000	0.08443	-9.20605 <<
13	1.000	0.600	11216	0.00000	0.00000	0.38475	-15.25383
14	1.000	0.650	7548	0.00000	0.00000	-0.16631	-20.99732
15	1.000	0.700	4299	0.00000	0.00000	0.19830	-23.68395
16	1.000	0.750	2098	0.00000	0.00000	-3.06439	-23.65141
17	1.000	0.800	1033	0.00000	0.00000	1.94436	-24.54188
18	1.000	0.850	519	0.00000	0.00000	-8.94669	-20.89507
19	1.000	0.900	268	0.00000	0.00000	-0.88440	-13.90878
20	1.000	0.950	163	0.00000	0.00000	17.50687	-7.53534
21	1.000	1.000	142	0.00000	0.00000	-6.63385	0.09665

Experiment 04: Gromacs 5.0.4. “Original”. What was the problem I was trying to solve???

- Result: 14.994982 Oh, yeah, now I remember.
- de = weighted_lamee[lamtrial] - weighted_lamee[fep_state];
- trialprob = exp(de+df);

5513	46.53859	-10.03418
5272	44.84118	-9.19353
5102	29.80796	3.24399
4946	16.78139	10.01233
4916	7.21226	12.75222
4983	0.26152	13.16871
4941	-0.13088	11.28018
4822	-0.02971	10.01129
4614	0.02359	7.35848
4553	0.17499	2.59400
4674	-0.10410	-3.81215
4736	-0.51824	-12.27899
4837	1.31860	-23.12369
4912	-0.19177	-35.67028
4982	-4.76124	-43.11507
5081	-6.17583	-44.30610
5120	-3.71591	-39.74054
5207	1.10246	-30.53366
5283	-3.89560	-20.16373
5262	2.31759	-9.46280
5244	3.30076	0.36947

Experiment 05: Gromacs 5.0.4. What happens if I use the system potential energy? Storing after acceptance.

- Result: 15.21 I may have made a mistake here but this value is too low.
- dfhist->dfavg[fep_state] = enerd->term[F_EPOT]; // storing the current potential energy of the system
- de = (dfhist->dfavg[lamtrial] - dfhist->dfavg[fep_state])*(1.0/(expand->mc_temp*BOLTZ));
- trialprob = exp(de+df);

5034	46.42349	-9.50230
4897	44.91140	-8.77961
4785	29.46859	3.72579
4848	16.34707	10.33221
4950	7.28419	12.97619
5134	0.14613	13.09541
5150	0.04727	11.57926
5054	-0.44127	9.67423
5126	-0.02249	7.28384
5198	-0.10105	2.49061
5234	-0.04756	-4.22533
5139	0.78904	-12.57646
5065	1.13486	-22.36572
5120	-1.85132	-32.61183

5209 -8.63185 -41.53792
5255 -6.12834 -42.82502
5046   3.80195 -38.22974
4827   6.16335 -30.21138
4753 -11.34125 -20.02636
4676 -8.65623 -9.53720
4500 -1.52599   0.34042

Experiment 06: Gromacs 5.0.4. Sanity check: Does it matter if I stored the energy after the acceptance criteria?

- Result: 15.22915175 Turns out that it doesn't matter.
- dfhist->dfavg[fep_state] = enerd->term[F_EPOT]; // storing the current potential energy of the system
- de = (dfhist->dfavg[lamtrial] - dfhist->dfavg[fep_state])*(1.0/(expand->mc_temp*BOLTZ));
- trialprob = exp(de+df);

4720 46.67164 -9.55974
4723 44.54892 -9.11998
4801 29.09277   4.14100
4900 16.48647 10.58882
4893   7.29709 12.81365
4855   0.18891 12.87095
4911   0.02726 11.78922
5018 -0.02113   9.80926
5103 -0.09286   7.04598
5097 -0.07511   2.67346
5089 -0.35996 -3.87375
5114 -0.10286 -11.78018
5107   0.83620 -22.17092
5035 -5.49702 -33.69070
4914 -5.45985 -40.72963
4944 -3.00243 -42.74207
4922 -0.89106 -38.45092
5013 -11.02205 -30.08710
5220 -8.81478 -19.64732
5315   2.31223 -9.45410
5306   4.21790   0.43542

Experiment 07: Gromacs 5.0.7. Does Wang-Landau improve the results of the first experiment?

- Results: 19.9079025
    - I wish this was right, but I don't trust it. The count is wrong because of WL and the incrementor never changed.
- Using WL lmc-stats
- Using exp(-dy+df)
- Using dy = (enerd->enerpart_lambda[lamtrial+1] - enerd->enerpart_lambda[0])*(1.0/(expand->mc_temp*BOLTZ));

MC-lambda information

Wang-Landau incrementor is: 1

N	CoulL	VdwL	Count	G(in kT)	dG(in kT)	dGd[CoulL]	dGd[VdwL]
1	0.000	0.000	3	0.00000	-24.00000	76.18220	-18.32298
2	0.200	0.000	27	-24.00000	-199.00000	45.12551	-14.62206
3	0.400	0.000	226	-223.00000	-1148.00000	27.57755	3.06690
4	0.600	0.000	1374	-1371.00000	-2246.00000	15.24698	10.12203
5	0.800	0.000	3620	-3617.00000	-2111.00000	6.61435	13.17645
6	1.000	0.000	5731	-5728.00000	-2534.00000	0.21605	12.80661
7	1.000	0.100	8265	-8262.00000	-2838.00000	-0.12298	11.37608
8	1.000	0.200	11103	-11100.00000	-2655.00000	-0.15315	9.76199
9	1.000	0.300	13758	-13755.00000	-1061.00000	-0.03365	6.41834
10	1.000	0.400	14819	-14816.00000	1496.00000	-0.03524	2.61114
11	1.000	0.500	13323	-13320.00000	2187.00000	0.33401	-3.03374 <<
12	1.000	0.550	11136	-11133.00000	2587.00000	-0.49519	-8.73637
13	1.000	0.600	8549	-8546.00000	2637.00000	0.22367	-15.71834
14	1.000	0.650	5912	-5909.00000	2512.00000	1.03098	-23.85140
15	1.000	0.700	3400	-3397.00000	1643.00000	-5.19456	-30.80623
16	1.000	0.750	1757	-1754.00000	833.00000	-4.65111	-33.92487
17	1.000	0.800	924	-921.00000	454.00000	-2.89526	-33.08152
18	1.000	0.850	470	-467.00000	198.00000	9.24032	-26.97158
19	1.000	0.900	272	-269.00000	89.00000	41.63144	-19.40153
20	1.000	0.950	183	-180.00000	35.00000	54.89856	-9.38006
21	1.000	1.000	148	-145.00000	0.00000	52.66681	0.76958

Experiment 08: Gromacs 5.0.4. Does Wang-Landau improve the results of experiment 5?

- Result: 15.157161 No, WL does not improve the results.

Experiment 09: Gromacs 5.0.7. Does setting nstdhdl = 1 improve the results of the first experiment?

- Current Result: 17.440563
  - Looks promising. Too early to tell tho.
- Not using wang-landau

Step	Time	Lambda
524000	1048.00000	0.00000

MC-lambda information

N	CoulL	VdwL	Count	G(in kT)	dG(in kT)	dGd[CoulL]	dGd[VdwL]
1	0.000	0.000	13997	0.00000	0.00000	53.82308	-15.10604
2	0.200	0.000	21685	0.00000	0.00000	46.00475	-7.88030
3	0.400	0.000	22065	0.00000	0.00000	28.99997	4.68835
4	0.600	0.000	22727	0.00000	0.00000	16.24241	10.28425
5	0.800	0.000	23215	0.00000	0.00000	7.08095	12.34013
6	1.000	0.000	23590	0.00000	0.00000	0.01234	12.92352
7	1.000	0.100	23428	0.00000	0.00000	0.00242	11.98814

8	1.000	0.200	23191	0.00000	0.00000	0.09482	10.71347
9	1.000	0.300	23776	0.00000	0.00000	0.20480	8.16583 <<
10	1.000	0.400	24570	0.00000	0.00000	0.44669	4.55118
11	1.000	0.500	25094	0.00000	0.00000	0.65109	-1.36654
12	1.000	0.550	25842	0.00000	0.00000	0.34660	-6.59550
13	1.000	0.600	26198	0.00000	0.00000	0.11900	-14.50642
14	1.000	0.650	25993	0.00000	0.00000	1.65002	-25.29356
15	1.000	0.700	26629	0.00000	0.00000	1.07992	-37.48923
16	1.000	0.750	27397	0.00000	0.00000	-0.08937	-44.98824
17	1.000	0.800	28110	0.00000	0.00000	-3.88415	-42.23527
18	1.000	0.850	28423	0.00000	0.00000	-2.73131	-32.37637
19	1.000	0.900	28876	0.00000	0.00000	-3.37336	-20.82586
20	1.000	0.950	29479	0.00000	0.00000	-1.06159	-9.70008
21	1.000	1.000	29715	0.00000	0.00000	3.74144	0.25536

Current Problem: Where are you stuck? Waiting
Possible resolutions: What are you planning to do? Wait longer

---

Date: 04/04/2017
Current Goal(s): What are you doing? Debugging Pymbar and AIM
Update: What have you done?

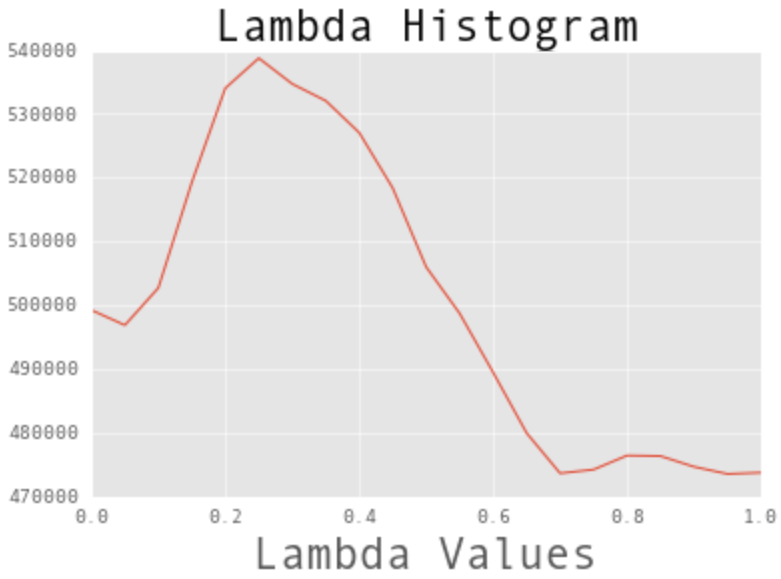
Pymbar: I debugged pymbar and found that the problem was with the autocorrection functionaligy in pymbar. I first made a bypass for the autocorrection but then I found a setting that turned off the autocorrection which then made my answer and Pymbar's match.

-i UNCORR_THRESHOLD, --threshold=UNCORR_THRESHOLD
Proceed with correlated samples if the number of uncorrelated samples is found to be less than this number. If 0 is given, the time series analysis will not be performed at all. Default: 50.

AIM: I've tried different scenarios of mdp options thinking that the wang-landau options would make a difference since it is a sort of equilibration time and it tracks the "flatness" ratio.

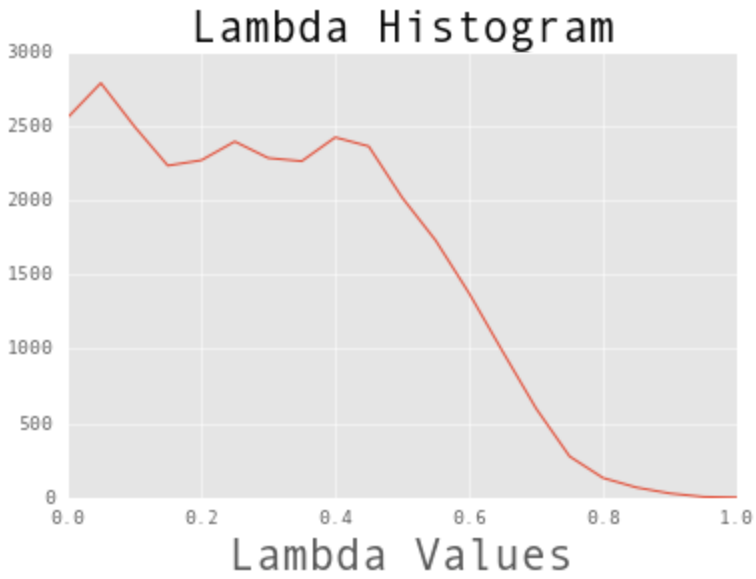
Running the simulation with no WL settings:

AIM DF = 15.8369975



Running the simulation with WL and the simulation is “WL aware”

AIM DF = 15.912575



“WL Aware” means that the when the WL algorithm resets the lambda count to zero, AIM resets the averages to zero. This ensures I don’t overestimate and takes advantage of the “equilibration”.

Both answers are similar. The benefit of using WL options is that I can set the “flatness” that I want. The simulation resets every time the flatness is achieved which acts as an equilibration. The problem with this is that the flatness won’t be measureable at the end of the simulation because the counts won’t be over the entire simulation.

Notice that the values for AIM DF are low. They should be closer to 19 units. On a whim, I decided to try calculating the average a different way:

```
dGAvg = dGdcoul + dGdvdw
myAvg = 0.0
# loop over to calculate the coulomb value
for k in range(len(coulLambdas)-1):
    myAvg += 0.5*np.dot(dlamCoul[k],(dGAvg[k]+dGAvg[k+1]))
    myAvg += 0.5*np.dot(dlamVdw[k],(dGAvg[k]+dGAvg[k+1]))

print(myAvg)

18.473761
```

Current Problem: Where are you stuck?
Possible resolutions: What are you planning to do?

Date: 03/28/2017
Current Goal(s): What are you doing? Debugging AIM
Update: What have you done?

Since Marty stated last week that the problem may be in my acceptance criteria I have gone back and checked all of the terms.

The acceptance criteria is exp(de+df) where de is the difference in the Hamiltonian in respect to lambda and df is the free energy approximation based on the average dvdl terms.

I have gone back and confirmed that the terms I am using are the decoupled dvdl terms and are the same terms that are printed out to the log and xvg file.

Output to confirm dvdl terms match those in xvg file and log file:

```
50 steps, output every step, nstdhdl = 1
step is          1
lambda is          0
myDF is    7.38518
dGd[  CoulL] F_DVDL   73.85179 linear    5.60232 non-linear   68.24947 sum   73.85179
dGd[   VdwL] F_DVDL  -16.77497 linear    1.97243 non-linear  -18.74739 sum  -16.77497
step is          2
lambda is          0
myDF is    7.37186
dGd[  CoulL] F_DVDL   73.58549 linear    5.59087 non-linear   67.99461 sum   73.58549
dGd[   VdwL] F_DVDL  -20.66039 linear    1.97245 non-linear  -22.63283 sum  -20.66038
```

I've also gone through and confirmed that I am using the correct energy terms for my acceptance criteria:

From the file expanded.c
The value I am using for de is this;
de = weighted_lamee[lamtrial] - weighted_lamee[fep_state];

Where weighted_lamee is this;
weighted_lamee[i] = dfhist->sum_weights[i] - scaled_lamee[i];

And scaled_lamee is this;
scaled_lamee[i] = (enerd->enerpart_lambda[i+1]-enerd->enerpart_lambda[0])/(expand->mc_temp*BOLTZ);

From the file force.c

```
for (i = 0; i < fepvals->n_lambda; i++)
{
   for (j = 0; j < efptNR; j++)
   {
      /* Note that this loop is over all dhdl components, not just the separated ones */
      dlam = (fepvals->all_lambda[j][i]-lambda[j]);
      enerd->enerpart_lambda[i+1] += dlam*enerd->dvdl_lin[j];
      if (debug)
      {
         fprintf(debug, "enerdiff lam %g: (%15s), non-linear %f linear %f*%f\n",
                 fepvals->all_lambda[j][i], efpt_names[j],
                 (enerd->enerpart_lambda[i+1] - enerd->enerpart_lambda[0]),
                 dlam, enerd->dvdl_lin[j]);
      }
   }
}
```

And the file forcerec.h

```
typedef struct {
   real             term[F_NRE];        /* The energies for all different interaction types */
   gmx_grppairener_t grpp;
   double           dvdl_lin[efptNR];    /* Contributions to dvdl with linear lam-dependence */
   double           dvdl_nonlin[efptNR]; /* Idem, but non-linear dependence                  */
   int              n_lambda;
   int              fep_state;          /*current fep state -- just for printing */
   double           *enerpart_lambda;     /* Partial energy for lambda and flambda[] */
   real             foreign_term[F_NRE]; /* alternate array for storing foreign lambda energies */
   gmx_grppairener_t foreign_grpp;       /* alternate array for storing foreign lambda energies */
} gmx_enerdata_t;
/* The idea is that dvdl terms with linear lambda dependence will be added
 * automatically to enerpart_lambda. Terms with non-linear lambda dependence
 * should explicitly determine the energies at foreign lambda points
 * when n_lambda > 0.
 */
```

From the file mdebin.c

```
   /* BAR + thermodynamic integration values */
   if ((md->fp_dhdl || md->dhc) && bDoDHDL)
```

```
{
    for (i = 0; i < enerd->n_lambda-1; i++)
    {
        /* zero for simulated tempering */
        md->dE[i] = enerd->enerpart_lambda[i+1]-enerd->enerpart_lambda[0];
        …..
    }
```

After re-reading through the Pymbar paper, Marty believes that we should have two energy: coulomb, and everything else (includes vdw, but could also include other stuff). Importantly, the everything else term should be the total energy minus coulomb energy. He thinks that if I set it up this way for acceptance and averages then I should be able to directly compare aim output to pymbar.

