Supporting material for 'A generalized residual technique for analysing complex movement models using earth mover's distance'

Jonathan R. Potts[1,a], Marie Auger-Méthé[1,2,b] Karl Mokross[3,4,c] Mark A. Lewis[1,2,d]

1 Centre for Mathematical Biology, Department of Mathematical and Statistical Sciences, University of Alberta, Canada

2 Department of Biological Sciences, University of Alberta, Edmonton, Canada

3 School of Renewable Natural Resources, Louisiana State University Agricultural Center, Baton Rouge, Louisiana, 70803

4 Projeto Dinâmica Biológica de Fragmentos Florestais. INPA. Av. André Araújo 2936. Petrópolis. Manaus. Brazil. 69083-000

a E-mail: jrpotts@ualberta.ca. Centre for Mathematical Biology, Department of Mathematical and Statistical Sciences, 632 CAB, University of Alberta, Canada, T6G 2G1. Tel: +1-780-492-1636.

b E-mail: marie.auger-methe@ualberta.ca

c E-mail: kmokro1@tigers.lsu.edu.

d E-mail: mark.lewis@ualberta.ca.

Appendix A: Instructions for running code

Running simulations

Simulations are coded in C, but can be run from within R (explained below). The file `dfr.exe` is ready to work on a Windows 64 bit computer. Users of 32 bit Windows operating systems should replace this file with `dfr32.exe`. Mac users should replace this file with `dfr_mac`. They should also replace all the `*.txt` files with the `*.txtm` versions (`txtm` stands for text-Mac) included in the SI. Users of other operating systems (e.g. Linux) will have to compile the source code themselves, which consists of all the attached `*.c` and `*.h` files. Linux/Unix versions of the text files are called `*.txtu` (text-Unix) rather than `*.txt`.

The programme `dfr.exe` enables the user to simulate paths in an environment with two layers that affect the movement as per Equation (8) in the main text. The user needs to construct these layers in advance of running the programme. Each layer should be saved as a plain text file with tabs separating the x-values and carriage returns separating the y-values. Examples are given in `random_field_scale_10.txt` and `random_field_scale_1000.txt`. The former described Layer 1 from the main text and the latter Layer 2.

The user also needs to construct a file denoting the environment-free jump distribution as a grid. The file `dfr_jump_probs_exp5.txt` provides an example of this. To aid construction, we provide both and R script `dfr_weibull_to_table.R` and a Python script `dfr_weibull_to_table.py`, which turns a Weibull distribution (a generalisation of the exponential distribution) to a grid useable for `dfr.exe`. To construct this using the R script, type the following into the R terminal

```
source("dfr_weibull_to_table.R")
```

The shape and scale parameters (`weibull_a` and `weibull_b`), together with the size of the jump array (`array_size_x` by `array_size_y`) and the lattice spacing (`lattice_spacing`) can be modified in the script.

For Python users, go to the DOS command line or Unix/Mac/Linux terminal and type the following

```
python dfr_weibull_to_table.py shape scale x y l > dfr_jump_probs_exp5.txt
```

where `shape` should be replaced with the Weibull shape parameter (e.g. 1 in the case in the paper), `scale` with the scale parameter (e.g. 5), `x` and `y` with the width and height (e.g. 79), respectively, of the jump distribution grid (outside this grid the jump probabilities are zero, to aid computational speed), and `l` with the distance between adjacent grid points in the layers (e.g. 1).

Finally, it is necessary to construct a file detailing the start positions of the simulated animals, which is a list of x- and y-values of the start positions with x in the left column and y in the right, as in `dfr_start_pos_random.txt`.

We first explain how to use the DOS command line or Unix/Mac/Linux terminal to perform example simulations, which is the recommended method if you are familiar with command line invocations. Then we explain how to do the same using R.

Using the command line/terminal. The following command will print out the positions of 100 different individuals, each performing 1000 steps according to the model in Eqn 4 of the main text, with $\alpha = 1.5$, $\beta = 10$.

```
dfr -l1 random_field_scale_10.txt -l2 random_field_scale_1000.txt -spf
dfr_start_pos_random.txt -jpf dfr_jump_probs_exp5.txt -l1w 1.5 -l2w 10 -kv 0
-i 100 -ts 1000 -bw 1000 -bh 1000 -jpw 40 -jph 40 > dfr.txt
```

The output of this simulation, which takes a couple of minutes to run, is found in `dfr.txt`.

