Efficient Bayesian Evolutionary Analysis using Hamiltonian Monte Carlo

Arman Bilge, Timothy Vaughan, and Alexei J. Drummond

email: abil933@aucklanduni.ac.nz

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

ALLAN WILSON CENTRE

Motivation

- ▷ Bayesian analysis is a powerful tool for testing evolutionary hypotheses with sequence data
- ▷ We want to sample model parameters $\boldsymbol{\theta}$ according to probability given the data $P(\boldsymbol{\theta} \mid \mathbf{D})$
- ▷ Sampling usually done with **Markov chain Monte Carlo** random walk algorithm
- ▷ Random walks move through space slowly and make for **inefficient samplers**
- ▷ **Hamiltonian Monte Carlo** algorithm applies Newton's laws of motion to methodically traverse the space and **avoid random walk behavior**
- ▷ HMC uses the space's slope, an **expensive but theoretically rewarding** calculation

Hamiltonian Dynamics

Imagine a frictionless skatepark where the elevation at any point $\mathbf{x}$ is given by

$$y = -\log \pi(\mathbf{x}),$$

with $\pi(\mathbf{x}) \equiv P(\boldsymbol{\theta} \mid \mathbf{D})$, the probability of the model parameters given the data.
Note that parameter values with **high probability** will map to points with **low elevation**.
Consider a skater of mass $\mathbf{M}$ in this park with position $\mathbf{q}$ and momentum $\mathbf{p}$.
Then their total energy, called the **Hamiltonian**, is

$$\mathcal{H}(\mathbf{q}, \mathbf{p}) = U(\mathbf{q}) + K(\mathbf{p}),$$

the sum of their potential energy

$$U(\mathbf{q}) = -\log \pi(\mathbf{q})$$

and their kinetic energy

$$K(\mathbf{p}) = \frac{1}{2}\mathbf{p}^T\mathbf{M}\mathbf{p}.$$

The skater moves through the park according to Hamilton's equations,

$$\frac{d\mathbf{q}}{dt} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}} \text{ and } \frac{d\mathbf{p}}{dt} = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}},$$

which are integrated to find the skater's position and momentum at a particular time.

The HMC Algorithm

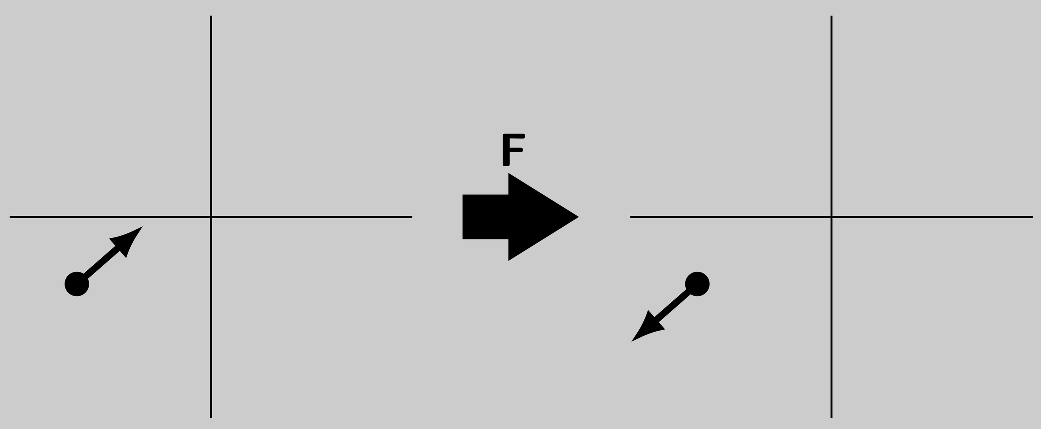
We control the skater with three operators, described below.
We can combine these operations to express a single iteration of the HMC algorithm.