Current Problem: Where are you stuck?
Possible resolutions: What are you planning to do? I need to get the code back to where it was 4 weeks ago where the value matched Pymbar.

Date: 03/21/2017
Current Goal(s): What are you doing? Running simulations and debugging AIM
Update: What have you done? I've run 1, 2 and 5 ns per lambda for 5 simulations each.

cmira@n065 /mnt/cmci/cmira/newLambdaSched/AIM/**one** $ cat aimgpu0* | grep myDF | awk '{print $3}'
16.12797
16.46640
16.09992
16.16036
15.93042

cmira@n065 /mnt/cmci/cmira/newLambdaSched/AIM/**two** $ cat aimgpu0* | grep myDF | awk '{print $3}'
15.45615
15.52480
15.13037
15.38607
15.41420

cmira@n065 /mnt/cmci/cmira/newLambdaSched/AIM/**five** $ cat aimgpu0* | grep myDF | awk '{print $3}'
15.22232
15.34766
15.39739
15.20885
15.09506

I also found out exactly how the mdp options; nstdhdl, nstexpanded, and nstcalcenergy are all tied together. For expanded ensemble free energy calculations, frequency is the minimum greatest common denominator of nstdhdl and nstexpanded. -.-

To be more precise dhdl is only (fully) calculated at each step where free energies are calculated, i.e. every nstdhdl steps. I wanted to see if setting nstdhdl = 1 made any difference, so for 4 runs of 1ns per lambda;

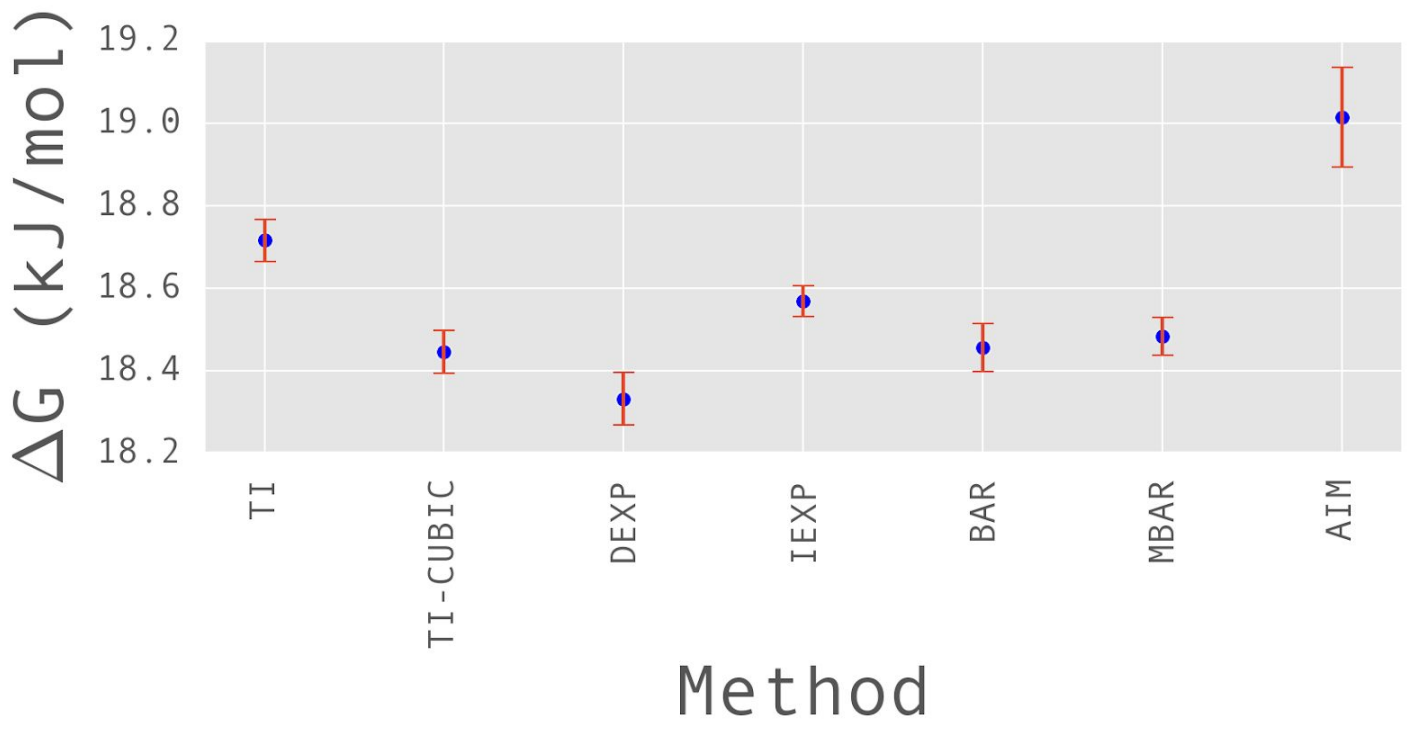
16.04457
15.82668
16.07101
15.93372

The unit here is kJ/mol for all of the above numbers.

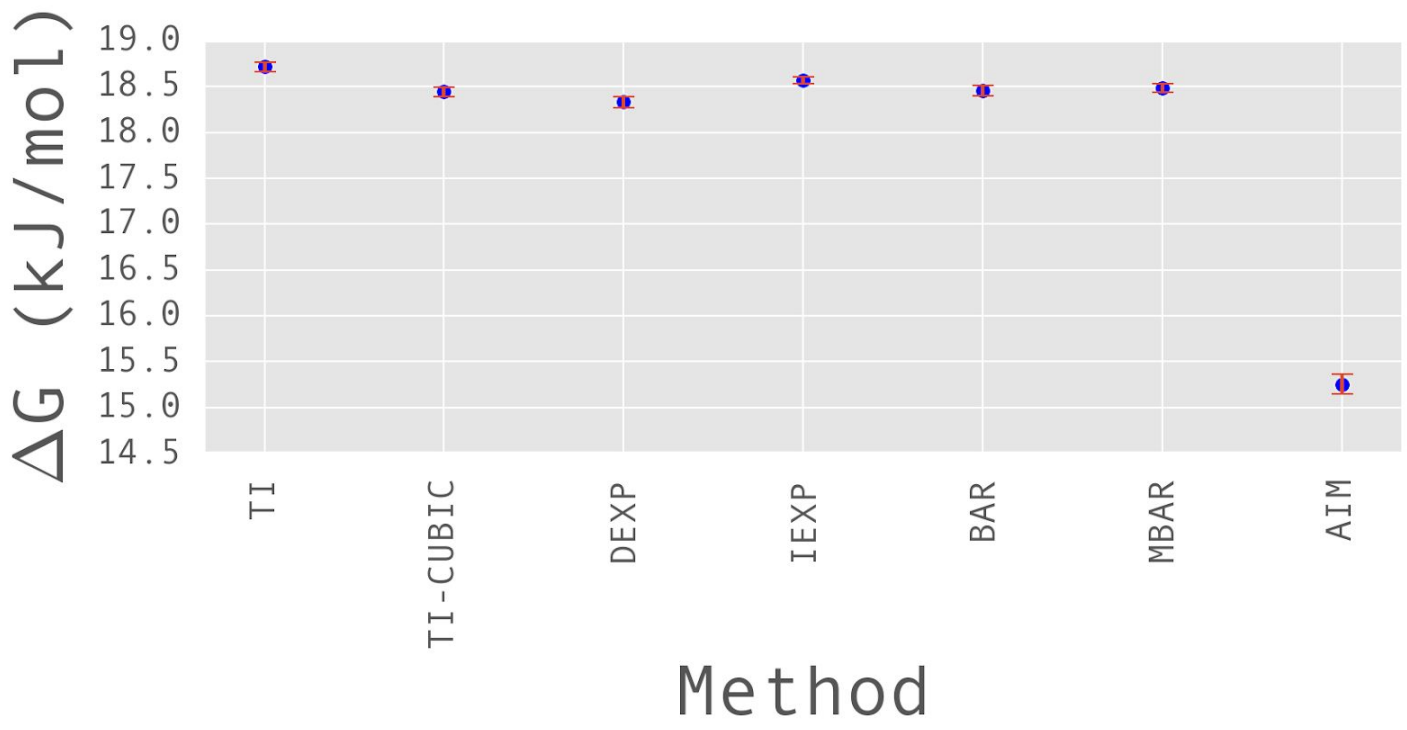
As can be seen from the output of one of the 4 simulations, the histogram is reasonably flat.

N	CoulL	VdwL	Count	G(in kT)	dG(in kT)	AIM Count	dGd[CoulL]	dGd[VdwL]
1	0.000	0.000	4	0.00000	0.02344	**455651**	56.85963	-19.45934
2	0.200	0.000	35	0.02344	0.01340	**475139**	42.00515	-4.73786
3	0.400	0.000	131	0.03683	-0.01420	**488153**	28.12801	5.09610
4	0.600	0.000	250	0.02263	-0.02422	494524	15.83910	10.62942
5	0.800	0.000	363	-0.00159	-0.01209	494324	6.90015	12.88567
6	1.000	0.000	437	-0.01368	0.00633	491166	0.03732	13.38842
7	1.000	0.100	464	-0.00735	0.00396	490872	0.03384	12.02502
8	1.000	0.200	453	-0.00339	0.00636	492727	0.02113	10.20869
9	1.000	0.300	504	0.00297	0.01630	495212	-0.00598	7.65316
10	1.000	0.400	545	0.01927	0.02696	498630	0.02279	3.65952
11	1.000	0.500	454	0.04623	0.00903	504907	0.03841	-2.99602
12	1.000	0.550	357	0.05526	0.01011	506964	0.19870	-8.23126
13	1.000	0.600	290	0.06538	0.01184	509103	0.37959	-15.61401
14	1.000	0.650	286	0.07721	0.00409	512448	-0.60648	-25.80568
15	1.000	0.700	395	0.08130	-0.01187	514522	-2.30234	-38.20147
16	1.000	0.750	453	0.06944	-0.01771	514689	-5.66234	-45.78705
17	1.000	0.800	466	0.05172	-0.01014	511954	-5.98211	-42.87374
18	1.000	0.850	472	0.04158	-0.00543	511205	-5.96278	-32.96725
19	1.000	0.900	456	0.03615	-0.00657	511329	-4.22154	-21.14546
20	1.000	0.950	427	0.02958	0.00058	512284	-2.86820	-9.82763
21	1.000	1.000	434	0.03016	0.00000	514197	-0.88023	0.14855

What started this was the graph:



Now the graph looks like this:



The error looks the same to me. Mind the scale.

Current Problem: Where are you stuck? The free energy should be around 19-20 kJ/mol. For 1ns lambda windows, Pymbar reports a value of 18.66 kJ/mol.

Possible resolutions: What are you planning to do?

Date: 03/07/2017
Current Goal(s): What are you doing? Being sick.
Update: What have you done? Slept
Current Problem: Where are you stuck? In bed
Possible resolutions: What are you planning to do? Sleep more

---

Date: 02/28/2017

Current Goal(s): What are you doing? Debugging AIM

Update: What have you done? I've gone through the code and confirmed that I should be using the enerd->dvdl_lin and enerd->dvdl_nonlin term by printing them out and comparing the values to the values in the log. I've made it to where my output goes into the log file instead of stderr. My output is now the average of the individual dvdl terms such as dv(coul)dl or dv(vdw)dl. I also made sure that the same calculation is being done for the acceptance criteria which was brought up during a conversation with Marty.

Current Problem: Where are you stuck?

Suppose overall free energy change for some reaction is ΔG

How does ΔG change with changes in constituents; i.e. coulomb or vdw energy components?

One would need to integrate ΔG with respect to these changes
(whatever the changes might be but call them x, y, z, etc)

$d\Delta G = (\delta\Delta G/\delta x)\, dx + (\delta\Delta G/\delta y)\, dy + ... (\delta\Delta G/\delta z)\, dz$

Imagine instead of changing a number of variables (x, y, z), only one variable is changed and everything else is kept the same

$\Delta Gi - \Delta Go = (\delta\Delta G/\delta x)(xi-xo)$

In my case, x is lambda coulomb, y is lambda vdw. There could be others, but typically these are the only variables. The mdp settings would be something like:

Coul-lambdas = 0.0 0.2 0.4 0.6 0.8 1.0 1.0 1.0 1.0 1.0 1.0 1.00 1.0 1.00 1.0 1.00 1.0 1.00 1.0 1.00 1.0
Vdw-lambdas = 0.0 0.0 0.0 0.0 0.0 0.0 0.1 0.2 0.3 0.4 0.5 0.55 0.6 0.65 0.7 0.75 0.8 0.85 0.9 0.95 1.0

So does this mean that when vdw-lambda is zero, there is no vdw contribution to ΔG? Or does it mean that when (xi-xo)=0, where xi is the index of vdw-lambdas, then there is no contribution?

I think it's the latter. I think I should be storing the average of $\Delta G/\delta x$ regardless of the value for vdw or coulomb but I'm really confused and need clarification.

Possible resolutions: What are you planning to do? Ask Marty and the group.

Date: 02/21/2017
Current Goal(s): What are you doing? I've been running AIM simulations and debugging
Update: What have you done? I've run several testing simulations trying to figure out the difference in running simulations on COAN and fourtyfour trying to figure out why I get different values in 5.0.4 versus 5.0.7. I thought there was a bug in my code but I found out that using npme -1 in my mdrun script on 5.0.7 (the only difference in my simulation settings) doesn't work on GPUs. I struggled with this one for several days and finally found this in the 5.0.6 release notes:

"Don't use PME ranks with GPUs and mdrun -npme -1. The code disabling the automated PME rank choice with GPUs was accidentally moved after init_domain_decomposition(). This caused PME ranks to be set up, but later a fatal_error occured for inconsistent PP rank and GPU counts.#1374."

Using npme -1 can optimize PME ranks but really only helps if you are using multiple nodes or multiple GPUs which I was doing at one point. Removing the npme -1 mdrun option fixed the odd behavior that I was seeing.

Current Problem: Where are you stuck? I'm now trying to figure out the value in using the average of all dvdl's versus using the average of the individual dvdl's.
Possible resolutions: What are you planning to do? I started a conversation with Marty over Slack to see what he thinks but there is a lot to look over.

Date: 02/14/2017
Current Goal(s): What are you doing? Debugging AIM. Running simulations.
Update: What have you done? Testing AIM simulations. AIM has been patched to Gromacs 5.0.7 and is running on the fortyfour cluster. I've tested the batch job submittal using Slurm and that works now. I was having a problem but I worked with Benji to figure it out. My problem was that I could only submit one job per node. It turned out that you have to specify an amount of memory when starting the job so each of my jobs was taking all of the memory of one node and I was unable to start multiple jobs on a node.
Current Problem: Where are you stuck? The calculation for AIM isn't right but I know why. The lambda count is being reset due to mdp options.
Possible resolutions: What are you planning to do? I'm going to create my own count variable like I did on COAN. I didn't do that at first because I was hoping that the count in Gromacs would work but it doesn't.

Date: 02/07/2017
Current Goal(s): What are you doing? AIM simulations
Update: What have you done? AIM now matches the output from pymbar.

myDF is 5.69769
pymbar 5.712 +- 1.292

Difference matrix:

```
array([ -1.35740775e-01,  -8.34480631e-06,  -7.44963134e-06,
         1.60562952e-05,  -8.82899521e-07,   3.63852133e-06,
        -2.31830622e-06,  -9.25014353e-07,   9.42500191e-06,
         1.16487794e-06,  -4.81137555e-06,  -1.16888412e-06,
         2.42711111e-06,   3.62233192e-06,   3.24905350e-06,
```

-1.02838095e-06, 7.26238282e-06, -1.56729071e-05,
 1.28341844e-05, -7.22598034e-06, -1.18617501e-05])

Now I can restart the AIM sims.

Just in case, I re-ran and checked against pymbar:

pymbar = -41.223 +- 2.590
myDF is -41.24383

Current Problem: Where are you stuck? Not currently stuck
Possible resolutions: What are you planning to do? Run sims

Date: 01/31/2017
Current Goal(s): What are you doing? Trying to figure out why AIM doesn't match TI
Update: What have you done? I figured out that I had the wrong index value for an array that I was using for my calculations.
Current Problem: Where are you stuck? I need to iteratively multiply delta lambda by the correct energy term, coulomb, vdw, etc. There are up to 6 energy values. Each energy value needs to be multiplied by delta lambda.
Possible resolutions: What are you planning to do?