Typing `dfr` with no parameters gives a usage statement, as follows, which explains how to make general use of this programme.

```
Usage:  dfr.exe [-l1 <layer-1>]
                [-l2 <layer-2>]
                [-jpf <jump-prob-file>]
                [-l1w <layer-1-weight>]
                [-l2w <layer-2-weight>]
                [-kv <k-val-for-von-Mises>]
                [-i <number-of-indivs>]
                [-spf <start-position-file>]
                [-ts <number-of-timesteps>]
                [-bw <box-width>]
```

```
                [-bh <box-height>]
                [-jpw <jump-prob-array-width>]
                [-jph <jump-prob-array-height>]
```

The parameter k is for using correlated random walks. It is the sole parameter of a von Mises distribution with mean 0, and gives the distribution of the turning angles between successive steps.

The user is encouraged to adapt the source code to make it useable for other scenarios (more layers, individual interactions etc.). However, when doing this, the authors kindly request that the user makes the code generally available and keeps the lead author informed of where it can be found. This will enable an *ad hoc* open source situation to evolve.

Using R. The following command will print out the positions of 100 different individuals, each performing 1000 steps according to the model in Eqn 4 of the main text, with $\alpha = 1.5$, $\beta = 10$.

```
x=system("dfr -l1 random_field_scale_10.txt -l2 random_field_scale_1000.txt
-spf dfr_start_pos_random.txt -jpf dfr_jump_probs_exp5.txt -l1w 1.5 -l2w 10
-kv 0 -i 100 -ts 1000 -bw 1000 -bh 1000 -jpw 40 -jph 40", ignore.stderr=TRUE,
intern=TRUE)
```

This takes a couple of minutes to run. Now put the output into a text file called `dfr.txt` using the following command

```
write.table(x, file="dfr.txt", quote=FALSE, row.names=FALSE, col.names=FALSE)
```

Typing `system("dfr")` gives a usage statement, as follows, which explains how to make general use of this programme.

```
Usage:  dfr.exe [-l1 <layer-1>]
                [-l2 <layer-2>]
                [-jpf <jump-prob-file>]
```

```
[-l1w <layer-1-weight>]
[-l2w <layer-2-weight>]
[-kv <k-val-for-von-Mises>]
[-i <number-of-indivs>]
[-spf <start-position-file>]
[-ts <number-of-timesteps>]
[-bw <box-width>]
[-bh <box-height>]
[-jpw <jump-prob-array-width>]
[-jph <jump-prob-array-height>]
```

The parameter k is for using correlated random walks. It is the sole parameter of a von Mises distribution with mean 0, and gives the distribution of the turning angles between successive steps.

Analysing data using EMD

In R, to find the EMDs between the each of the simulated data points in the file `dfr.txt`, created above, and the model used to create it, use the `dfr_emm.R` file. Users should feel free to alter the values of `pos_file`, `layer1`, `layer2`, `jump_prob`, `alpha`, and `betav`.

Python users should instead run the following from the command line:

```
python dfr_emm.py dfr.txt random_field_scale_10.txt random_field_scale_1000.txt
dfr_jump_probs_exp5_py.txt 1.5 10 > dfr_emm.txt
```

In general, the format of the input is as follows:

```
python dfr_emm.py [positional-data] [layer-1] [layer-2]

[jump-probability-grid] α β > <output-file>
```

The output of either the Python or the R script is a file `<output-file>` with four columns. The first is the EMD and the second is the standardised EMD. The third and fourth are the x- and y-coordinates of the vector $\hat{\mathbf{v}}_n$ from the main text (Eqn 5). This gives enough information for the user to construct wagon wheels, dharma wheels and so forth.

Please feel free to email JRP for any questions regarding these instructions.

Appendix B: Generalizing the Earth Mover's Distance (EMD)

The Main Text takes a pedagogical approach to introducing the EMD. Here, we take a more formal, mathematical approach, by introducing the Wasserstein Metric in its full generality and then proving that it gives Equation (1) from the Main Text with the assumptions stated therein.

Let (Ω, d) be a ~~measure~~ space (Ambrosio, 2005). Let p be a strictly positive integer. Let μ and ν be two probability measures on M with finite p-th moment. Then the p-th Wasserstein distance (Vasershtein, 1969) between μ and ν is

$$W_p(\mu,\nu) = \left[\inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\Omega\times\Omega} d(x,y)^p \mathrm{d}\gamma(x,y) \right]^{1/p}, \tag{1}$$

where $\Gamma(\mu,\nu)$ is the collection of probability measures on $\Omega\times\Omega$ whose first and second marginals are μ and ν respectively.

