MrBayes

Motivation:
http://blog.ted.com/how-a-ted-fellow-is-working-to-save-african-cassava/

Bayesian Inference:
https://www.youtube.com/watch?v=5NMxiOGL39M

Github for Dataset:
https://github.com/anders-savill/mrbayes-scc17

Thinking:
Creating evolutionary trees, maximizing (Model | measure)
MonteCarlo: randomness
Markov change: the next point depends on the prior point (think the tree itself)

MCMC:
https://www.youtube.com/watch?v=OTO1DygELpY

Simple tutorial:
http://evomics.org/learning/phylogenetics/mrbayes/
http://mrbayes.sourceforge.net/wiki/index.php/Tutorial_3.2

Terminology/Key resource:
http://treethinkers.org/tutorials/mrbayes/#Manipulating_Markov_chain_Monte_Carlo_MCMC_Settings

Manual:
http://mrbayes.sourceforge.net/wiki/index.php/Tutorial_3.2

Steps:

1. Read the Nexus data file
2. Set the evolutionary model
3. Run the analysis
4. Summarize the samples

Specification comes in the setting of the model
The setting of the params for execution, not really in the model itself

Need to get a sense of the time it will take to run

http://mrbayes.sourceforge.net/manual.php

- Uses MPI

- Both memory and compute heavy
- Relies on the BEAGLE library (large performance increase with NVIDIA cards, some improvement without them too, can work without downloading this library though)
    - BEAGLE is a high-performance library that can perform the core calculations at the heart of most Bayesian and Maximum Likelihood phylogenetics package. http://beast.bio.ed.ac.uk/beagle
    - Doesn't support OpenCL
- Interpreter style interface

Background:

- Performs "Bayesian inference and model choice across a large space of phylogenetic and evolutionary models"
    - https://en.wikipedia.org/wiki/Bayesian_inference_in_phylogeny
- Basically taking in dna data and trying out different possible trees then outputting likelihoods
- Understanding of math and the model you are simulating to set parameters
    - Bayesian inference
        - https://www.youtube.com/watch?v=5NMxiOGL39M
    - Evolutionary trees
    - Monte Carlo Simulations
        - https://en.wikipedia.org/wiki/Monte_Carlo_method
    - Markov chains

Compilation

- Download: http://mrbayes.sourceforge.net/download.php
- cd src
- autoconf
- ./config --with-beagle=no
- Make

Following example in the manual:

Red oval indicate expected time left

```
Using a relative burnin of 25.0 % for diagnostics

Chain results (20000 generations requested):

    0 -- [-8861.421] (-8177.817) (-8952.779) (-8922.991) * [-8443.670] (-8651.715) (-8597.967) (-8804.174)
  100 -- [-6447.096] (-6450.089) (-6765.497) (-6773.727) * (-6681.994) [-6711.407] (-7582.269) (-6741.800) -- 0:00:00
  200 -- [-6275.114] (-6250.518) (-6488.537) (-6292.688) * [-6307.441] (-6394.447) (-6643.454) (-6453.652) -- 0:01:39
  300 -- (-6212.726) (-6154.617) (-6391.441) [-6144.459] * [-6160.011] (-6242.674) (-6435.432) (-6231.794) -- 0:01:05
  400 -- (-6195.237) (-6143.921) (-6297.437) [-6074.333] * [-6082.597] (-6185.517) (-6244.331) (-6142.121) -- 0:00:49
  500 -- (-6171.945) (-6110.667) (-6220.365) [-6012.803] * [-6025.190] (-6168.985) (-6148.062) (-6102.624) -- 0:00:39
  600 -- (-6173.623) (-6085.185) (-6142.191) [-5966.113] * [-6005.380] (-6162.727) (-6137.176) (-6091.285) -- 0:00:32
  700 -- (-6119.239) (-6076.910) (-6111.988) [-5966.080] * [-5996.654] (-6137.062) (-6071.430) (-6081.389) -- 0:00:27
  800 -- (-6074.865) (-6036.418) (-6096.266) [-5959.447] * [-5996.865] (-6089.393) (-6048.726) (-6036.568) -- 0:00:24
  900 -- (-6016.660) (-6026.443) (-6086.090) [-5949.576] * [-5995.199] (-6070.883) (-6016.161) (-6026.863) -- 0:00:42
 1000 -- (-5997.858) (-6018.176) (-6063.201) [-5941.688] * (-5981.683) (-6036.847) (-6013.853) [-6004.118] -- 0:00:38

Average standard deviation of split frequencies: 0.015713

 1100 -- (-5995.335) (-5997.876) (-6061.016) [-5934.508] * (-5968.485) (-6029.164) (-5983.701) [-5908.384] -- 0:00:34
 1200 -- (-5973.123) (-5973.726) (-6053.331) [-5901.829] * (-5969.423) (-6012.188) (-5974.375) [-5913.180] -- 0:00:31
 1300 -- (-5931.530) (-5965.211) (-6053.696) [-5897.422] * (-5955.139) (-6003.347) (-5962.832) [-5883.090] -- 0:00:28
 1400 -- (-5924.119) (-5920.937) (-6015.868) [-5885.988] * (-5926.103) (-5997.075) (-5936.724) [-5875.514] -- 0:00:26
 1500 -- (-5920.566) (-5880.207) (-5997.132) [-5857.303] * (-5929.008) (-5994.674) (-5923.139) [-5838.484] -- 0:00:24
```

```
Average standard deviation of split frequencies: 0.000520

Continue with analysis? (yes/no): no

Analysis completed in 26 seconds
Analysis used 25.41 seconds of CPU time
Likelihood of best state for "cold" chain of run 1 was -5714.68
Likelihood of best state for "cold" chain of run 2 was -5714.68

Acceptance rates for the moves in the "cold" chain of run 1:
   With prob.   (last 100)   chain accepted proposals by move
      36.2 %     ( 33 %)     Dirichlet(Revmat)
      60.1 %     ( 62 %)     Slider(Revmat)
      17.8 %     ( 20 %)     Dirichlet(Pi)
      21.5 %     ( 23 %)     Slider(Pi)
      31.4 %     ( 29 %)     Multiplier(Alpha)
      67.5 %     ( 58 %)     Slider(Pinvar)
       0.4 %     (  0 %)     ExtSPR(Tau,V)
       0.1 %     (  0 %)     ExtTBR(Tau,V)
       0.2 %     (  0 %)     NNI(Tau,V)
       0.7 %     (  0 %)     ParsSPR(Tau,V)
      36.7 %     ( 28 %)     Multiplier(V)
      24.4 %     ( 26 %)     Nodeslider(V)
      12.9 %     (  7 %)     TLMultiplier(V)

Acceptance rates for the moves in the "cold" chain of run 2:
   With prob.   (last 100)   chain accepted proposals by move
      29.9 %     ( 26 %)     Dirichlet(Revmat)
      64.0 %     ( 59 %)     Slider(Revmat)
      13.1 %     ( 16 %)     Dirichlet(Pi)
      30.1 %     ( 20 %)     Slider(Pi)
      37.1 %     ( 40 %)     Multiplier(Alpha)
      70.7 %     ( 70 %)     Slider(Pinvar)
       0.1 %     (  0 %)     ExtSPR(Tau,V)
       0.1 %     (  0 %)     ExtTBR(Tau,V)
       0.3 %     (  0 %)     NNI(Tau,V)
```

```
       14.7 %     ( 23 %)     TLMultiplier(V)

  Chain swap information for run 1:

            1     2     3     4
      -------------------------
    1 |        0.69  0.47  0.31
    2 |  3296        0.69  0.43
    3 |  3402  3293        0.61
    4 |  3353  3362  3294

  Chain swap information for run 2:

            1     2     3     4
      -------------------------
    1 |        0.63  0.44  0.30
    2 |  3353        0.71  0.45
    3 |  3247  3381        0.63
    4 |  3316  3340  3363

  Upper diagonal: Proportion of successful state exchanges between chains
  Lower diagonal: Number of attempted state exchanges between chains

  Chain information:

    ID -- Heat
   -----------
    1 -- 1.00  (cold chain)
    2 -- 0.91
    3 -- 0.83
    4 -- 0.77

  Heat = 1 / (1 + T * (ID - 1))
     (where T = 0.10 is the temperature and ID is the chain number)


MrBayes >
```

```
MrBayes > sump

   Summarizing parameters in files ../examples/primates.nex.run1.p and ../examples/primates.nex.run2.p