1. myDF = 0.5*(lambda_new - lambda_old)*(<E>_ln + <E>_lo)

E = [fep, coul, vdw,...]
coul-lambdas = 0.0 0.2 0.4 0.6 0.8 1.0 1.0 1.0 1.0 1.0 1.0 1.00 1.0 1.00 1.0 1.00 1.0 1.00 1.0 1.00 1.0
vdw-lambdas = 0.0 0.0 0.0 0.0 0.0 0.0 0.1 0.2 0.3 0.4 0.5 0.55 0.6 0.65 0.7 0.75 0.8 0.85 0.9 0.95 1.0

All_lambda[j][i] where j is an energy index and i is a lambda index

All_lambda[2][2] = 0.4

enerd[F_DVDL+2] = coulomb dvdl

enerd[F_DVDL+3] = vdw dvdl

myCoulDF = 0.5*(all_lambda[2][i+1] - all_lambda[2][i])*(enerd[

myVDWDF=

Total = myCoulDF + myVDWDF

Track each energy average separately

Date: 01/24/2017

Current Goal(s): What are you working on? AIM Research. I'm trying to compare AIM to TI using the Pymbar package.
Update: What have you done? I have reproduced the output of pymbar in my own python script. I have written code to compare the averages of a short simulation.
Current Problem: Where are you stuck? I have the results and it looks pretty good. The differences are very small.
Possible resolutions: What are you planning to do? Ask for more input and get back to writing.

Presentation:
R vs. Python
Infographic: https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis#gs.YXniGCE
Plotting: https://www.dataquest.io/blog/python-vs-r/

Date: 01/17/2016
Update: Tried to write a little bit everyday. Re-generated figures for my paper. Also, I ran simulations to test whether or not TI calculations agree with AIM but I didn't correct the nstexpanded option so I am rerunning the simulation again. I am running simulations on COAN's GPU.

Based on quick sims, my quadrature algorithm may be wrong. I'm running the simulation with more steps to see what happens.

Agenda:

Jan. 24 Presentation -- ChrisM
Jan. 31 Jagdish Presentation and/or poster
Feb. 7 Caleb Presentation
Feb. 14 No Meeting
Feb. 21 Erin Presentation
Feb. 28 Zahid Presentation
Mar. 7 No meeting
Mar. 14 Spring Break No Meeting
Mar. 21 Johnathan Presentation
Mar. 28 Chris Caleb Brainstorm
Apr. 4 Kyle Prelim 2 practice presentation
Apr.11 Tentatively Jagdish Zahid Brainstorm
Apr. 18 John Erin Brainstorm
Apr. 25 Kyle Brainstorm
May 2 Open
May 9 End

Other questions: Is TI in pymbar rectangular quadrature? Double check that it's rec quad. Should get the same number.
If not the same algorithm, output dg between each lambda.

Date: 12/08/2016

Update: The reality is that I haven't had time to work on research this last month. I need to run simulations that I can use to compare TI to AIM. This would be a simulation that outputs every step of an expanded ensemble simulation so I can compare the two. I also need to get AIM coded and running on the FortyFour cluster and run the mutation simulations, alanine to valine.

For brainstorming I would like feedback on the outline of my paper. Here is the link:
https://drive.google.com/open?id=10EXEHp2MYyfTEDnInJm4kfILOzqHe7Vnv96CFR1btGM

I already know that the images aren't consistent in the way they represent the x axis and I don't really have any questions. The whole document is open for discussion.

Date: 12/01/2016
Update: No update. Benji says that we have the ability to run jobs on the GPU's using slurm but I haven't tested it.

Date: 11/10/2016 - 11/17/2016
Update: No update since I was out of state all week. Trump is president. The world didn't catch fire (give it time).I wrote a little bit. Simulations ran.

Date: 11/03/2016
Update: Simulations are running. The 1ns, 2ns and 5ns lambda windows have run. The 10ns is running. I will not be here next week.

Date: 10/27/2016
Update: I've been stalled due to problems with the slurm job scheduler. When I submitted a job there was no output, no log file. It's like the program wasn't running. I could see the job in the queue but there were no output files. I expected there to be a log file but there wasn't. And the job should have been done in less than a minute, but had been in the queue for 10 minutes. I asked Benji for help. We spent an hour Tuesday poking at the problem and he found that the cgroup locks weren't being released properly. Benji spent pretty much all day yesterday on this - and no combination of configuration settings seems to resolve the issue. He updated slurm to the newest minor version without any change in the behaviour.

It appears to be an open bug

https://bugs.schedmd.com/show_bug.cgi?id=2493
https://groups.google.com/forum/#!topic/slurm-devel/GcWWx0ymlJk

For now, don't use the --gres=gpu:1 flag, and we'll just have to coordinate who is using the GPU's on each node at any given time.

However, gromacs is no longer detecting GPUs when using the job queue.

If I submit a job using sbatch, Gromacs doesn't see GPUs. However, if I ssh to the node and run my script then the GPUs are detected. This is very annoying and frustrating. I have emailed Benji.

Date: 10/20/2016

Update: I have recompiled GROMACS 5.0.7 and have run some tests. Simulations are running now. MTTK with constraints is deprecated as of 5.1. I have no idea how to change my simulations since they were taken from tutorials.

Mohammed asked me to switch presentation days with him so today I will be presenting my research to the group. I'm just going to over my poster from the EAC.

Some things I may bring up: http://www.alchemistry.org/wiki/Main_Page

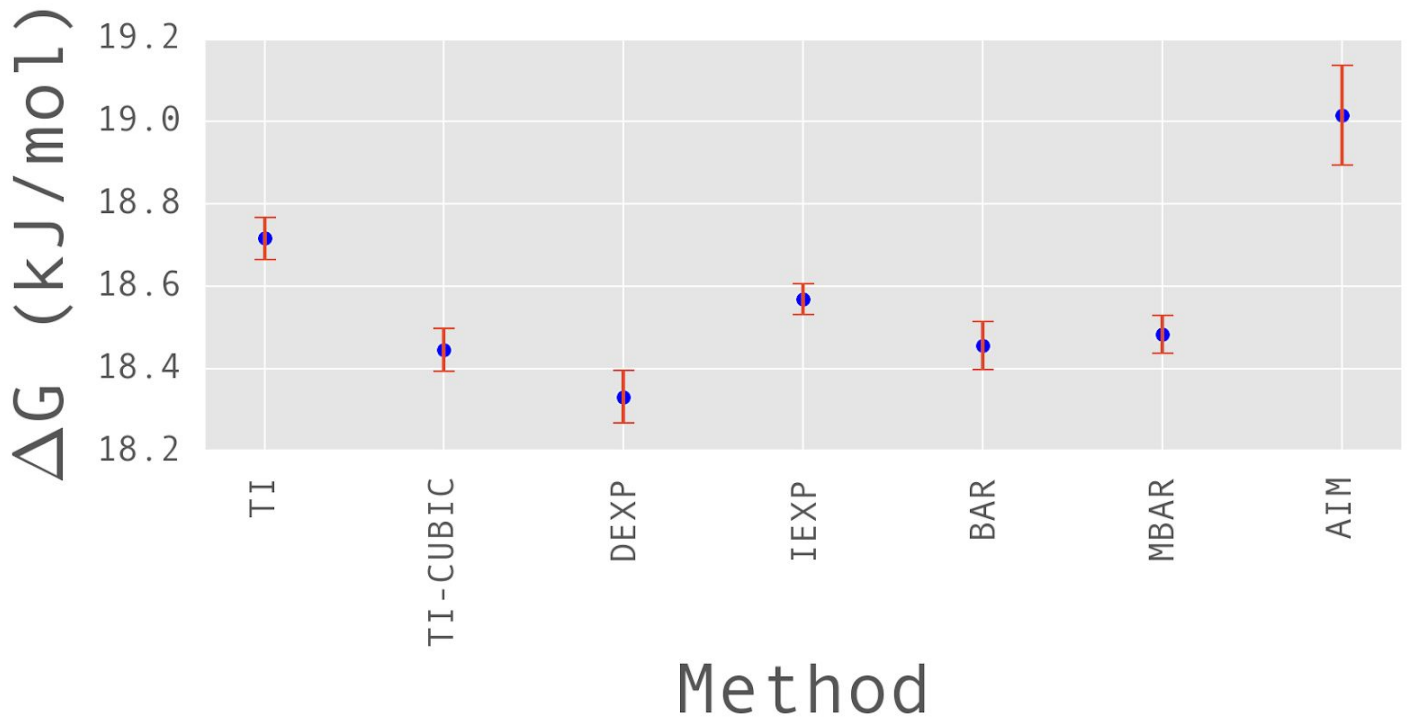
---

Date: 10/12/2016

Update: Ran into a problem with sharing GPUs on fortyfour. Sent an email to Benji to try and resolve the issue of being able to specify node, GPU and number of CPUs so we can share GPU resources on one node. We don't want the node to report as busy if only one GPU is being utilized.

I went to run AIM simulations on fortyfour using gromacs 5.1 and got an error. I was using the same mdp and topology files from 5.0.4 but now they don't work. After speaking to Marty, I am going to downgrade back to 5.0.4 and see if that fixes everything.

The simulation that I ran is the "direct" sims with the current lambda schedule and 5ns lambda windows. This is averaged over 5 runs.



My questions:

For analysis, are there standards or best practices? Such as, how many steps should I ignore from a production simulation? Remove 50 percent. Do this soon and respond to Marty.

Should I do an alternate thermodynamic cycle? Do this.

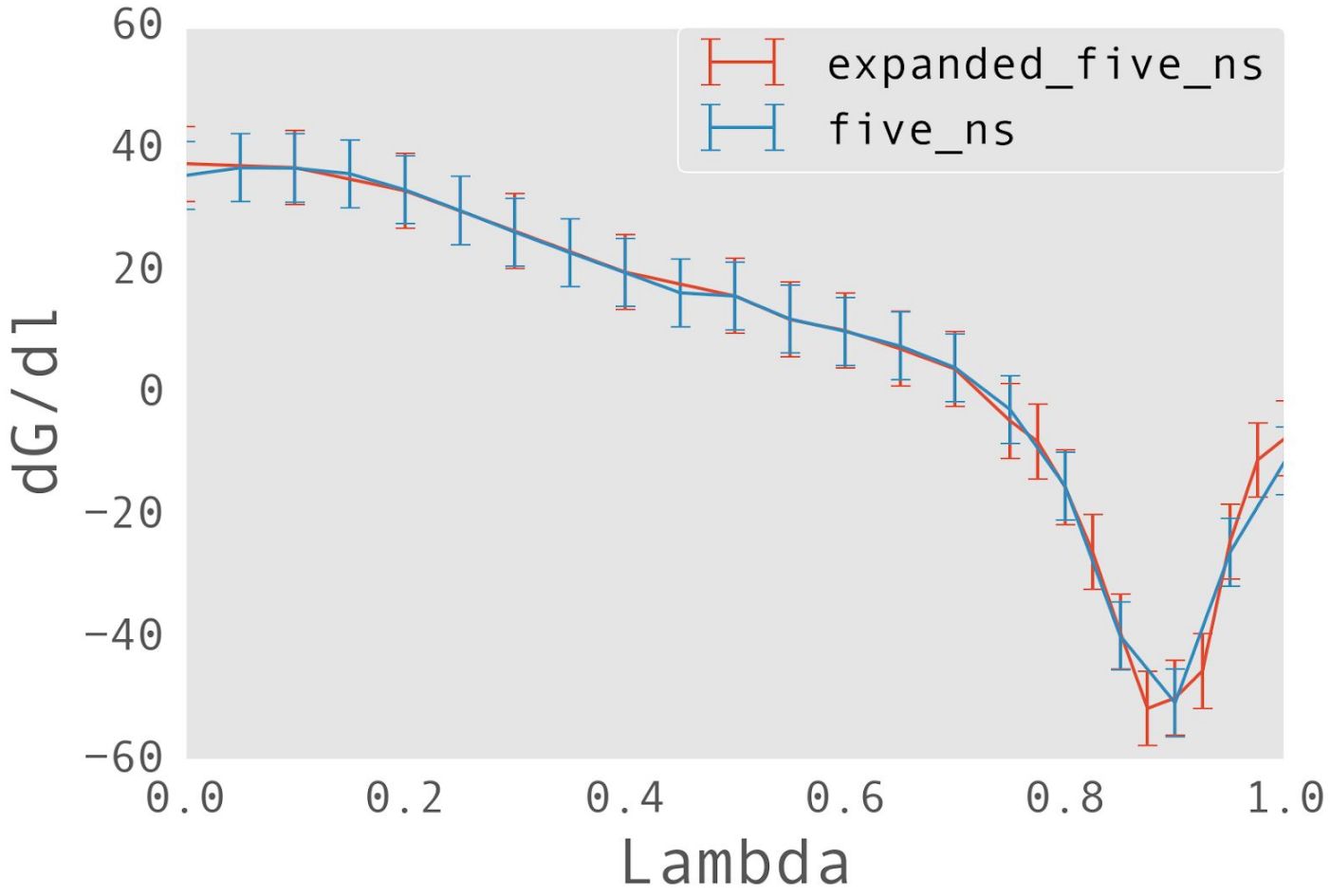
What journal should I be writing for? Wait to see more results before deciding.

ToDo: look at histograms of AIM outputs to see if they are converged.
Run 1ns, 2ns, 5ns, and 10. Wait for 20ns.

Simulations are done in “serial”.
If I output AIM for every step, what is the result for analysis from TI? If the answer is different then there is possibly a mistake in AIM calculation. Because not within uncertainty.
Can start writing the methods section since nothing big should change.

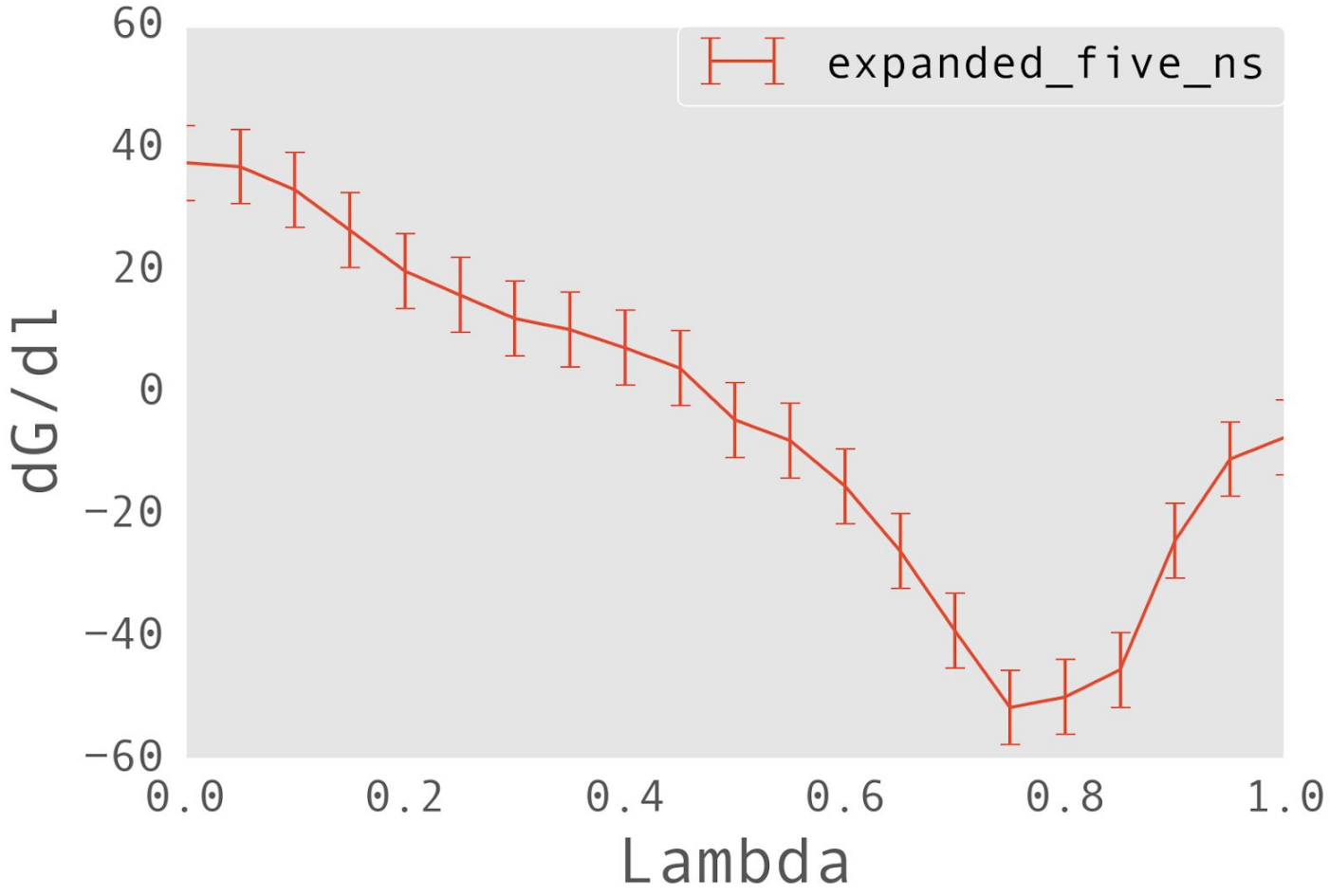
Date: 10/06/2016
Update:I have been working on the EAC poster and working on getting Gromacs installed and tested on the fortyfour cluster. I have run a very simple simulation of Ethanol in water on the GPU of n065. Everything looks correct and Gromacs has recognized the GPUs. My next task is to test AIM. Since I converted over to Gromacs 5.1, I reorganized the code to better match the way things are done in 5.1. Once I’ve tested this I plan on running “direct” simulations to match the current lambda schedule. N065 has 4 GPUs. Using 1 GPU and 1 CPU, I managed 133 ns/day with my test simulation. Since there are 4 GPUs and 40 cores I expect the simulations to go quickly. I have a skeleton paper. Even though my current results don’t match the “preferred” lambda schedule, I’m going to use them as place holders. I’m not a confident writer since I don’t have any experience writing journal papers but I’m trying to write a little bit everyday.