Suppose now that μ is the measure associated with the probability density function $P^M(\mathbf{X}|\mathbf{Y})$ from the Main Text and that $\nu(\mathrm{d}\mathbf{X}) = \delta[\mathrm{d}(\mathbf{X}-\mathbf{Y})]$, where δ is the Dirac delta measure. If the data are noisy then ν could represent a probability distribution with mean given by a data point and distribution reflecting the noise in the data. Equation (1) would be more complex to compute than if ν is a delta measure, but various methods exist (Ling & Okada, 2007). In general, it is guarenteed that the Wasserstein distance is well-defined only if μ and ν are Radon measures (Ambrosio, 2005). The Dirac delta measure is not a Radon measure as it is not locally finite. However, one of the consequences of the Theorem 1 (below) is that the definition is still valid in this particular case.

Let $\gamma \in \Gamma(\mu,\nu)$ and denote by F the function such that $\int_A F(\mathbf{A},\mathbf{B})\mathrm{d}\mathbf{A}\mathrm{d}\mathbf{B} = \int_A \gamma(\mathrm{d}\mathbf{A},\mathrm{d}\mathbf{B})$ for any measurable set A. Notice that F may not strictly be a real-valued function. For example, it could be a multiple of the Dirac delta 'function'. However, this is an oft-used abuse of notation convenient for calculations. It is also convenient to let δ denote the Dirac delta function as well as the Dirac delta measure, with the ambiguity cleared up by the context.

With these definitions, the following three results hold.

Lemma 1. *$F(\mathbf{A},\mathbf{B}) = P^M(\mathbf{A}|\mathbf{Y})\delta(\mathbf{B}-\mathbf{Y})$ for all $\mathbf{B} \notin S \cup Y$, where S is a set of measure zero. $F(\mathbf{A}, s)$ is finite for all $\mathbf{A} \in \Omega$, $s \in S$. If $F(\mathbf{A},\mathbf{Y}) \neq P^M(\mathbf{A}|\mathbf{Y})\delta(0)$ then $P^M(\mathbf{A}|\mathbf{Y}) = 0$.*

Proof. Suppose there exists $(\mathbf{A}',\mathbf{B}')$ such that $F(\mathbf{A}',\mathbf{B}') \neq P^M(\mathbf{A}')\delta(\mathbf{B}'-\mathbf{Y})$. First we look at the case when $\mathbf{B}' \neq \mathbf{Y}$. In this case $F(\mathbf{A}',\mathbf{B}') \neq 0$, since $\delta(\mathbf{B}'-\mathbf{Y}) = 0$. Since the second marginal of γ is $\delta[\mathrm{d}(\mathbf{A}-\mathbf{Y})]$, we have

$$\int_\Omega F(\mathbf{A},\mathbf{B})\mathrm{d}\mathbf{A} = \delta(\mathbf{B}-\mathbf{Y}). \tag{2}$$

It follows that

$$\int_\Omega F(\mathbf{A},\mathbf{B}')\mathrm{d}\mathbf{A} = 0, \tag{3}$$

so $F(\mathbf{A}',\mathbf{B}')$ must be finite, as F is non-negative. Furthermore, since $\mathbf{B}'$ is an arbitrary point not equal to $\mathbf{Y}$, F must be zero outside the subset $\mathbf{B} = \mathbf{Y}$ except on a set S of measure zero, and $F(\mathbf{A}, s)$ must be finite for all $\mathbf{A} \in \Omega$, $s \in S$.

Next suppose $\mathbf{B}' = \mathbf{Y}$. Then $F(\mathbf{A}',\mathbf{Y})$ is finite, since $\delta(\mathbf{B}'-\mathbf{Y}) = \infty$. Since $F(\mathbf{A}',\mathbf{B}) = 0$ for any $\mathbf{B} \notin S \cup Y$ and is finite for $\mathbf{B} \in S$, it follows that

$$\int_\Omega F(\mathbf{A}',\mathbf{B})\mathrm{d}\mathbf{B} = 0. \tag{4}$$

However, since the first marginal of γ is the probability measure associated with the probability density function $P^M(\mathbf{X}|\mathbf{Y})$, we have

$$\int_\Omega F(\mathbf{A}',\mathbf{B})\mathrm{d}\mathbf{B} = P^M(\mathbf{A}'|\mathbf{Y}). \tag{5}$$