function IterateHMC($\{\mathbf{q}, \mathbf{p}\}$)
  $\{\mathbf{q}', \mathbf{p}'\} \leftarrow \mathbf{FLR}\{\mathbf{q}, \mathbf{p}\}$ ▷ Make proposal and flip momentum
  $a \leftarrow \min(1, \exp(\mathcal{H}(\mathbf{q}, \mathbf{p}) - \mathcal{H}(\mathbf{q}', \mathbf{p}')))$ ▷ Calculate acceptance probability
  $\{\mathbf{q}, \mathbf{p}\} \leftarrow \begin{cases} \{\mathbf{q}', \mathbf{p}'\} & \text{with probability } a \\ \{\mathbf{q}, \mathbf{p}\} & \text{with probability } 1 - a \end{cases}$ ▷ Accept or reject proposal
  $\{\mathbf{q}, \mathbf{p}\} \leftarrow \mathbf{F}\{\mathbf{q}, \mathbf{p}\}$ ▷ Flip momentum back
  return $\{\mathbf{q}, \mathbf{p}\}$ ▷ Return next sample
end function

Flip Operator

Flips the skater's momentum.

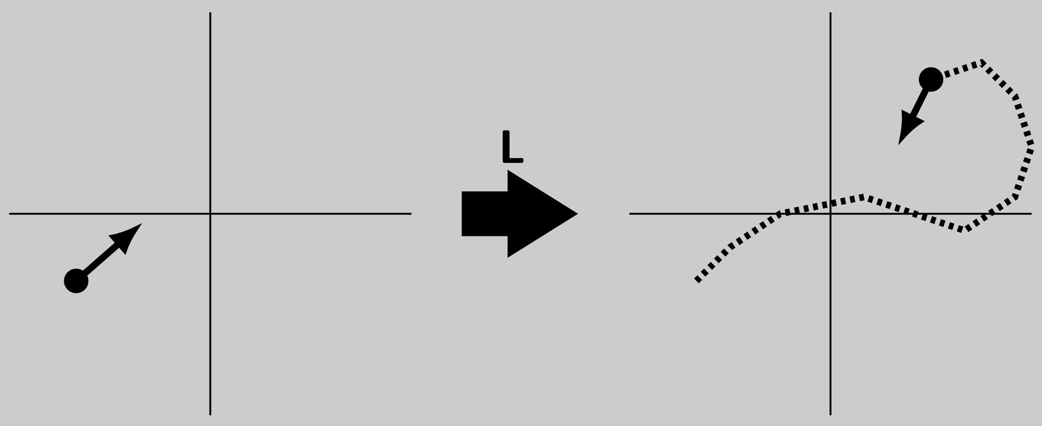
$$\mathbf{F}\{\mathbf{q}, \mathbf{p}\} = \{\mathbf{q}, -\mathbf{p}\}$$



Leapfrog Operator

Simulates the movement of the skater for some time s.

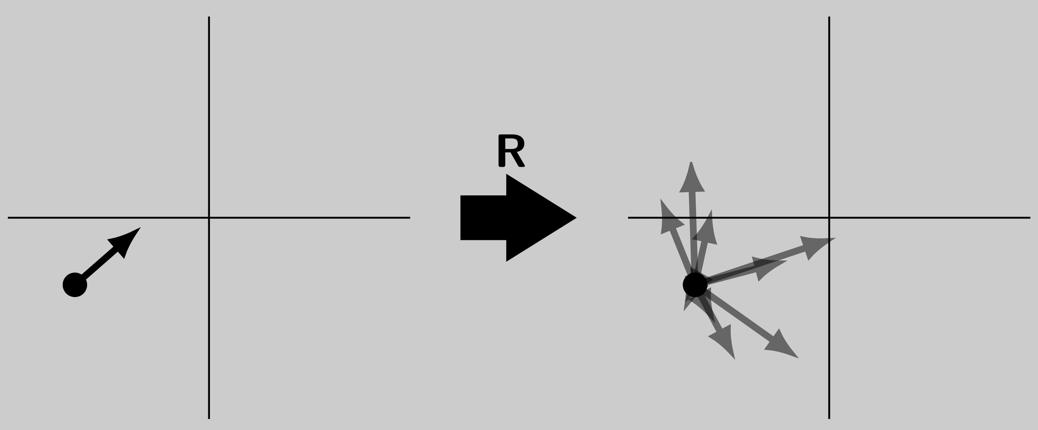
$$\mathbf{L}\{\mathbf{q}^t, \mathbf{p}^t\} = \{\mathbf{q}^{t+s}, \mathbf{p}^{t+s}\}$$