Date: 09/29/2016
Update: Worked on poster. AIM code has been compiled for fortyfour. I had to work with Benji to get everything I needed. I also have the graph that Marty asked comparing the 5 ns runs with the expanded “dip”.



Date: 09/22/2016
Update: I'm working on transferring the AIM code to GROMACS 5.1 and over to the fortyfour cluster. The code is written and I'm working through compile errors. I'm also slowly writing as I go but I need to think about what results I should talk about.

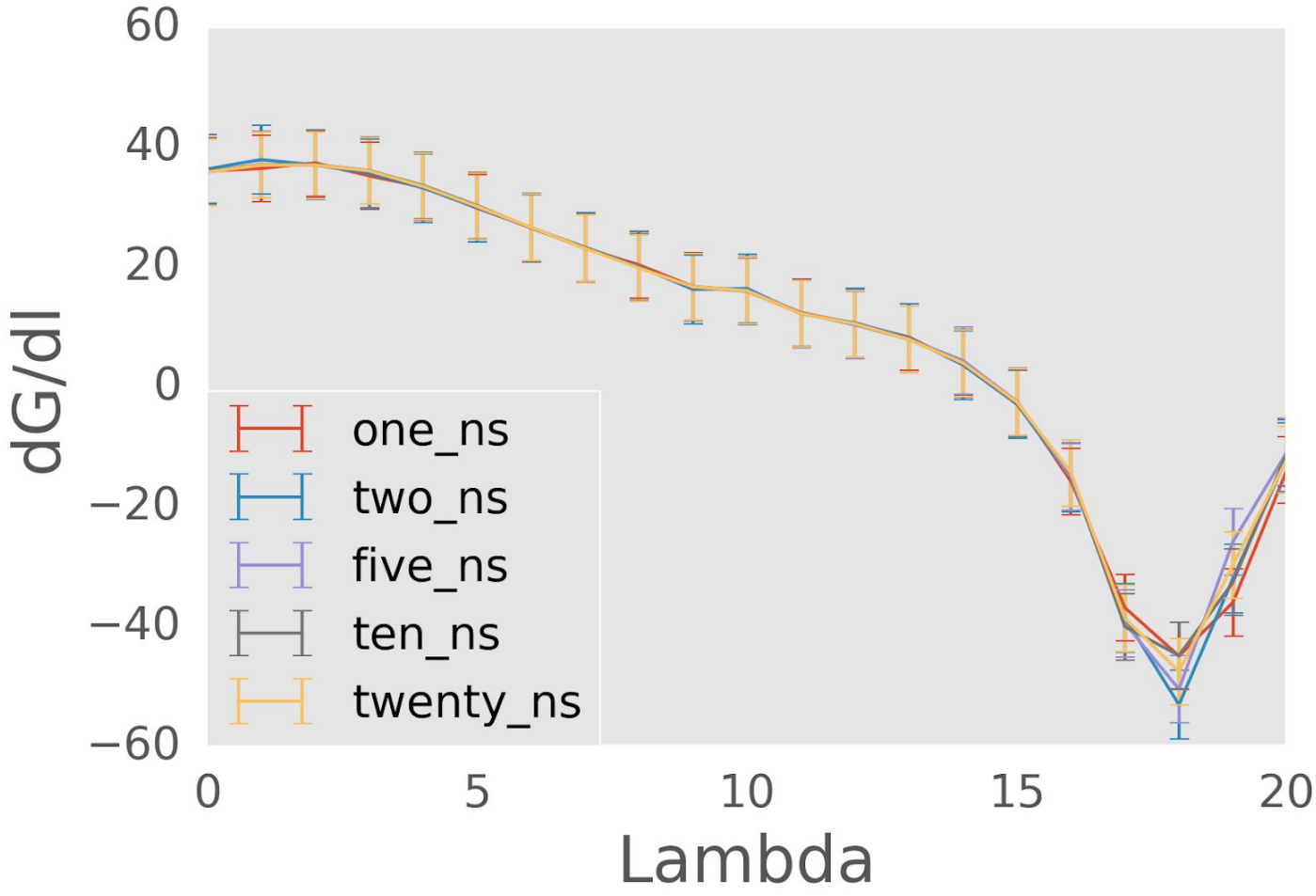
Date: 09/15/2016
Update: I have tested Pymbar for the analysis. It seems to be working on my laptop and on COAN. I have been trying to write a little bit everyday. I also have the results from the simulations to look at the "width" of the dip. This is averaged over 5 runs. There doesn't seem to be a huge difference in the bottom point of the well from the previous results.



Date: 09/08/2016
Update: I am running simulations of the same type as last week but this time with the width of the dip 5 lambdas wider. The simulations are still running. Once they are complete I will graph the output for analysis.

Date: 09/01/2016

Update: Testing convergence for AIM using different sized time windows for each lambda. Basically the same as previous update but now with 20 ns. These are averaged over 5 runs for each. The 20 ns run just finished.

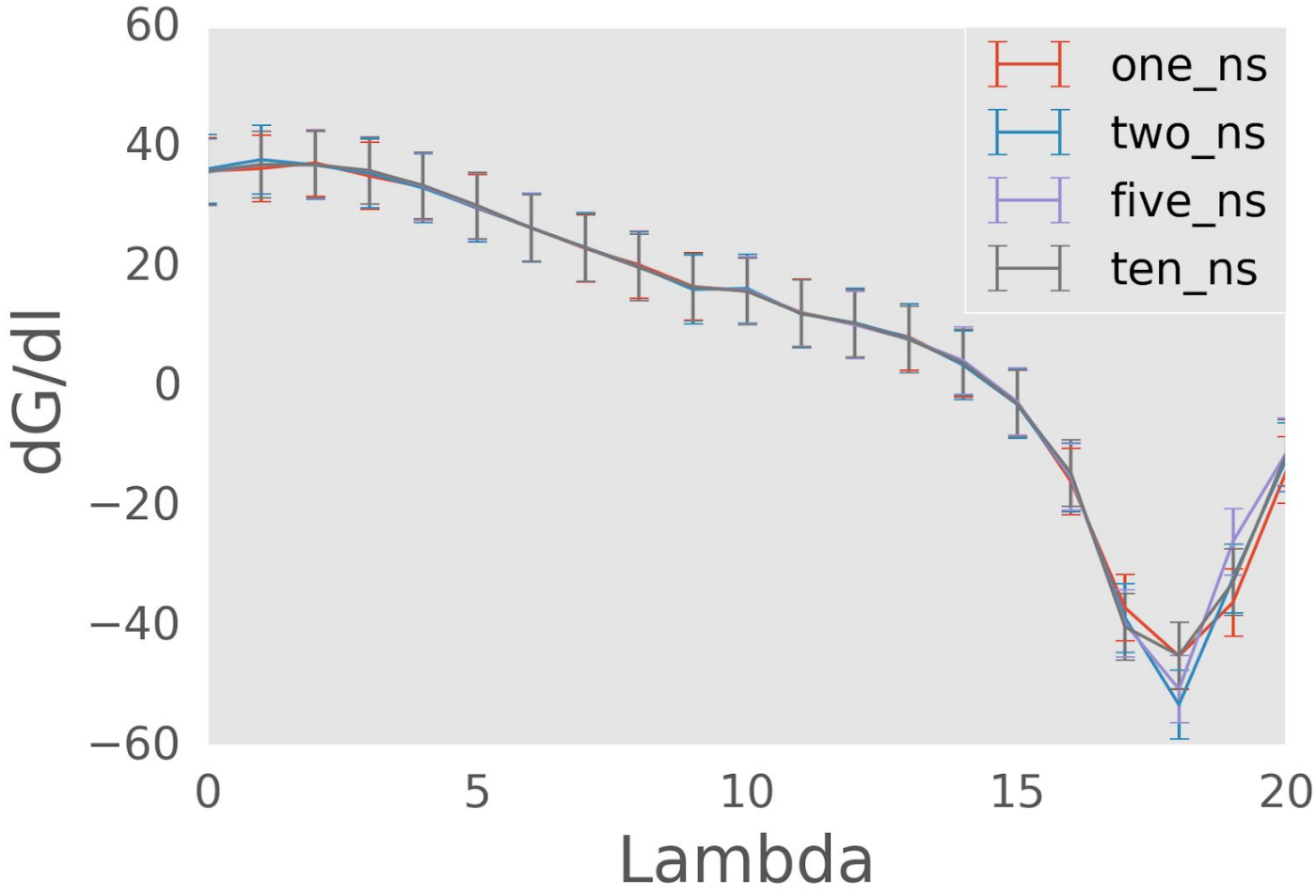


Date: 08/10/2016

Update: Testing how many ns per lambda is needed for AIM to converge to a result. I've run one, two, five, and ten ns per lambda simulations of ethanol with 21 lambdas. The ten ns simulation isn't complete but I have 4 of the 5 runs complete. We wanted 5 for an average of each run. For the four runs of each simulation I have averaged the results and plotted them below. I think I need to change the values of the lambdas (keep the same number of lambdas) between 15 and 20. But I'm not sure. Basically, I think there needs to be more lambda search space between 15 and 20. I would simply change the values of the vdw and coul lambdas.

The current values are:
coul-lambdas = 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
vdw-lambdas = 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

And I think I could change them to:
coul-lambdas = 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
vdw-lambdas = 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.2 0.4 0.5 0.6 0.7 0.75 0.8 0.85 0.9 1.0



These are averages.

Date: 07/27/2016
Current Goal(s): What are you working on? Testing AIM. Teaching at LCSC
Update: What have you done? Running simulations of different times for 21 lambdas to find when the simulations converge

---

Date: 07/20/2016
Update:
I've been testing different mdp options for AIM simulations. I still have the 'dip' in my output. I've tried different soft core parameters and I've run longer simulations with more lambdas.

This site;

http://www.alchemistry.org/wiki/Best_Practices

provides guidelines for alchemical simulations. It's really good and the team should probably read over it. Maybe Marty could tell us whether or not he agrees with everything there.

This pdf from a conference this last May brings up some really good points and things that I've been thinking about for AIM simulations and the paper:

http://www.alchemistry.org/wiki/images/9/96/Sampling_Discussion.pdf

The question I'm currently working out is how to set up soft core potentials:

http://www.alchemistry.org/wiki/Constructing_a_Pathway_of_Intermediate_States#Soft_Core_Potentials

The settings that I am currently using and that I used for any simulations that I discuss below are;
sc-power = 1
sc-alpha = 0.5
sc-r-power = 6

Marty suggested that I use sc-alpha = 0, but the system became unstable and spit out an error. Before changing the soft-core parameters, Marty suggested that I run longer simulations with more lambdas.

Even after using more lambdas and more steps, AIM has the "dip" in the last few lambdas.

Step 50000000
myDF is 19.57400

Lambda	Count	dG
0	22965	35.88552
1	23465	37.11775
2	23366	36.87073
3	23317	35.88832
4	23327	33.46716
5	23438	29.96007
6	23963	26.45360
7	24038	23.07173
8	23880	19.72313
9	23805	16.70285
10	23320	15.96691
11	22916	12.28992
12	23275	10.62676
13	23605	7.66606
14	23466	3.91690
15	23976	-2.52552
16	24464	-14.29925
17	24622	-37.14558
18	24690	-46.21323
19	24923	-28.93780
20	25179	-5.60654

Notice how lambdas 16 through 19 make a dip into the negatives but come back up. My question is, what should we expect it to look like?

I probably missed the point but I think the downward spike in the output was what we want to see with a free energy simulation.

From what I've read here,
http://www.bevanlab.biochem.vt.edu/Pages/Personal/justin/gmx-tutorials/free_energy_old/07_analysis.html

I expect the downward dip. Does it just need to be smoother?

Date: 07/06/2016
Current Goal(s): What are you working on? AIM simulations and writing. I have everything prepared to see to look at results for simple toy systems and a protein system where I can simulate an amino acid mutating from one type to another.
Update: What have you done?
Conference in D.C. last week. I changed the poster:
http://nisbre.ipostersessions.com/default.aspx?s=75-40-E5-10-09-38-C1-DC-98-42-8C-E2-D5-ED-94-0D&guestview=true

AIM:
I added code to print out the dG so I can plot the average and standard error of dgdl as a function of lambda.
I'm running simulations of ethanol, direct and expanded ensemble, in order to compare AIM to TI and bar.
Once these simulations are done I need to simulate the amino acid mutation of alanine to valine.

Date: 06/22/2016
Current Goal(s): What are you working on? iPoster
Update: What have you done? Deleted it and started over

Date: 06/15/2016
Current Goal(s): What are you working on? iPoster presentation, Ebola hypothesis
Update: What have you done?
http://nisbre.ipostersessions.com/default.aspx?s=B7-5B-B3-F0-57-CE-73-17-42-29-17-D7-2B-91-34-02&guestview=true

Ebola Hypothesis
Sequence hydrophobicity and cellular environment determine flexibility in the intrinsically disordered mucin-like domain of Ebola Glycoprotein.

Supporting Research:
Amino acid substitutions are constrained differently in different local environments as defined by the secondary structure, solvent accessibility and hydrogen bonding. Hydrophobicity and relative solvent accessibility are strongly correlated with positive selection. Several authors (I need references) have claimed that either the flexibility matters or it doesn't, but since no one has actually measured flexibility, it is timely to do so.

How I plan to test it:
I plan to select temporally separated sequences from each type of host (maybe I'm saying this incorrectly, I want to mimic what the Epistasis paper did).

MD is a huge limiting factor.
There are several approaches mentioned in papers that Jagdish has sent :

1) Predicting flexibility by a fast method without MD
2) Protein + Implicit solvent to explore conformational space and speeding up the calculation
3) Protein + Explicit solvent

I also would like to use hydrophobicity scales (available on expasy.org and in literature) to visualize the differences in sequence hydrophobicity across different host sequences that are available in sequence databases. I need help finding these sequences because I need to look at as many hosts as possible including in vitro sequences. How do I search for in vitro sequences? This is not obvious to me. I will continue to figure this out.

The overall idea (as suggested by Jagdish and from previous discussions) would be to compare how the mucin like domains have evolved over time and determine what features have changed. At the same time, looking at lectins in bats and other hosts and determine what features are different. Comparing these to see if there is any correlation.

I think this can be accomplished by looking at solvent accessibility and secondary structure of the predicted structures and features of transmembrane proteins in other hosts. From there we can infer the biological role of flexibility in the mucin-like domain.

Finding what's different about lectins per host should be determinable without me having to measure anything. If not, then I don't think it is necessary, except as a way to infer selective pressure. The hypothesis implies that I will be looking at hydrophobicity and structure prediction in the MLD. There isn't actually anything stating that I will determine the origin of selection, just features of positive selection and flexibility. Lectins/Selectins is entirely speculative at this point but is mentioned in Celeste's paper and from research, seems to be an arguable/plausible candidate.

Current Problem: Where are you stuck? Not really stuck. Marty has suggested working on generating a list of sequences that I would like to test from different hosts and time frames. We need to get an idea of how many sequences we would need to calculate flexibility for to get a shot at testing your hypothesis. The number of sequences will determine how we need to proceed with calculations.

Possible resolutions: What are you thinking about? I should add current results to the poster, maybe in the discussion?

Date: 06/07/2016
Current Goal(s): What are you working on? AIM simulations and Ebola hypothesis
Update: What have you done? AIM methanol and ethanol solvation free energy simulations
Results

Methanol Expected value is -9.37 kJ/mol

Output from AIM shows that I am getting -9.46 kJ/mol and the histogram is relatively flat. I have mdp options using nstdhdl = 100, nstexpanded = 100, nstcalenergy = 100.

```
 Step 1527400
Make sure dhdl is not zero. dhdl =  -38.48342
No multiples, pure myDF is   -9.46615 kJ/mol
The count at each lambda
 0   729
 1   718
 2   725
 3   734
 4   736
 5   737
 6   749
 7   752
 8   741
 9   741
 10   743
 11   735
 12   728
 13   726
 14   715
 15   707
 16   711
 17   707
 18   713
 19   719
 20   708
```

Ethanol solvation free energy
Expected output is around 21 kJ/mol -- can’t find the reference at the moment

AIM out has the free energy at 21.25 kJ/mol.

```
 Step 13906100
Make sure dhdl is not zero. dhdl =    7.92085
No multiples, pure myDF is   21.25175
The count at each lambda
 0   1368609
 1   1362662
 2   1365428
 3   1369507
 4   1367469
 5   1368055
 6   1383606
 7   1420140
 8   1424614
 9   1476010
```

Ebola hypothesis. I've sent a hypothesis for review to Celeste and Jagdish hoping that they can dissect it when they have the time, otherwise I'm taking a small break from Ebola.

Current Problem: Where are you stuck? Not stuck. Need to make a presentation for the upcoming conference in D.C. They want a 'rough draft' by June 13.

