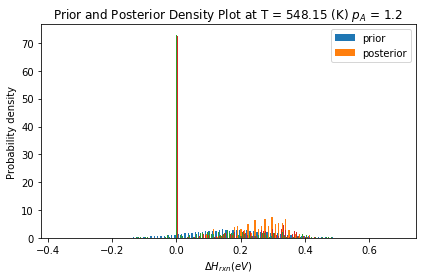
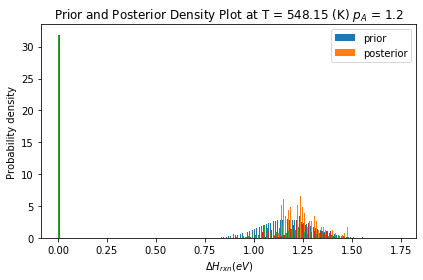
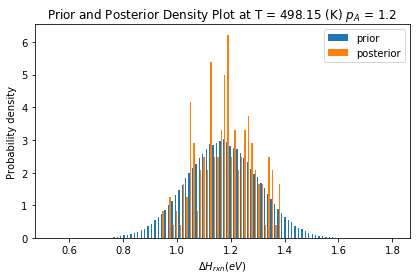
The good news is that I think things are working. You just need to do 100

The bad news is that I have two problems, but I think they are related. Read both 1) and 2) before trying to interpret.

1. I am seeing some bizarre high probability density spikes in the prior and the posterior around the prior and the posterior for DelaH\_Rxn = 0:
   1. 
   2. Originally I thought this spike might be due to numerical errors (either in the distribution sampling or in the simulations). However, I had no reason to believe that for this range of numbers, and also it was puzzling to see this in the posterior.
      1. The problem \***did not**\* go away with my uncertainty scaling code in runfile\_Example13\_CO\_H2O\_four\_parameters\_replacement\_doe\_pos\_1std\_offset\_1e2\_scaling
      2. The problem \***did not**\* go away if I shifted deltaH by a unit of 1 in the sampling and then shift back during simulation: runfile\_Example13\_CO\_H2O\_four\_parameters\_replacement\_doe\_pos\_1std\_offset\_deltaH\_plus1
         1. This is particularly suspicious. There is no obvious reason for this. I think there is some kind of indexing problem. See point #2 below.



* 1. I have also found that the spike is never in the last posterior graph. So if I change the temperature range, then it still disappears in the last graph. So although it looks like the highest temperature is “free of the problem”, that is not because of temperature.
     1. 

1. there is some kind of serious issue in the info gain being calculated for KL. You may want to just use UserInput.parameter\_estimation\_settings['mcmc\_info\_gain\_returned'] = 'log\_ratio' and then calculate the KL\_divergence info\_gain in the runfile.
   1. For any of the three below files, try making burn\_in 100 and mcmc\_length of 200. You will get an array broadcasting error. It is NOT because the sampling is too small. This is happening right now any time you try to make burn\_in more than half of the mcmc\_length.
      1. runfile\_Example13\_CO\_H2O\_four\_parameters\_replacement\_doe\_pos\_1std\_offset
      2. runfile\_Example13\_CO\_H2O\_four\_parameters\_replacement\_doe\_pos\_1std\_offset\_deltaH\_plus1.py
   2. To me, it looks like there is an indexing or slicing error due to us shifting to using more than one parameter in this model. I think the current “built in” KL code cannot handle that correctly. So I think that (for now) you need to use UserInput.parameter\_estimation\_settings['mcmc\_info\_gain\_returned'] = 'log\_ratio' and calculate KL\_divergence in the runfile.