   Writing summary statistics to file ../examples/primates.nex.pstat
   Using relative burnin ('relburnin=yes'), discarding the first 25 % of samples

   Below are rough plots of the generation (x-axis) versus the log
   probability of observing the data (y-axis). You can use these
   graphs to determine what the burn in for your analysis should be.
   When the log probability starts to plateau you may be at station-
   arity. Sample trees and parameters after the log probability
   plateaus. Of course, this is not a guarantee that you are at sta-
   tionarity. Also examine the convergence diagnostics provided by
   the 'sump' and 'sumt' commands for all the parameters in your
   model. Remember that the burn in is the number of samples to dis-
   card. There are a total of ngen / samplefreq samples taken during
   a MCMC analysis.

   Overlay plot for both runs:
   (1 = Run number 1; 2 = Run number 2; * = Both runs)

   +-------------------------------------------------------------+ -5718.99
   |                    1     22        11                  1 1 |
   |      1           2  2  1  1          1    1  1       1   1  |
   |               1       *   2 1  2     11   *2 1 1     1     |
   |      2         1       2 1     1       1     2  12   21  2 |
   |          2   12    *1*  1     *           1       1        |
   |     2        2   1          21  2 22    2     1 2 *22       |
   |           1**                   1*   222        1    22  2|
   | 1     ***   *               1                2 2           |
   |    1           22 2        2             2              2 1|
   |     1                                                      |
   |   22                                    1      2           |
   | 2*                            2                            |
   |1  1                                  2                     |
   |2                                                           |
   |                 1                                          |
   +------+-----+-----+-----+-----+-----+-----+-----+-----+-----+ -5733.99
```

```
   (Use the harmonic mean for Bayes factor comparisons of models)

   (Values are saved to the file ../examples/primates.nex.lstat)

Run   Arithmetic mean   Harmonic mean
------------------------------------
  1      -5719.66         -5732.23
  2      -5720.43         -5737.70
------------------------------------
TOTAL    -5719.97         -5737.01
------------------------------------


Model parameter summaries over the runs sampled in files
   "../examples/primates.nex.run1.p" and "../examples/primates.nex.run2.p":
   Summaries are based on a total of 302 samples from 2 runs.
   Each run produced 201 samples of which 151 samples were included.
   Parameter summaries saved to file "../examples/primates.nex.pstat".

                                  95% HPD Interval
                                 -------------------
Parameter      Mean      Variance     Lower       Upper       Median    min ESS*  avg ESS    PSRF+
-------------------------------------------------------------------------------------------------
TL          3.105825    0.090132    2.580004    3.653156    3.066408     14.52     43.71    0.997
r(A<->C)    0.043870    0.000077    0.027572    0.062378    0.043066     27.37     46.73    1.002
r(A<->G)    0.468491    0.002998    0.356836    0.570345    0.464441      8.82     12.13    1.006
r(A<->T)    0.037959    0.000058    0.020306    0.050366    0.037903     18.74     32.52    1.004
r(C<->G)    0.029920    0.000143    0.008782    0.051068    0.029976     21.32     21.79    1.008
r(C<->T)    0.401917    0.002200    0.308521    0.490502    0.402339     10.91     17.47    1.006
r(G<->T)    0.017842    0.000220    0.000138    0.046866    0.014266     10.55     28.50    1.014
pi(A)       0.354114    0.000182    0.331414    0.378681    0.355472     51.24    101.12    1.027
pi(C)       0.320615    0.000141    0.297910    0.340336    0.320324     82.17     89.34    0.998
pi(G)       0.082629    0.000046    0.071524    0.094418    0.082455     36.00     40.54    0.997
pi(T)       0.242643    0.000114    0.221015    0.265373    0.241646     34.23     38.13    1.036
alpha       0.636562    0.033750    0.355321    0.999574    0.613434      6.08      8.27    1.032
pinvar      0.162247    0.006458    0.000437    0.287807    0.173655      6.77      7.90    1.042
-------------------------------------------------------------------------------------------------
* Convergence diagnostic (ESS = Estimated Sample Size); min and avg values
```