Possible resolutions: What are you thinking about?

Date: 6/01/2016
Current Goal(s): What are you working on? AIM and Ebola Evolution, presentations, poster for conference

Update: What have you done? Added delta lambda function to AIM estimate and AIM seems to be working correctly. I need to test methanol and another toy model.

See Ebola Presentation
https://drive.google.com/open?id=1o31IeiK0l0SYmsVATX6sIy1LZHrWcDtgNFdgPJ-a0mE

Current Problem: Where are you stuck? Found a paper that seems to be the higher level, as in less detailed, version of my research with Ebola. Not sure how it changes my research.

Possible resolutions: What are you thinking about?

Date: 5/25/2016
Current Goal(s): AIM and Ebola simulations
Update: I've been running test simulations on the INL cluster and I've been working on getting AIM to output the expected value. It's been a lot of 'write some code, compile, run a sim, watch some Netflix, check the output, debug for more output, write some code…' I did find the solution of one of the problems I was having with AIM, I found the value for dhdl in Gromacs. That fixed the problem I was having with mdp files using separate lambdas (fep, vdw, coul…) and getting zero for dhdl. Now it doesn't matter what the user uses for lambdas.
Current Problem: It seemed that AIM was creating output in the wrong units but I've gone through and looked at every piece of the code very closely. All of the values being input to AIM are in kJ/mol, so the output should also be in kJ/mol. I'm really stuck on this and have been for a while.
Possible resolutions: Maybe if I could figure out how pymbar is calculating the value for thermodynamic integration I could figure out why my output is wrong. I dunno.

Date: 04/27/2016

Current Goal(s): What are you working on? I am working on running simulations on INL computers. And analysis of graphs of hydrophobic scales on Ebola GP. Too many to paste on here.

Update: What have you done? Written code that scrapes the ProtScale website to pull all of the available protein scales. Some of them aren't necessary. Wrote code that applied the scales to the sequences and generated graphs.

I've also written code that builds the extended peptide sequences and generates pdb files.

Current Problem: Where are you stuck? I need to setup my home directory on the INL server. I need to install python libraries there and start testing. I will be generating a lot of data and need to consider the logistics of drive space. I also don't know the limitations on drive space.

Possible resolutions: What are you thinking about? The hydrophobic scales has generated too many graphs to look at and compare side-by-side. I need a better way of comparing the output and I need to research the differences between the scales. Some of them do not show any differences in the hydrophobicity of different ebola gp sequences.

Date: 03/09/2016

Current Goal(s): What are you working on? Ebola glycosylation prediction using different available servers. Modpred.org and Glycomine (http://www.structbioinfor.org/Lab/GlycoMine/).

Update: What have you done? When we used the NetOGlyc server there were two ebola mutants that only had one mutation between them, A to T at site 448. The prediction from NetOGlyc showed there were a large number of o-linked glycosylation sites due to that one mutation which didn't make sense. I've used Modpred and GlycoMine to see if they predict the same thing.

Modpred made a much more reasonable o-linked glycosylation prediction, it only showed one change in o-linked glycosylation due to the single mutation. However, using Modpred to detect n-linked glycosylation shows that Modpred predicts 6 novel sites using the Myinga (1976) strain which we have experimental data saying those 6 sites don't exist. Modpred may be a little relaxed in it's cutoff.

GlycoMine correctly predicted the n-linked sites found experimentally in the Myinga strain. I am currently using it to look at the o-linked sites.

Current Problem: Where are you stuck? GlycoMine is really slow and there is no downloadable version.

Possible resolutions: What are you thinking about?

Date: 02/10/2016

Current Goal(s): What are you working on? Ebola Flexibility. AIM

Update: What have you done? Run several 100ns jobs for testing expanded ensemble using different combinations of lambda settings.

Ebola Flexibility: Spoke with Celeste. She wants to see how many times mutations gain or lose a T or S and associate these mutations with a correlation to disorder/flexibility. T and S are the amino acids associated with O-linked glycosylation sites.

If mutant loses a T/S, does it gain a T/S upstream or downstream? She wants to see this on the tree.
If mutant gains a T/S, how does flexibility change? Does it increase? This is where I would run simulations of 11mers.

If mutant loses a T/S, does flexibility go down at all or stay the same?
If mutant gains flexibility, does it gain a T/S?
If mutant loses flexibility, does it lose a T/S?

Current Problem: Where are you stuck?There seems to be a bug with the expanded ensemble. The “direct” simulations show expected results. The “expanded” simulations have very strange results.

Possible resolutions: What are you thinking about?

Date: 12/08/2015

Current Goal(s): What are you working on?

I spent the weekend working on AIM and starting the 11mer simulations.

Update: What have you done?

11mers: I have use python (biopython) to sort the sequences and have 44 11mers after filtering through the 96 sequences. These simulations are ready to be started ASAP since it will take a while to finish. On 42, I can run 2 sequences per node. The sequences are not large enough to take advantage of dual GPU’s and testing shows that 1 GPU with 20 procs is the best configuration for these systems. Each sequence will take 2 days. I can run 4 sequences every 2 days, so it should take about 23 days. I added a day for error.

AIM: I emailed Michael Shirts and asked for support and advice for completing AIM in Gromacs. In Gromacs, I found how they support replica exchange. I found this:

/* de[i][j] is the energy of the jth simulation in the ith Hamiltonian minus the energy of the jth simulation in the jth Hamiltonian. */

It contains the difference; de[i][j] = enerpart_lambda[i+1] - enerpart_lambda[0];

where enerpart_lambda[i] += foreign_term[F_EPOT];

In gromacs, the native lambda terms are given in the mdp options by “init-lambda-state”. The foreign lambda terms are the ones given by the mdp option “fep-state”.

In the code that I am working with, expanded.c, I have seen the same difference used for the Metropolis mover and other expanded ensemble “moves”. I’m convinced now the the term I was using is correct:

de = weighted_lamee[lamtrial] - weighted_lamee[fep_state];

where weighted_lamee[i] = sum_weights[i] - scaled_lamee[i];

where sum_weights[i] = init_lambda_weights;

and

where scaled_lamee[i] = (enerpart_lambda[i+1] - enerpart_lambda[0])/(mc_temp*BOLTZ);

sum_weights[i] is initialized by the mdp option “init-lambda-weights” which is “The initial weights (free energies) used for the expanded ensemble states. Default is a vector of zero weights. format is similar to the lambda vector settings in fep-lambdas, except the weights can be any floating point number. Units are kT. Its length must match the lambda vector lengths.”

sum_weights is then updated by the lmc-stats option; no, metropolis transition, barker transition, wang-landau or min-var.

Therefore, if we want to start the simulation with initial guesses for the free energies, we need the sum_weights term in the calculation.

The scaled_lamee[i] term then is the same as de[i][j] from replica exchange but scaled by 1/kT.

This changes the way my acceptance criteria works. Which is okay, I just wish I had seen it two months ago. I just need to scale df by the same 1/kT and then figure out what changes to make in the acceptance criteria.

Acceptance criteria: In the mdp options, the metropolis mover “Randomly chooses a new state up or down, then uses the Metropolis criteria to decide whether to accept or reject: Min{1,exp(-(beta_new u_new - beta_old u_old)}”. Beta = 1/kT

The criteria is similar to AIM, so I looked at the code in expanded.c. The Metropolis acceptance criteria shows a negative de but the code has exp(de), a positive de. This may mean the value for de is already negative. I think this is the case, given the way de is defined in the code. This changes my acceptance criteria.

trialprob = exp(de+df) should be correct.

Current Problem: Where are you stuck?

Compute-0-24 was reset on November 17 and no longer has CUDA and cmake installed.

Possible resolutions: What are you thinking about?

Date: 12/01/2015

Current Goal(s): What are you working on? I’ve been working with Celeste to pick out new mutations to run simulations on.

Update: What have you done? I’ve used Clustal to align the 96 sequences. We chose the reference sequence and I’ve been looking through the alignments to find the peptides at the residue locations. I was making mistakes writing so I used python to search the sequences and remove duplicates. I have 32.

Current Problem: Where are you stuck? I have 32 mutant peptides. I don’t think this is right. I think I have too many. I need to check the sequences by hand.

Possible resolutions: What are you thinking about? I’m not comfortable reading the tree data. I need to do sanity checks. If Caleb is around, I need to ask for his help reading the phylo tree.

Date: 11/17/15

Current Goal(s): What are you working on? AIM, O-linked glycosylation

Update: What have you done? Researching the prediction of O-linked glycosylation and using other software for the prediction. For AIM, I've gone and talked to a professor in CS to ask questions about the code. His insight was helpful and he said I could come back as needed.

Current Problem: Where are you stuck? I need to understand the F_DVDL term better.

Possible resolutions: What are you thinking about? nothing really. just chuggin along.

Date: 11/10/2015

Current Goal(s): What are you working on? AIM, Glycosylation sites. I need to understand O-linked glycosylation prediction better, so I'm trying to research that topic. Most literature relates either to HIV or influenza.

Update: What have you done? Research. For AIM, I thought I would be able to use the results from a previous simulation as the initial weights to test AIM. It turns out the the weights are the initial values for the free energies (G) that get updated by the lmc-stats option in expanded ensemble dynamics. I don't have these values for the previous simulation, I seem to only have delta G values from the xvg files. Because I was doing free energy simulations, not expanded ensemble simulations, the log file doesn't have the values for G, nor delta G. So, I thought about it and decided the simplest solution would be to run a solvation free energy simulation using expanded ensemble dynamics. I did this for ethane in water and methane in water, due to having the needed files available from two other online tutorials. The solutions are well referenced and seemed like the quickest and easiest solution. Now that I have these solutions I can run AIM with initial weights and see if this helps me track down the bug. Do you know what the original computer bug was?

https://en.wikipedia.org/wiki/Software_bug#Etymology

Current Problem: Where are you stuck? It doesn't make sense that AIM isn't working in Gromacs. But maybe it is? I'm running the ethanol simulation now, with the init-lambda-weights set.

Possible resolutions: What are you thinking about? I'm hoping to get some feedback from the devs at redmine.

Date: 11/03/2015

Current Goal(s): What are you working on? AIM, glycosylation site prediction, data presentation

Update: What have you done? I submitted a "feature" to redmine.gromacs.org in order to get some feedback from the devs for AIM. I also ran 96 sequences through the O and N glycosylation prediction server. I forwarded the results to those in need.

Current Problem: Where are you stuck? I need to find a better way to represent the data from the glycosylation predictions and I'm waiting for feedback from the devs at Gromacs.

Possible resolutions: What are you thinking about? If they called them “sad meals”, would kids still buy them? Also, *where* do you stick the feather and call it macaroni?

Date: 10/27/2015

Current Goal(s): What are you working on? Bioinformatics code, AIM and researching intrinsically disordered prediction, molecular evolution and thinking more about protein flexibility in relation to positive selection and glycosylation sites.

Update: What have you done? I wrote the disordered proteins predictor paragraph for Celeste’s paper. I need to add a sentence to explain the “why”. I’ve learned a lot about effects of mutation in ordered regions of proteins. I wrote some python programs for interacting with gene bank sequences.

Current Problem: Where are you stuck? The information is still cooking in my head. I don’t know what to do with it. My thought process keeps getting interrupted. I need to run simulations, but I’m not sure exactly what needs to be run and why.

Possible resolutions: What are you thinking about? I feel like bioinformatics is an incomplete field. I’m not finding the structure that I need. Maybe that’s just the nature of PhD work.

Date: 10/20/2015

Current Goal(s): What are you working on? AIM, disordered proteins write up

Update: What have you done? Went to a conference last week. Spent two nights in a resort. That was new. Spent some time on AIM.

Current Problem: Where are you stuck? I ‘ave no idea what to write for Celeste’s paper. AIM is partially working, thanks to help from Marty. Some output that I’m currently thinking about.

```
Print out status of current AIM step
The current potential energy is -34479.33594
de =    0.00000
(lamtrial - fep_state) =    0.00000
1.0/(nlim-1) = 1.0/(  21.00000 - 1) =    0.05000
dfavg[lamtrial] =  156.38287
dfavg[fep_state] =  156.38287
df =    0.00000
de - df =    0.00000
-beta*(de-df) =   -0.00000
...skipping...
dfavg[fep_state] =  105.76698
df =    5.28017
de - df =  839.30188
-beta*(de-df) = -336.48076
exp(-beta*(de-df)) =    0.00000
trialprob =    0.00000
```

r2 = 0.80230
lambda new was rejected
Lambda Old was 2
Lambda New is 2

Lambda	AIM Count	laccept	POT
0	15581	50.63535	-33496.39453
1	978	22.39264	-33421.69531
2	65094	0.30264	-34280.15625
3	256	78.90625	-33435.57422
4	227	93.39207	-33330.49219
5	220	95.00000	-33312.01953
6	208	99.03846	-33544.25391
7	138981	0.15038	-33500.04297
8	97604	0.22130	-33502.15234
9	219	94.97717	-33523.20312
10	217	92.16590	-33442.42188
11	2557	8.09542	-33412.73828
12	236	89.83051	-33419.52344
13	224	95.08929	-33439.62109
14	427913	0.04954	-33457.26562
15	1532	14.68668	-33460.88281
16	50292	0.44938	-33389.01172
17	28803	0.78464	-33400.79297
18	14007	1.67773	-33405.64062
19	132170	0.17704	-33409.75000
20	22681	50.78259	-33400.75391

Possible resolutions: What are you thinking about? AIM seems to be getting stuck at particular spots. What values of de is there a chance of acceptance?

Date: 09/29/2015

Current Goal(s): What are you working on? Poster for upcoming conferences. Job search. AIM (mostly thinking about). Need to work on figure legend and need to write my section on disorder prediction.

Update: What have you done? Finished the graphic for Celeste's paper. I have a rough poster thrown together. Parts of AIM works. Lambda moves as expected. The units don't seem right.

Current Problem: Where are you stuck? I don't know what else I should show on the poster. What "methods" should I put in for a poster? What is my "take home" message? Did I actually show that flexibility is related to positive selection? Isn't that inferred? I don't know if I'm performing AIM tasks in the right order or if order even matters. Does storing the average before accept/reject matter? Is there a way to test units?

ARGUMENT: Amino acid flexibility affects(?) positive selection.

POSIT: protein structure can predict site-specific evolutionary sequence variation (i actually don't know what this means… site specific evo seq var doesn't necessarily mean positive selection or negative selection. may not be useful here.) http://www.biorxiv.org/content/biorxiv/early/2014/07/21/004481.full.pdf
INFERENCE: evolutionary variation is responsible for positive selection
POSIT: mutations change the protein local dynamics and structure
POSIT: protein flexibility and protein function are strongly linked.
POSIT: rmsf is a measure of local protein dynamics
INFERENCE: rmsf is useful for predicting sequence variation

M: look at sasa avgs; does comparison with rmsf give any insight? is there anything interesting?
look at glycosolation sites. Where are these fragments located relative to gc sites? is there a relationship?
more beneficial if mutation is … distance means something. where are the glyco sites?

Possible resolutions: What are you thinking about? For AIM, is there a way to determine the range of values for df? How can I test if I'm accepting/rejecting correctly? I need a programming buddy.

Date 9/22/2015

Current Goal(s): What are you working on? Poster, AIM, Papers

Update: What have you done? Worked on poster

http://inbre.uidaho.edu/images/uploads/Poster%20information%20June%2024_20154.pdf

Poster presentation notes:
What's the story? What are you trying to share?

A picture is worth a thousand words. It is always better to show a well-designed picture than a list of bullet points. It is always better to show a plot than a table.

Let figures tell the story as much as possible.

Font:
Use Arial or similar font with large font sizes. It has been shown that Arial or similar fonts are much easier for people to read. In addition, a good rule of thumb is to use a minimum font size of 20. Clarity should be your goal so this is not the time to get creative with fonts.

Title: Calibri (headings) 96
Authors: Calibri (headings) 66
References: Calibri (headings) 40
Body: Calibri (body) 36
Figure captions: Calibri (body) 20

Take home message:
Give the audience a single take home message. A commonly used phrase from the book Dazzle 'em with Style is "tell 'em what you're gonna tell 'em, then tell 'em, then tell 'em what you've told 'em." In other words, you

should have one lesson that you want the audience to learn, then tell them about it multiple times in your presentation.

Title:
The effect of Ebola glycoprotein evolution on protein flexibility

Authors:
Christopher A. Mirabzadeh*, Aran Z. Burke, Caleb J. Quates, Celeste J. Brown, Craig R. Miller, Erin L. Johnson, Holly A. Wichman, Kyle P. Martin, Tanya A. Miura, F. Marty Ytreberg**
University of Idaho CMCI
*christopherm@uidaho.edu
**ytreberg@uidaho.edu

Abstract:
The data gathered during the recent Ebola epidemic is providing a wealth of information that can be used to understand Ebola evolution and how it could modify the efficacy of vaccines The goal of this study is to determine the effects of previous, ongoing, and future viral evolution on the structure and antibody binding properties of GP. We are focusing on the disordered mucin-like domain of the Ebola virus glycoprotein (GP) that is a target of vaccines and antibody-based therapeutics. We performed molecular dynamics simulations of small fragments of the mucin-like domain of GP to understand the biophysical implications of amino acid mutations. We will discuss how mutations change the protein local dynamics and structure in this disordered region of the protein.