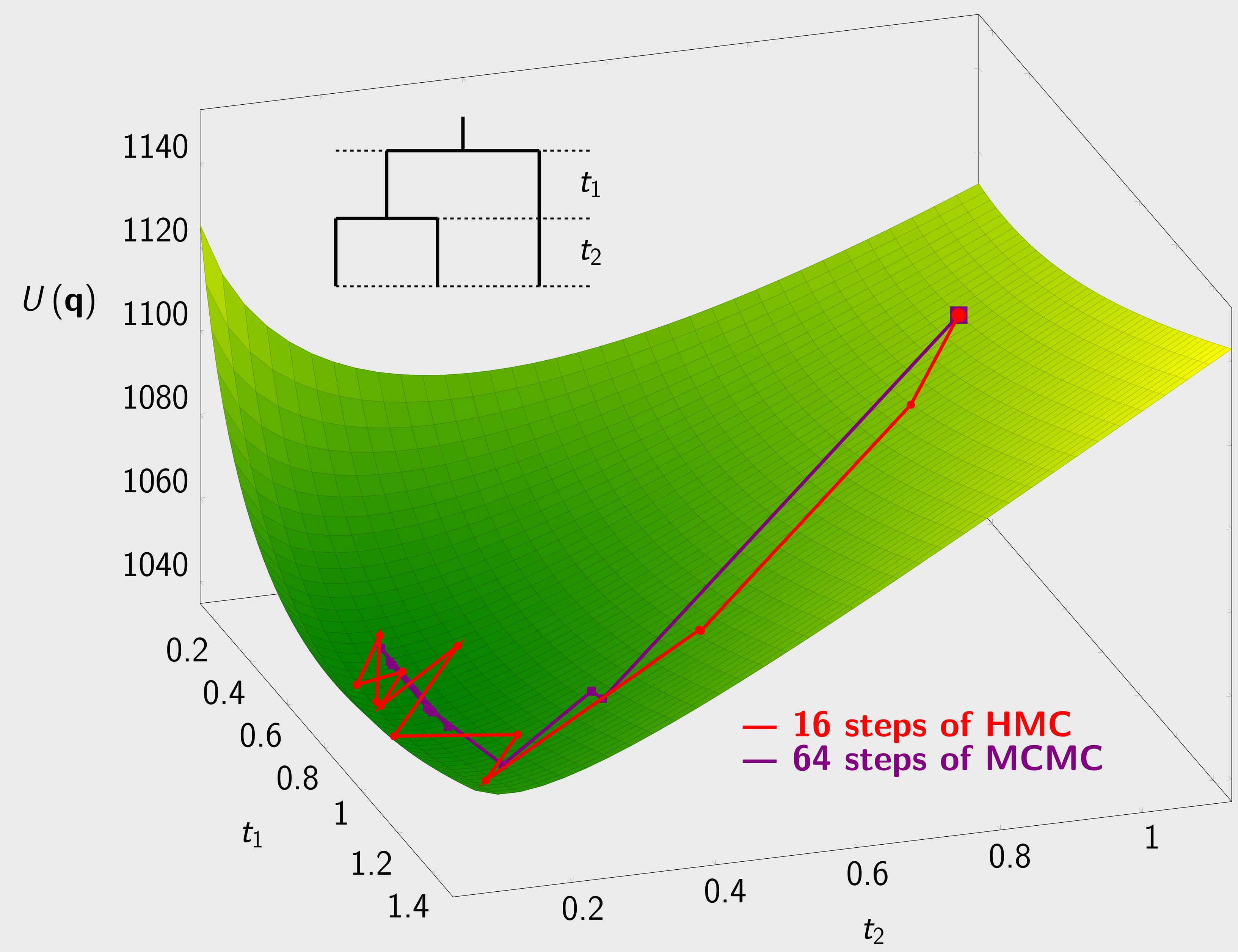
Momentum Randomization Operator

Changes the skater's momentum with a push in a random direction.