Goal: What was your question(s)?
What does the RMSF imply about flexibility and positive selection?
What if we look upstream or downstream from these sites?
We are using the RMSF as a mean measure of the flexibility of each residue.

Methods: Computational? Experimental?
What did you do?
How did you do it?
Provide a full summary description of the methods.

Results:
What did you find?
RMSF C-alpha plots
Ancestor versus Mutant plots
Errorbars

Conclusion:
Did you answer your question(s)?
If you failed, why?
What could you do differently?
What did you learn?

Additional images:
CMCI logo

Talking Points:
My work is based on the non-structured region
Ebola
Viral Evolution
efficacy of vaccines

Questions to prepare for:
Disordered/Unstructured versus Ordered/Structured regions
How has Ebola evolved?
What is the mucin-like domain?
What is the RMSF?
What is c-alpha?
What have you done?
What is the CMCI group?
What is the focus of the CMCI group?
What is your part in the CMCI group?

Date: 09/15/2015

Current Goal(s): What are you working on? Presentation, Poster, AIM

Update: What have you done? Finished the graphic for Celeste.

Current Problem: Where are you stuck? Need to decide what constitutes as “AIM is working”

Possible resolutions: What are you thinking about? All the separate parts of AIM and how to test each.

Date: 09/08/2015

Goals as of Fall 2015

- Short term (semester):
    - Update these goals
        - Read through the previous years for ideas, copy-paste
    - Start applying for (ugh..) jobs.
        - Already started. Need to be more serious about it.
    - Start writing skeletons of papers to publish.
    - Bike as much as possible
        - may give up around October.
    - Use AIM to run a simulation
        - fail, debug, rinse, repeat
- Medium term (next year):
    - Graduate

        - If I don't graduate, I may take time off and work and write papers and come back to graduate later.
- Long term (5-10 years):
    - Have a job that is interesting but not too challenging such that I can still do programming and whatever else on the side.
    - Develop an app
    - Publish the stories that I started writing.
    - In 5 years convince Brahm to go to college
    - In 10 years, convince Bella to go to college

Current Goal(s): What are you working on? Goals, AIM, Running rmsf and other analysis on new Hamsters, staying warm, Poster for conference, Presentation for next week.

Update: What have you done? Ran one more set of simulations to take averages from. Worked on AIM.

Step	Time	Lambda
5885600	5885.60000	0.60000

MC-lambda information

N	FEPL	Count	G(in kT)	dG(in kT)	dfavg	
1	0.000	14480	0.00000	2.67412	102.78301	
2	0.050	25256	2.67412	2.72802	145.33255	
3	0.100	41410	5.40214	2.54247	139.28026	
4	0.150	63872	7.94461	2.33945	110.66930	
5	0.200	92670	10.284	2.16433	88.79339	
6	0.250	125502	12.44838	1.99390	72.58244	
7	0.300	162455	14.44228	1.83748	61.61971	
8	0.350	199886	16.27977	1.69830	53.81909	
9	0.400	235204	17.97807	1.56581	47.48801	
10	0.450	272272	19.54387	1.44872	42.72556	
11	0.500	305058	20.99259	1.34045	38.73650	
12	0.550	335975	22.33304	1.24218	35.45882	
13	0.600	363686	23.57522	1.15324	32.65841	<<
14	0.650	389098	24.72847	1.07430	30.30346	
15	0.700	410664	25.80277	1.00090	28.28588	
16	0.750	429696	26.80367	0.93643	26.53337	
17	0.800	446331	27.74010	0.88150	25.10836	
18	0.850	460961	28.62160	0.83832	24.01502	
19	0.900	476119	29.45992	0.80987	23.31944	
20	0.950	496935	30.26979	0.81005	23.42277	
21	1.000	538070	31.07985	0.00000	25.22414	

Current Problem: Where are you stuck? Not sure if AIM is working correctly. I think it is. But that could be because I want it to be.

Possible resolutions: What are you thinking about? Bulking up and being Zangeif for halloween. Not sure if I can pull of the red shorts in late October though.

Date: 09/01/2015

Current Goal(s): What are you working on? AIM. Doing one more set of simulations on 42 just to double check my averages due to some of the error bars overlapping. Overlapping error bars

Standard Error of the Mean (SEM) http://www.graphpad.com/support/faqid/1362/

“If two SEM error bars do overlap, and the sample sizes are equal or nearly equal, then you know that the P value is (much) greater than 0.05, so the difference is not statistically significant. The opposite rule does not apply. If two SEM error bars do not overlap, the P value could be less than 0.05, or it could be greater than 0.05. If the sample sizes are very different, this rule of thumb does not always work.”

Update: What have you done? I’ve gotten feedback from the devs at gmx_developers.

“To be more precise dhdl is only (fully) calculated at each step where free energies are calculated, i.e. every nstdhdl steps.
To find where the final values are stored, it's often easiest to look in mdebin.c, in this case at line 1244.” -Berk Hess

Current Problem: Where are you stuck? I’ve been trying to figure out how to use what I found. enerd->term[F_DVDL] seems to be the right term, but I haven’t really been able to print out the elements. I have to make changes to the code or write my own function to get at the data.

“In any case, it would be relatively easy to change the existing machinery
to force it do the full calculation dhdl calculation every step. If you
wanted to make it very easy, just set nstdhdl = 1, and introduce a new
variable that controls the printing of dhdl to the file alone.” -Michael Shirts

Possible resolutions: What are you thinking about? I need a c/c++ guru. Maybe someone in the CS department?

Date: 08/24/2015

Current Goal(s): What are you working on? Abstract, AIM, Graphics

Update: What have you done? Not a lot. The smoke has really been bad and hard to get things done with headaches and just feeling crappy. I’m trying to make error bars for my plots.

Current Problem: Where are you stuck? What data do I use for the error bars?

Possible resolutions: What are you thinking about? Moving to some place that isn’t of fire.

Date: 08/17/2015

Current Goal(s): What are you working on? Abstract, AIM, Hamster Graphics

Abstract

Title: The effect of Ebola glycoprotein evolution on protein flexibility

The data gathered during the recent Ebola epidemic is providing a wealth of information that can be used to understand Ebola evolution and how it could modify the efficacy of vaccines The goal of this study is to determine the effects of previous, ongoing, and future viral evolution on the structure and antibody binding properties of GP. We are focusing on the disordered mucin-like domain of the Ebola virus glycoprotein (GP) that is a target of vaccines and antibody-based therapeutics. We performed molecular dynamics simulations of small fragments of the mucin-like domain of GP to understand the biophysical implications of amino acid mutations. We will discuss how mutations change the protein local dynamics and structure in this disordered region of the protein.

Author List:
Christopher A. Mirabzadeh, Aran Z. Burke, Caleb J. Quates, Celeste J. Brown, Craig R. Miller, Erin L. Johnson, Holly A. Wichman, Kyle P. Martin, Tanya A. Miura, F. Marty Ytreberg

keywords: Glycoprotein, Ebola, protein flexibility

*look up how to not use “which”
needs a sentence on why is important to study ebola? why should Marty care? 1 or 2, not too much.

idea: imagine explaining this to mom or dad. where would you start? massive outbreak? people are dying. why is it so important to develop vac?

something more cumulative -- EBOV has killed this many people since 1976 with the largest outbreak being the recent 2014 outbreak. Has been happening over time, but more important now due to the size of the outbreak and affected populations.

Marty’s definition of an abstract taken from his website;

> The abstract should be a concise and easy to read summary of your article. Scientists will typically scan an abstract to decide whether they want to read the article in depth so it is important to avoid discouraging them with jargon or technical details. Your abstract should be one paragraph in length and should summarize why your work is worthy of publication, the methods you used, your results, and your conclusions. An abstract should be able to stand on its own so avoid using abbreviations or citations.

http://webpages.uidaho.edu/ytreberg/teaching/FMYtreberg_paper_guidelines.pdf

---

Date: 08/10/2015

Current Goal(s): What are you working on? I am working on AIM and collecting data from simulations.

Update: What have you done? I actually took some days off. I wasn't able to do much because of the schedule policy on 42.

AIM notes: Code in place doesn't work correctly. Seems to be the wrong dh/dl. Break down each step and make sure each step is doing what you expect it to be doing.
is it symmetric?
is it moving one step left or one step right? --confirmed that when it moves it moves by 1, up or down.
print out lots of stuff! histogram. get more information about what's going on.
most likely mistake is the acceptance probability.
difference in area between lambdas.

Current Problem: Where are you stuck? N/A

Possible resolutions: What are you thinking about? INBRE abstract

Abstract:
Add introductory sentence or two about GP and Ebola.

We are focusing on the Ebola virus glycoprotein (GP), which is the target of vaccines and antibody-based therapeutics. The goal of this study is to determine the effects of previous, ongoing, and future viral evolution on the structure and antibody binding properties of GP. To understand the biophysical implications of mutations in the Ebola glycoprotein mucin-like domain, we performed molecular dynamics simulations of small fragments of the protein. Using these simulations we determine how mutations change the protein local dynamics and structure in this disordered region of the protein.

Date: 08/02/2015

Last week at LCSC.

Current Goal(s): What are you working on? Still working on AIM and my hamsters. I need to collect data from the jobs that have run, but I'm waiting for the last two to finish so I don't over use resources. Any jobs that I submit on 42 are going to the same nodes and I'm accidentally over using the hardware.

Update: What have you done? I gave Celeste some graphics to look at. There are three on the root directory of the Ebola dropbox.

Current Problem: Where are you stuck? Teaching rockets to a bunch of kids. But I'm also having fun.

Possible resolutions: What are you thinking about? I want to start submitting resumes. Amazon is hiring.

Date: 07/26/2015

Current Goal(s): What are you working on? AIM. Hamsters are done. Creating graphics. I think the jobs on coan are dead. I need to check them again. I need to average all of the simulation runs and come up with

graphics to represent the data. I also need to make graphics for Celeste's data. I want something by Tuesday for the group meeting.

Update: What have you done? Finally figured out Celeste's data. Now I need to create the graph. I'm looking at SciDavis to make the job quick and easy.

Current Problem: Where are you stuck? The graphic may be pretty busy. I need to have something presentable for Tuesday, or want to have something presentable. I've played with the data, but some of it looks like it's missing. I need to ask Celeste about this.

Possible resolutions: What are you thinking about? Using GPU's and machine learning to predict disordered proteins. ASICS would be faster.

Date: 07/21/2015

Current Goal(s): What are you working on? GPU jobs. AIM. Figures for Celeste.

Update: What have you done? All GPU jobs are running over 500 ns/day. I have run all 8 11mers using the GPU's on fortytwo in one week. I will run another set of jobs next week and then assess how much longer the jobs running on the default nodes will take. I then need to decide whether or not I've run enough jobs or need to run more.

AIM might be working, as seen below.

Step	Time	Lambda
2000000	2000.00000	1.00000

Writing checkpoint, step 2000000 at Fri Jul 17 16:20:55 2015

MC-lambda information

N	FEPL	Count	G(in kT)	dG(in kT)
1	0.000	1	0.00000	22.84482
2	0.050	1	22.84482	20.72015
3	0.100	2	43.56497	16.68775
4	0.150	5	60.25272	8.77760
5	0.200	1	69.03032	6.93835
6	0.250	15	75.96867	0.82997
7	0.300	560	76.79865	0.55733
8	0.350	2401	77.35598	0.48017
9	0.400	4541	77.83615	0.47411
10	0.450	7581	78.31026	0.45037
11	0.500	12510	78.76063	0.42822
12	0.550	19840	79.18885	0.40965
13	0.600	30416	79.59850	0.39268
14	0.650	46143	79.99117	0.37867
15	0.700	68128	80.36984	0.36532
16	0.750	99482	80.73516	0.35385
17	0.800	144768	81.08901	0.34767
18	0.850	208633	81.43668	0.34401

```
19  0.900   299362   81.78069    0.34576
20  0.950   429937   82.12645    0.35351
21  1.000   625673   82.47996    0.00000 <<
```

It seems to be spending too much time in the lower lambdas. I need to test more to make sure I'm using the correct term. I found an array in gromacs, sum_dg, that is defined to be the free energies of the states. I'm using it to update the free energy estimates. I also need to figure out how to get my version of mdrun to run using mpirun.

Current Problem: Where are you stuck? Not stuck, just thinking really hard. It's that constipated look that we all get when we're close to the solution.

Possible resolutions: What are you thinking about? I need to start writing or at least outlining.

Date: 07/13/2015

Current Goal(s): What are you working on? Still running jobs. AIM. GPU testing.

Update: What have you done? Managed to get 507 ns/day on the GPUs but I can't seem to use more than two GPUs per node. There are 4 available.

Current Problem: Where are you stuck? Can't seem to use more than two GPUs per node. The performance dies off considerably. Down to 1 ns/day.

Possible resolutions: What are you thinking about? I think that the systems are too small to take advantage of multiple GPUs

---

Date: 07/06/2015

Not available for meetings this week. I'm at LCSC playing with water bottle and sling shot rockets! I am available via email and text and will attempt to reply in a timely manner. I will be back on Friday, probably exhausted.

Current Goal(s): What are you working on? AIM and running reliability checks on my hamsters. I have 8 gpu's…. !!! I need to find out if the latest version of gromacs supports multiple gpu's, I don't think so. I need to find out how to efficiently spread the work-load across multiple cards and processors. I'm having fun!

Update: What have you done? AIM is almost working! AIM is moving in lambda space as can be seen from the log output below. I've had the code in place for a while but the dv/dl term stayed at zero. I've been reviewing the math/code and everything looks right. I spent all day Thursday tweaking mdp options and then remembering that the topology has to have two lambda states. I felt really dumb. But it was a really good day spent to confirm that I am on the right track. I want AIM working by the end of Summer. I'm being conservative. I really want it working by the end of next week.

```
 Wang-Landau incrementor is:         1.2
 N   FEPL    Count   G(in kT)  dG(in kT)
 1  0.000        2    0.00000    1.20000
```

```
 2  0.100       1    1.20000   -4.80000
 3  0.200       5   -3.60000    3.60000
 4  0.300       2    0.00000    1.20000
 5  0.400       1    1.20000   -1.20000
 6  0.500       2    0.00000   -1.20000
 7  0.600       3   -1.20000    1.20000
 8  0.700       2    0.00000    1.20000
 9  0.800       1    1.20000    0.00000
10  0.900       1    1.20000 -411.60114
11  1.000     344 -410.40112    0.00000 <<
```

Current Problem: Where are you stuck? AIM gets stuck at lambda=1.0. I think I'm using the wrong energy term. Gromacs has several derivative energy terms. F_DVDL, F_DVDL_VDW… I went with F_DVDL first because that's what I think is being used in the gmx_bar program. The incrementer may be too large as well. I'm using the wang-landau method to update the ensemble weights, but it may make more sense to use metropolis-transition. I'm still debating and trying to understand the difference between them.

Gromacs doesn't detect whether or not a gpu is busy. If you have multiple gpu's in a node, you have to use the -gpu_id flag. And PBS doesn't currently differentiate between the two available nodes. If I specify the gpu_id, I currently don't know a way to specify which node I want to run through the PBS script because both gpu's nodes are tied to the queue "gpu". Is that true? Is there a way to specify which node to run on through PBS? I know I can just ssh to the node and run jobs, but that defeats the purpose of PBS.

I did manage to spread the work load across all 8 gpu's but the performance was horrible. One server was getting 1 ns per day. I need to figure out why. I think it has something to do with the "teams" in parallel processing. I may have to play with the pinoffset or figure out how to use two gpu's per simulation. Still thinking about this one.

Possible resolutions: What are you thinking about? The code in gmx_bar has been very helpful to look at and I need to spend more time there.

I think I found a solution:

#PBS -l host=<hostname>	Run your job on a specific host

Date: 06/29/2015

Current Goal(s): What are you working on? I am re-running all hamsters for reliability testing.

Update: What have you done? I have 16 jobs running on forty-two and 18 running on coan. I'm getting back into AIM while I wait.

Current Problem: Where are you stuck? Not stuck, just waiting.

Possible resolutions: What are you thinking about? ...Apparently not a whole lot.

Date: 06/22/2015

Current Goal(s): What are you working on? Graphics. Reading journal articles. Learning gmx sasa (not to be confused with salsa, it's tastier cousin). SASA computes the solvent accessible surface area which is the surface area of the biomolecule that is accessible to solvent (water). Accessible surface area can be used to improve prediction of protein secondary structure.

Update: What have you done? This week I created a ton of graphics trying to determine the difference between the available options for gmx rmsf, gmx do_dssp, and gmx xpm2eps. Each of these programs have separate dependencies; i.e. downloading external databases, or parameter files. I've been following this website;

http://ringo.ams.sunysb.edu/index.php/MD_Simulation:_Protein_in_Water_(Pt._2)

It's good as a starting place for analysis and what the different analysis programs can show you. It's not a complete source but there is good information there.

Current Problem: Where are you stuck? I want the image at the bottom of that page but my total time steps makes the image too large. I need to use the -dt flag with do_dssp to reduce the image size. But doing this loses information. using -dt of 100 fits the letter but I lose helix information.

Possible resolutions: What are you thinking about? I'm just going to generate the larger dataset with all time so we can see what's going on and then we can decide.

Date: 06/15/2015

Current Goal(s): What are you working on? I am currently re-running two simulations to validate the previous simulations. I chose hamsters 3 and 4. I chose those because of all the problems I had with 4 during the first simulation run and 3 is the ancestor/mutant combination. I am also looking at different way to plot the analysis data. Craig would like to see mutation per residue along the x-axis. The hope is to see the RMSF as a function of residue and make a better association to flexibility.

Update: What have you done? I started the rerun simulations late last week and the GPU job finished on Sunday. I'm waiting for the highmem node to complete, then I can start comparing. I finished the online course/training as required for the grant. I've generated some graphs based on c-alpha and I'm looking them over.

Current Problem: Where are you stuck? I'm not exactly sure how to make the plots. I wouldn't say that I'm stuck, just exploring.

Possible resolutions: What are you thinking about? Graduating… Besides that, Marty had a "crazy idea" last week that we should look upstream and downstream of these residue mutation sites and determine the proximity to ligand binding sites (i think that was the gist). It could tell us if the flexibility at this mutation site could be correlated with positive selection. I found precedent for this idea in a paper "Relation between

flexibility and positively selected HIV-1 protease mutants against inhibitors", Braz. I need to read the paper much closer, but wanted to bring it up.

Date: 06/08/2015

Current Goal(s): What are you working on? Plots/Graphs are all finished and ready to discuss

Update: What have you done? Sent out emails with graphs to discuss.

Current Problem: Where are you stuck? Definitions need to be researched.

Possible resolutions: What are you thinking about? Where did the time go?

Date: 05/31/2015

Current Goal(s): What are you working on? The basic idea is predicting peptide flexibility and equating this to positive selection. I've run all of the simulations and I am still putting together the analysis. Not happy that it's taking this long. I was getting incorrect data from the log file on peptides 3 and 4. I found out that the logs were written slightly differently, thus, there isn't as much consistency across log files as I had hoped. Running an automated script to extract the data has to be localized for each peptide.

Update: What have you done? I have concatenated the log file for peptide 4 back together. Since it was restarted multiple times, the log file was in separate chunks. I've also run the analysis tools and have plots to look at.

Current Problem: Where are you stuck? Just looking at the data right now.

Possible resolutions: What are you thinking about? What does it all mean? I have plots of the data, so now I'm just trying to decipher it which means reading more papers.

Date: 05/26/2015

Current Goal(s): What are you working on? All simulations have finished. I am now looking at the xvg files and doing analysis.

Update: What have you done? Typical analysis protocol from Marty and web

1. Run "gmx mindist -pi" to make sure the molecule never "saw" its periodic image. The minimum distance output by the command should be less than the smallest cutoff (r cutoff) you used in your mdp file. This is the most important thing to check because if the shortest distance is less than your cutoff then the simulation cannot be trusted.
2. Look at the potential energy (gmx energy, option 11) as a function of time. Look for rough convergence. ← Not as helpful with temperature lambdas since, unless you can create separate edr files for each lambda, you have temperature fluctuations throughout the simulation.
3. Process the trajectory (gmx trjconv) file to only keep the components you want (remove water and ion information) and recenter and remap the xtc and .tpr files. Not necessary but could save disc space.

4. Use RMSD (gmx rms) to make sure the simulation has run long enough ← Probably not as important for these runs. For a protein simulation this is good to make sure you didn't get unfolding.
5. Use RMSF (gmx rmsf) to look at local fluctuations. Choose the group appropriate to the fluctuation you find most interesting. Ada's paper looked at RMSF of the backbone atoms.
6. Use do_dssp (gmx do_dssp) to view secondary structure analysis as a function of time. This requires a local install of DSSP which is outlined on the Gromacs website.
7. Looking at the RMSF visually (VMD, PyMOL) could be useful, but need to be able to visualize the difference between the mutant and reference peptides.

Result of mindist is good.
The shortest periodic distance is 1.30308 (nm) at time 588537 (ps), ./2.SKGTDLLDPAT/xtcFiles/analysis.log
The shortest periodic distance is 1.18402 (nm) at time 77327 (ps), ./5.STSPQPPTTKT/xtcFiles/analysis.log
The shortest periodic distance is 1.21801 (nm) at time 672649 (ps), ./7.TAAGPLKAENT/xtcFiles/analysis.log
The shortest periodic distance is 1.23531 (nm) at time 225424 (ps), ./6.STSPQSPTTKT/xtcFiles/analysis.log
The shortest periodic distance is 1.26076 (nm) at time 69097 (ps), ./8.TAAGPPKAENT/xtcFiles/analysis.log
The shortest periodic distance is 1.3034 (nm) at time 404805 (ps), ./1.SKGTDFLDPAT/xtcFiles/analysis.log
The shortest periodic distance is 1.37933 (nm) at time 256339 (ps), ./4.SKSADSLDLAT/xtcFiles/analysis.log
The shortest periodic distance is 1.39843 (nm) at time 165051 (ps), ./3.SKSADFLDLAT/xtcFiles/analysis.log

Current Problem: Where are you stuck? Deciphering what I'm looking at. I have data, now I need to figure out what it all means.

Possible resolutions: What are you thinking about? I'm reviewing Ada's paper to see what she did to get some clues.

Date: 05/18/2015

Current Goal(s): What are you working on? 6 of the 8 simulations (hamsters!) finished yesterday so I'm looking to make sure they're healthy.

Update: What have you done? I've added AIM code to gromacs and tested that I can run simulations with my specific mdp option. It was actually more difficult getting gromacs to recognize the mdp options then getting the AIM code in the right place.

Current Problem: Where are you stuck? Not stuck, just busy. I need to analyze the simulations (hamsters) and run some simulations against my code. Extracting the data from the peptide simulations (hamsters) needs to be done ASAP.

Possible resolutions: What are you thinking about? My hamsters were born!

Date: 05/11/2015

Current Goal(s): What are you working on? The simulations are running for a week now. I'm looking at the gromacs code and I'm working on how to analyze the peptides once the simulations are finished. What exactly do we want to know? What is important? Is it important to know that the simulation performed as expected? We want to say something about the peptides. Do we just look at the RMSF? Is "flexibility" the only property we're interested in? With the code, is everything we need already in place and all we really need is to set the proper mdp options?

Update: What have you done? I've looked at the log files, .edr files and .xtc files to try and determine what I can look at (what information is available) using a combination of g_energy and g_trjconv. These are analysis tools within gromacs. I've figured out some important things. For example, if we want all frames associated with a particular lambda point, there is no native way to do this in gromacs. What you can do is view the log file and look for the table of temperatures and weights;

Step	Time	Lambda
200	1.00000	0.00000

MC-lambda information

Wang-Landau incrementor is: 2

N	Temp.(K)	Count	G(in kT)	dG(in kT)
1	310.000	1	0.00000	369.00000 <<
2	312.448	0	369.00000	388.00000
3	315.154	0	757.00000	424.00000
4	318.144	0	1181.00000	462.00000
5	321.449	0	1643.00000	488.00000
6	325.102	0	2131.00000	531.00000
7	329.138	0	2662.00000	570.00000
8	333.599	0	3232.00000	620.00000
9	338.529	0	3852.00000	634.00000
10	343.978	0	4486.00000	702.00000
11	350.000	0	5188.00000	0.00000

If you look, you'll notice the "<<". This tells you the current lambda state, i.e. the current temperature for the simulated tempering run. So this tells me that at step 200 or time 1.0000 ps, lambda=1. I can search through the log and collect this data and determine what lambda state is being visited at what time/step. Then I can use "trjconv - dump" to extract the frame associated with that timestep and do so for each instance of the particular lambda state.

While doing the simulated tempering simulations, we noticed that there are actually several pieces of the AIM method already available in gromacs.

Current Problem: Where are you stuck? Now that I know how to extract the frames, to what purpose will we be using them? Also, I know parts of the wang-landau method in gromacs will work for AIM, but which parts and how can I take advantage of it?

According to
http://www.alchemistry.org/wiki/GROMACS_4.6_example:_Ethanol_solvation_with_expanded_ensemble

“Expanded ensemble calculations build up the free energies as the simulations are performed. It does this by building up simulation weights as the simulation progresses, so each different thermodynamic state (lambda state) has a different weight. If it didn't have this weight, then the simulation would spend all the time in the lowest free energy states. When it visits each state equally, **then the weights will be exactly equal to the free energies**. “

Possible resolutions: What are you thinking about? How do I analyze these structures? Without bias (bias=knowledge of desired outcome)? How do I define selection? Also, I really need to know what are the similarities and differences between AIM and Wang-Landau. Can I use wang-landau weights with an aim move? If the weights of the wang-landau method are the free energies and it also calculates dG and counts the number of times visited by each lambda, isn’t that everything I need for AIM? Would I just need a move method based on these weights? Can this be done with some clever combination of mdp options? --I actually don’t think so, but it’s worth thinking about.

Date: 05/02/2015

Current Goal(s): What are you working on? I’ve been working on how to choose the number of temperatures and the temperature range for simulated tempering.

Update: What have you done? A lot of simulations and reading of research papers. I decided to use 10 temperatures and a temp range of 310 to 350. These were guesses based on the literature. Each temp is a lambda. Each lambda has a probability weight factor that isn’t a priori known and is not easy to determine initially. Saying that it’s non-trivial and tedious is a nice way of putting it. The weights are important because they tune the simulations. It’s all about efficiency. In order to automate finding the weights I tried running simulations to find out how long it took to visit each lambda at least once. I worked up to 5 ns and was only half way through the lambdas. I then decided to guess at initial weights based on an equation that I found in the literature. The way I worked the equation was wrong so I decided I would need to adjust the weights as I went. It turned out to be a good place to start.

Starting with that initial guess and then separate 1ns runs, I manually updated the next 1ns run with the weights (in the G column) found from the previous run. The result shown took about 9 runs;

N	Temp.(K)	Count	G(in kT)	dG(in kT)
1	310.000	13	0.00000	362.35999
2	312.448	15	362.35999	387.60028
3	315.154	20	749.96027	425.03949
4	318.144	23	1174.99976	459.91992
5	321.449	17	1634.91968	496.47998
6	325.102	18	2131.39966	531.15991
7	329.138	25	2662.55957	571.60010
8	333.599	15	3234.15967	609.68018
9	338.529	21	3843.83984	652.24072
10	343.978	24	4496.08057	690.63867
11	350.000	32	5186.71924	0.00000

Obviously I can’t count because I wanted 10 temps and got 11. Zero based indexes are funny like that.

The higher counts at the higher temps are good because more frequent transitions at the higher temps result in improved sampling at the lower temps.

Allowing the simulation to run for 10ns, with no initial guess

N	Temp.(K)	Count	G(in kT)	dG(in kT)	
1	310.000	1995	0.00000	364.00000	
2	312.448	1813	364.00000	394.00000	
3	315.154	1616	758.00000	428.00000	<<
4	318.144	1402	1186.00000	460.00000	
5	321.449	1172	1646.00000	504.00000	
6	325.102	920	2150.00000	530.00000	
7	329.138	655	2680.00000	574.00000	
8	333.599	368	3254.00000	618.00000	
9	338.529	59	3872.00000	118.00000	
10	343.978	0	3990.00000	0.00000	
11	350.000	0	3990.00000	0.00000	

The simulation would need to run longer to get the last two weights but you can see that the weights in the G column are very close to the ones above. I can only assume that the counts would be greater on the higher temps as the simulation was allowed to run longer.

Current Problem: Where are you stuck? I basically guessed at how many temperatures to use and the range of temperatures. The literature states that efficiency gains are independent of the number of temps as long as the temp changes fast enough. What I assume is fast enough is small deltaT. I also guessed on the temp range (310-350). The temp only needs to be high enough to allow the system to escape from states of energy local minima, i.e. energy barriers. I'm not sure if I can justify my choices.

Possible resolutions: What are you thinking about? I'm going to just pretend that it was obvious that these were the correct temperature range and number of temperatures based on my other-world knowledge. You know, like Jackson's textbook. It's simple to see that since my choices arrived at the solution, they were obviously correct.

Date: 04/28/2015

Current Goal(s): What are you working on? Trying to understand and analyze a simulated tempering simulation.

Update: What have you done? I've researched the topic and tried to find settings specific to Gromacs. I've gone through the different mdp parameter settings trying to figure out how to make it work properly.

Current Problem: Where are you stuck? Simulated tempering is an expanded ensemble algorithm that explores energy space by attempting to make moves to different temperatures. I have a list of temperatures for the simulation to cycle through but even after 2 ns the simulation is stuck on lambda 0, the first temperature. Or at least that's what I think is going on. When I look in the log file, it shows a count for the temperatures. Only the first temperature has an increasing count. Maybe this is just a reflection of the init-lambda setting? I do have an energy distribution, and that looks right. I'm just not able to make a histogram of the temperature

distribution. For all I know the simulation is running and just needs to run for more than 2 ns, I don't know how to tell.

Possible resolutions: What are you thinking about? I hate it, but if I'm going to get this to work I need to spend more time monkeying around with it. I'm also thinking about getting into the expanded ensemble code and see if I can see what the code is doing and maybe figure out how to tell if it's running correctly. Maybe a few strategically placed print statements. I could try chanting. Know any good spells?

Date: 04/20/2015

Current Goal(s): What are you working on? Performance testing using vsites and the 11mer's. I'm trying to resolve an error that I get using vsites and neutral caps. Whenever I use the charmm22star forcefield, I have to choose a specific combination of N- and C- terminal caps. I can either use the N- as NH3+ and C- as COO- or N- as None and C- as CT2. If I use any other N- terminal then I get an error "In molecule type "Protein" Virtual site 3fad construction involves atom 5, which is a virtual site of equal or high complexity. This is not supported." The problem is when the vsites are constructed, there are two hydrogen atoms in virtual_sites3 heading in the top file. I've tried removing this line but that just creates even more problem. Using the amber99sb-star-ildn force field fails when I use genion using ACE and NME terminal residues. It works fine without the terminals.

Update: What have you done? I've managed to dramatically increase the performance using vsites:

GPU:	Performance 352 ns/day	
CPU's on GPU node:	Performancd 211 ns/day	Restricted to using 1 node and 40 ppn
default:	Performance 79 ns/day	Restricted to using 1 node and 8 ppn
highmem:	Performance 180 ns/day	Restricted to using 1 node and 24 ppn

I understand how to use the -ntomp option now. If a node, like the gpu node, has 40 cores, then you can call qsub with 1 node and 8 cpu's and then use the mdrun option -ntomp 5 to use 5 OpenMPI threads per process. 8*5=40. I was curious if I could do the same thing with the default nodes using 4 cpu's and 2 OpenMPI threads, if that would speed it up even more. I didn't see a change but I was able to spread the work load over 2 nodes at 60 ns/day.

8 cpu's seems to be the best way to break up the systems since they are so small. Then more OpenMPI threads per process gives even more of a speedup.

All jobs on fortytwo failed because I had an error in my job file.

Current Problem: Where are you stuck? I really want to use the charmm22star force field since it has been shown to be optimal in describing conformational changes of small peptides. I also want to have neutral terminals since I don't want any non-physical torsions in the peptides. I also want to run with vsites to get the best performance. Is not having N- or C- termini physical?

Possible resolutions: What are you thinking about? If I want the ideal simulation and best performance, I'm thinking of running without capping. What I've researched says that the problem is with "fibrillation". I'm not

sure if this applies to our peptides. See "**The Importance of Being Capped: Terminal Capping of an Amyloidogenic Peptide Affects Fibrillation Propensity and Fibril Morphology"** The abstract:

The formation of aggregated fibrillar β-sheet structures has been proposed to be a generic feature of proteins. Aggregation propensity is highly sequence dependent, and often only part of the protein is incorporated into the fibril core. Therefore, shorter peptide fragments corresponding to the fibril core are attractive fibrillation models. The use of peptide models introduces new termini into the fibrils, yet little attention has been paid to the role these termini may play in fibrillation. Here, we report that terminal modifications of a 10-residue peptide fragment of human islet amyloid polypeptide strongly affect fibrillation kinetics and the resulting fibril morphology. Capping of the N-terminus abolishes fibrillation, while C-terminal capping results in fibrils with a twisted morphology. Peptides with either both termini free or both termini capped form flat fibrils. Molecular dynamics simulations reveal that the N-terminal acetyl cap folds up and interacts with the peptide's hydrophobic side chains, while the uncapped N-terminus in the C-terminally capped version results in twisting of the fibrils due to charge repulsion from the free N-termini. Our results highlight the role of terminal interactions in fibrillation of small peptides and provide molecular insight into the consequences of C-terminal modifications frequently found in peptide hormones *in vivo*.

Date: 04/13/2015

Current Goal(s): What are you working on? Performance tuning gromacs for the 8 "11mer" systems from Celeste.

Update: What have you done? I've managed to increase Gromacs performance from 7.5 ns/day to 31 ns/day on the Coan. On the highmem node, which has 24 processors, I've managed to get 66 ns/day and on the GPU, 131 ns/day.