$$\mathbf{R}\{\mathbf{q}, \mathbf{p}\} = \left\{\mathbf{q}, \sqrt{1-\alpha}\mathbf{p} + \sqrt{\alpha}\mathbf{n}\right\}, \mathbf{n} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{M}^{-1}\right)$$

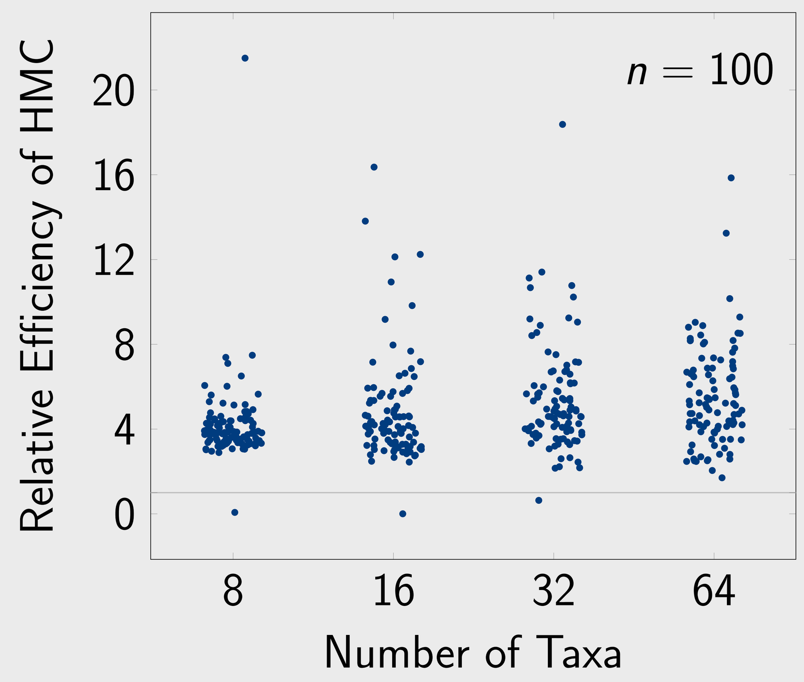


The Bayesian Skatepark



Performance of HMC versus MCMC

- ▷ Simulated 100 datasets for each problem size under Yule and HKY models
- ▷ Estimated node heights with HMC and MCMC, both optimally-tuned
- ▷ Measured efficiency as **effective sample size** of tree length per unit time
- ▷ ESS = number of independent samples
- ▷ HMC **consistently out-performed** MCMC for all problem sizes
- ▷ On average, HMC was **5 times more efficient** than MCMC



Future Work

- ▷ Inferring parameters for other evolutionary models; e.g., substitution and clock models
- ▷ Creating a skatepark with many tree topologies and the physics to move between them
- ▷ Automatic tuning of HMC for optimal performance, especially the skater's mass $\mathbf{M}$
- ▷ Implementing and testing more sophisticated flavors of HMC
- ▷ Applying HMC to the analysis of real datasets

Acknowledgements

- ▷ New Zealand eScience Infrastructure
- ▷ SMBE Undergraduate Travel Award
- ▷ Members of the Computational Evolution Group at The University of Auckland

References

AJ Drummond et al. *Mol Biol Evol* 29.8 (2012). doi:10.1093/molbev/mss075
S Duane et al. *Physics Letters B* 195.2 (1987). doi:10.1016/0370-2693(87)91197-X
N Metropolis et al. *J Chem Phys* 21.6 (1953). doi:10.1063/1.1699114
RM Neal. *Handbook of Markov Chain Monte Carlo.* Chapman and Hall/CRC (2011).
EE Schadt et al. *Genome Research* 8.3 (1998). doi:10.1101/gr.8.3.222

Download this poster.
doi:10.6084/m9.figshare.1473743

Fork the HMC source code.
http://git.io/vqlCz