Current Problem: Where are you stuck? Manually tuning gromacs is grueling. I'm not stuck, just worn out. I now know how to parallelize a simulation across 64 cores. I complained about not being able to do that a few weeks ago. On Coan, each node has 8 processors, so that's 8 nodes times 8 processors. More cores isn't necessarily a good thing. The communication between nodes slows the system down to 11 ns/day, using 64 cores. 32 gets 21 ns/day and using 16 cores, 2 nodes by 8 processors, I get the 31 ns/day. These scales have variation. It's not spot on 31 ns/day. There is variation. Sometimes I get 25 ns/day with the same configuration. The best configuration is more cores per node so as to not spread communication across too many servers. The fortytwo cluster only allows 4 to 6 cores per node if I don't want to wait for resources. The best performance on fortytwo was 21 ns/day. I might be able to get more if I spend more time manually tuning pp/pme ranks.

Possible resolutions: What are you thinking about? A nap would be nice.

Date: 04/06/2015

Current Goal(s): What are you working on? 4ns simulations are running. I'm learning more about simulated tempering for the simulations requested by Celeste in the Ebola group. Hacking gromacs.

Update: What have you done? Before running the 4ns simulations I checked to see if I could get them to run any faster on the GPU. Approx. 35ns per day is the fastest I can get for the system. It's paradoxical. If the system were bigger it would run faster. For the simulations from Celeste, I have capped the 11mers with neutral C and N termini using pmx. I've also spent some time learning simulated tempering for the very same simulations. For hacking gromacs, I have emailed the gmx-developers group asking for advice on the derivative of the hamiltonian. I need to narrow down the variable.

Current Problem: Where are you stuck? I need to learn how to properly tune gromacs for simulated tempering. I have many possible variables for the derivative of the hamiltonian and I need to narrow it down to the correct one. For all I know, everything I've found points to the same memory location.

Possible resolutions: What are you thinking about? I need to run many tests to figure out simulated tempering and how to best run it so I don't waste cpu cycles. I need to run the simulations for Celeste for 10's or 100's of ns so it's important that I get it right.

Date: 03/30/2015

Current Goal(s): What are you working on? I'm running 4 ns simulations to calculate the free energy difference of the test system. I'm still hacking gromacs.

Update: What have you done? I ran simulations for 2 ns for the bound and unbound test system. The free energy difference for 2 ns was -1.15 kcal/mol. The difference for 1 ns was -2.74 kcal/mol.

	unbound	bound
1ns	393.11	381.62
2ns	388.03	383.21

Because of the difference (which isn't as bad as I thought) I just need to run the simulations longer to see if the system converges any closer to the actual value. I've also made some progress in writing code. Gromacs is compiled on my laptop so I can make changes there without having to connect to Coan through the VPN all the time.

Current Problem: Where are you stuck? There are several functions in the Expanded Ensemble code that I've been tracing to determine what they do. There is one that I have no clue what it's doing but I don't think it's necessary. I need to choose the functions that I can use and functions that I need to write based on my python code. I've found everything except for the derivative of the free energy.

Possible resolutions: What are you thinking about? I'm thinking that I'm just going to start typing and worry about whether or not I'm duplicating code later, once I've got something working.

Date: 03/23/2015

Current Goal(s): What are you working on? Are we doing updates this week? I spent most of the last 1.5 weeks trying to figure out how to optimally parallelize the free energy calculation for the test protein 2WPT, for mutation DA33L. I wasted a better portion of the previous week (before break) and I've been trying to make up

for it. I'm still hacking at Gromacs. I've also started using Marty's gmx_run script for Gromacs simulations. Very handy.

Update: What have you done? 1) I have found exactly where to implement my AIM code in Gromacs, the file is expanded.c and I am taking advantage of the expanded ensemble using lmc-mc-move. All I need to do is add an a lmc-move enum of elmcmoveAIM and add an acceptance criteria to the MC code that is already in place. I need to go back over my old code to see what else I need. 2) I went through all sorts of test scenarios to optimize the bound (46000 atoms) and unbound (15000 atoms) protein simulation runs. Using any more than 1 node with 8 ppn's gets a domain decomp error. This error is a result of Gromacs trying to partition the system into equally sized chunks to spread over the processors. I've tried manually setting the number of pme ranks with -npme, I've changed fourierspacing (decreased from 0.12 to 0.10) and pme_ranks (increased from 4 to 6/8) in the mdp file to get the PME mesh to be around 0.33 as suggested in Lemkul's tutorial. I found papers that suggested settings for best performance (number of MPI threads versus OpenMP threads) that seemed to suggest being able to run 40000 atoms in parallel, but I can't get it to work. I tried mpirun -np to manually set the number of processors, I even tried running on highmem with 24 procs and was still denied with the domain decomp error. I ended up just running everything on the GPU.

Current Problem: Where are you stuck? I realized that I haven't been able to run an actual parallel simulation. Kinda depressing. I can't get more than 3 ns/day using 8 procs. For 20, 1ns simulations, that's almost 7 days and I'm supposed to be increasing the time from 1 ns to 2 ns.

Possible resolutions: What are you thinking about? I compiled my own version of gromacs, but it's in my home directory which isn't available across the cluster. Could we have a developer version of Gromacs in the share directory available to all nodes in the cluster? This would also benefit my hacking of Gromacs. The problem with this is making sure I don't change the currently working Gromacs package when I recompile. We don't want to be constantly changing a running package across the nodes. I think this can be achieved by creating a developer group and a dev folder in /share/apps with dev group read/write access and adding me to the developer group so I don't have to use sudo to write to the directory.

Date: 03/09/2015
Current Goal(s): What are you working on? I'm currently running simulations of 2WPT protein to obtain the free energy of mutation DA33L. While that is running I'm hacking Gromacs.

Update: What have you done? I have started two simulation runs consisting of 21 jobs each. I think it will be done in 80 hours from 11 a.m. today. Sometime Thursday around 7 p.m. unless it crashes.

Current Problem: Where are you stuck? Hacking other peoples code is never fun. Fortunately, Gromacs is well documented and we have the jenkins documentation, http://jenkins.gromacs.org/job/Documentation_Gerrit_master/javadoc/index.html#.

Possible resolutions: What are you thinking about? There's an outdated tutorial on patching mdrun, http://www.gromacs.org/Developer_Zone/Programming_Guide/Patching_mdrun. I've been using it but I really wish it would get updated.

Date: 03/02/2015

Current Goal(s): What are you working on? I'm currently contemplating the catastrophic effects of simulating ebola on our server. I think it would be world ending. The internet is infinitely interconnected by "the cloud" which makes the degree of separation between nodes like 1. Can you imagine the digital evolution of a virus with that much computational power available to it? It's mind boggling. Besides that, I'm trying to get the vacuum simulation to work for my tripeptide. Kinda boring, but Ebola, right?

Update: What have you done? Why are you giving me the third degree? I've worked really hard all week. I was able to run the forward and reverse simulations using pmx and gromacs on the tripeptide and validated the procedure. So, yeah…. I got nothing. Which was expected when you take the difference between the free energy runs of the two systems, zero, nothing. Now I'm trying (heavy emphasis) to run the same simulations in vacuum.

Current Problem: Where are you stuck? At home mostly. I'm a shut in. When I do get out, I tend to find dark corners and ignore people. Oh, I'm stuck on the vacuum simulation. It keeps violently blowing up. The atoms are shooting off into the great beyond, like thousands of nanometers away from the origin. I'm using the same mdp settings and topologies that I used in the water simulations, minus the water.

Possible resolutions: What are you thinking about? Latin words ending with -cide. Like, did you know the word uxorcide means to murder ones wife. And sorocide, think sorority, yep, means murder ones sister. Just the words, not the actual act. I have no idea why my system (not sister) is blowing up, so I'm googling "blowing up gromacs". Do you think I'll get tagged by some government acronym? I don't think it's the topology or structure file, the water sims ran perfectly. Mayhaps it's something to do with the mdp settings? Position restraints? I even started from scratch with the original pdb files and placed the polypeptide in a box, took it home and showed the kids. No solvent, obviously. The explosion happens on minimization, pretty quick. The particles come flying out of the server at high speeds and put holes in the wall. It's fun to watch, I'm sure. I honestly have no idea. I'm not allowed in there since "the incident". The error is in LINCS. Should I use shake? I just need rattle and roll and I've got a band. Should I change the temperature settings? 300k in vacuum, is that physical? Should be like 3k. I doubt the system blowing up has anything to do with highly energetic particles, maybe… ? Tutorials don't help. Nothing is mentioned about changing the temperature or pressure settings. They say use the same mdp file. About as useless as the letter 'h' in 'yeah', or red lights in Grand Theft Auto.

Date: 02/22/2015

Current Goal(s): What are you working on? I am rerunning simulations. I'm still trying to get the free energy from mutation for a small peptide using pmx. pmx is very picky with input files.

Update: What have you done? I used pmx to mutate an alanine dipeptide to valine and ran forward and reverse simulations using gromacs. I computed the free energy difference. I was expecting zero, but I got 0.38 kcal/mol. I either need to run the simulation longer or start with a better structure. I think the structure is the problem so I spent some time learning how to create my own peptides. I used Avogadro to make my own alanine and valine peptides. Avogadro makes creating peptide structures very easy. I'm now running the mutations A2V and V2A. I should have the end product by Monday afternoon.

Current Problem: Where are you stuck? I was stuck with residue terminals and residue structure. Gromacs creates termini based on the current residues in the structure file. My original structure was only two alanines

so gromacs was adding hydrogens in order to create terminals. This in turn makes pmx fail because there are more hydrogens than expected. pmx expects a very specific list of atoms for each residue. The alanine-alanine structure had terminal residues but I couldn't find a valine dipeptide. I decided to create my own peptides. The reason I wanted to create my own peptide is because I felt I was hacking something together and didn't really understand all of the failures I was getting.

Possible resolutions: What are you thinking about? I created peptides using Avogadro, 4 alanines, Ala-Ala-Ala-Ala and another one with Ala-Val-Val-Ala. I'm using the end alanines as end terminals so I can mutate the middle two residues. Thus, the first polypeptide AAAA will mutate to AVVA and the second polypeptide AVVA will mutate to AAAA. This fixed all the errors I was getting from gromacs and the errors from pmx about unexpected structure. I'm hoping this will validate pmx. I plan to calculate the difference in free energy of the two systems and hopefully get zero or something really close to zero. My next step is to run the same simulations in vac and compute the solvation free energy.

Date: 02/15/2015

Current Goal(s): What are you working on? Same goals. I'm creating the output files for the free energy run that I finally got to work using alanine dipeptide.I'm doing the simulation in vacuum now. I would like to start digging into the gromacs source code again and figure out how to incorporate AIM.

Update: What have you done? I've created automation scripts to make my life easier and followed along on two tutorials that match the gist of what I'm trying to accomplish. I've run the free energy simulation in water. I now need (according to one of the tutorials, but I'm not sure this is right) to run the sim in vacuum. It's currently running. Since there are no water molecules, I assume it should take half the time. The previous sim, solvated in tip3p, took 2 hours per lambda run, so about 40 hours total.

Current Problem: Where are you stuck? Currently I'm not stuck on anything. I'm waiting for the simulation to finish. This is a good thing. I worked through several problems and feel pretty confident about my ability to run simulations now, at least in Gromacs. I'm sure Marty will take me down a notch in the near future. There are some things I'm concerned about. I don't know how well I've minimized and equilibrated my system. It's a small system, so I'm hoping the number of steps was sufficient. I know how to check so I may take a few hours to take a look.

Possible resolutions: What are you thinking about? I'm worried how long a simulation for a large protein will take. I really don't like the unknown time of completion. Time scales for simulations are dependent on the number of particles. Do we know a way to figure out how long a simulation of N atoms should take? I can't recall, but I don't think it's a linear scale.

Date: 02/09/2015

Current Goal(s): What are you working on? Goal 1 is still the same. I'm currently attempting to use the pmx script on a small peptide.

Update: What have you done? I found a paper on the collagen-like peptide, PDB 1CGD, a small-ish peptide with the relative free energy of mutating the alanines to glycines. pmx easily created the topologies, but the MD simulation was taking too long so I looked for a simpler peptide, dipeptide, amide…. 3mer… The simplest models for polypeptides are the glycine and alanine dipeptides. I found that mutating alanine dipeptide to valine dipeptide is a very common and widely studied mutation. The mutation is chemically similar to converting methane to propane.

Current Problem: Where are you stuck? I have found coordinate files for alanine dipeptide with two alanines online from various tutorials and peptide data banks. Not all of them are the same. Some of them are generated for the Charmm27 forcefield which isn't supported by pmx. I have found others that work with pmx forcefields but after creating the mutation with pmx I have to run Gromacs pdb2gmx to convert the pdb to gromacs readable structure. Gromacs gives me an error about a dangling bond at nonspecific terminal end. I can get pmx to work, but Gromacs fails or I can get Gromacs to work and pmx fails by manipulating atoms in the structure file.

Possible resolutions: What are you thinking about? I think I need to build my own alanine dipeptide properly terminated, but I don't know how. The problem with pmx is that Gromacs keeps adding extra hydrogens even when I tell it to ignore hydrogens. This makes pmx fail because the order of atoms it expects for the mutation from alanine to valine is interrupted by extra hydrogens. I'm tempted to go back to the 1CGD collagen-like polypeptide simply because the pdb file is complete and neither pmx nor Gromacs complain about the structure. I can attempt to change all of the mdp settings in such a way as to run the most minimal simulation possible but it could still take too long. Too long is more than an hour per lambda run, considering I aim to do 21 lambda runs. A lambda run consists of minimization, equilibration, and production md as lambda goes to lambda + delta_lambda.

UPDATE: This just in… I have a properly terminated structure file now but when I run pdb2gmx I'm getting another error, "atom N not found in building block 1ACE while combining tdb and rtp". But, from what I can see, there is no N atom in the ACE residue in the input.pdb file which means Gromacs is trying to add it? ---Confirmed. the rtp is trying to add beginning and end terminus atoms. I ran

gmx-5.0 pdb2gmx -f input.pdb -o output.pdb -ff oplsaamut -ter -ignh -water tip4p

then choose none for start terminus and end terminus. Does this create problems? I've seen tutorials where this is done, but is it correct?
After all this, pmx failed to create the hybrid topology. So close!

Date: 02/02/2015

Current Goal(s):

My goal 1 hasn't changed. Goal 2 is resolved. I've gotten pmx to work but I'm still trying to calculate the binding affinity for the mutation.

Update:

pmx works as a way to generate topologies so we're going with this tool instead of Mobley's code. It's picky, but it works. There are caveats when using it. All input and output file extensions have to be .pdb. Gromacs doesn't care what file extension you use, so this isn't a problem.

Current Problem:

Keeping up with this update is harder than I thought it would be. Currently, I'm having a problem completing an entire simulation. The simulation for one lambda step (lambda is method of tracking the alchemical process of growing or decoupling a molecule as lambda goes from 0 to 1.) I get through the energy minimization and equilibration steps, but there is a LINCS error where some of the atoms have too much energy and the system is still blowing up.

Possible resolutions:

I just have to keep trying different sets of mdp settings and topology changes and possibly minimizing the system for a longer period of time. The topology created by pmx may be the problem, but I'm not sure at the moment.

Date: 01/25/2015

Current Goals:

1. Calculate the binding affinity change from PDB 2WPT, chains A and B. Mutant is DA33L. This means change D to L at position 33 in chain A.
    a. Expectation -3.4 kcal/mol
2. Test and try to understand the script created by Mobley for generating mutated topologies .

Update:

I have gone through several Free energy tutorials on the websites;
http://www.bevanlab.biochem.vt.edu/Pages/Personal/justin/gmx-tutorials/
http://www.alchemistry.org/wiki/Main_Page

I have tailored the tutorials to calculate the free energy of a larger molecule. Primarily, I have replicated the tutorial found here,
http://www.bevanlab.biochem.vt.edu/Pages/Personal/justin/gmx-tutorials/free_energy/index.html

I have created my own script files to automate the process involved. I did have one problem where I was getting a warning from GROMACS about nonbonded interactions. Researching the problem I found that the system was "blowing up". Forces were becoming too large between time steps. I went through all steps found in the troubleshooting guide found here;
http://www.gromacs.org/Documentation/Terminology/Blowing_Up

I've learned a lot about troubleshooting GROMACS but didn't find the answer easily. I finally sifted through each parameter setting and found that I could fix my problem by setting couple-intramol=yes. The simulation is a de-coupling simulation where we are slowly removing a protein or residue or ligand from a solvated system. If you tell mdrun (the part that actually computes the forces between atoms in Gromacs) that your calculation should not couple intramolecular interactions (couple-intramol=no) then all non-bonded interactions will be calculated at full strength. But, part of our system is being removed as a function of lambda. By setting couple-intramol=yes, we are scaling the non-bonded interactions as a function of lambda instead of at full strength. This is beneficial for large systems (such as the one I'm working with).

Current Problem:

Currently I am reviewing code used to generate topologies in GROMACS based on changing some part of the structure, i.e. a residue. We have code written by David Mobley and his team. The code doesn't work for proteins. It uses an external library, OEChem. Marty has also found a paper, recently published, that uses PMX to mutate a given protein for free energy calculations. Which is exactly what we need. We need to figure out which one to use.

Possible resolutions:

If I use Mobley's code, I will have to fix it. I currently don't know enough about either system to make the better informed decision. We could use both and compare, but why? What would that accomplish? The simplest thing would be to use the one that works the best with the most generalized functionality.