Thus $P^M(\mathbf{A}'|\mathbf{Y}) = 0$. □

Lemma 2. *Let* $f : \Omega \times \Omega \to \mathbb{R}_{\geq 0}$ *be integrable. Then*

$$\int_{\Omega\times\Omega} f(\mathbf{A},\mathbf{B})F(\mathbf{A},\mathbf{B})\mathrm{d}\mathbf{A}\mathrm{d}\mathbf{B} \geq \int_{\Omega\times\Omega} f(\mathbf{A},\mathbf{B})P^M(\mathbf{A}|\mathbf{Y})\delta(\mathbf{B}-\mathbf{Y})\mathrm{d}\mathbf{A}\mathrm{d}\mathbf{B}$$

Proof. Let T be the set of points $\mathbf{A} \in \Omega$ such that $F(\mathbf{A},\mathbf{Y}) \neq P^M(\mathbf{A}|\mathbf{Y})\delta(0)$. Then, by Lemma 1,

$$\begin{aligned}
&\int_{\Omega\times\Omega} f(\mathbf{A},\mathbf{B})[F(\mathbf{A},\mathbf{B}) - P^M(\mathbf{A}|\mathbf{Y})\delta(\mathbf{B}-\mathbf{Y})]\mathrm{d}\mathbf{A}\mathrm{d}\mathbf{B} \\
&= \int_T f(\mathbf{A},\mathbf{Y})[F(\mathbf{A},\mathbf{Y}) - P^M(\mathbf{A}|\mathbf{Y})\delta(0)]\mathrm{d}\mathbf{A} \\
&= \int_T f(\mathbf{A},\mathbf{Y})F(\mathbf{A},\mathbf{Y})\mathrm{d}\mathbf{A} \geq 0. \qquad (6)
\end{aligned}$$

The final inequality comes from the fact that F and f are both non-negative. □

Theorem 1. *The p-th Wasserstein distance between* $\delta[\mathrm{d}(\mathbf{X}-\mathbf{Y})]$ *and the measure* μ *associated to* $P^M(\mathbf{X}|\mathbf{Y})$ *is*

$$W_p(\mu,\delta) = \left[\int_\Omega d(\mathbf{X},\mathbf{Y})^p P^M(\mathbf{X}|\mathbf{Y})\mathrm{d}\mathbf{X}\right]^{1/p}. \qquad (7)$$

Proof. Follows directly from Lemma 2 and the definition in Equation (1). □

Equation (1) from the Main Text is simply Equation (7) with $p = 1$. Fig 1 demonstrates what happens when $p \neq 1$. This allows the user to decide how to penalize the areas in and around the peaks of multi-modal distributions.

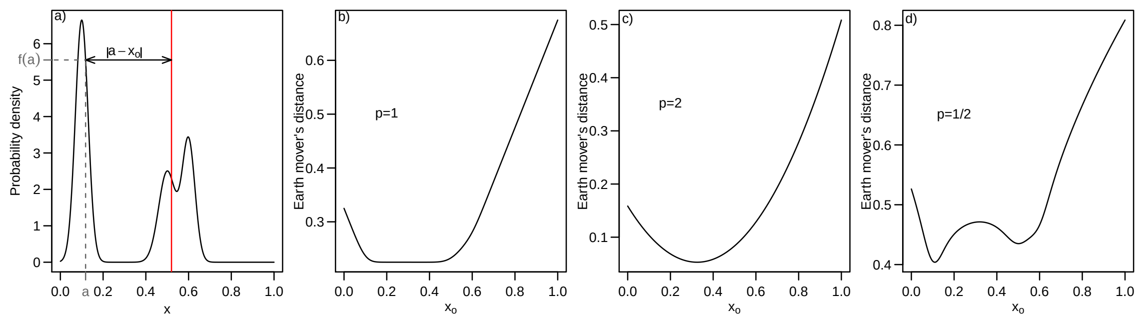

Fig. 1. The Wasserstein metric for arbitrary p**.** Panels (a) and (b) are figures 1b and 1c from the main text respectively. Panel (a) shows a probability distribution and a Delta function at $x = x_o$. Panel (b) shows the Wasserstein metric with d Euclidean and $p = 1$. Panel (c) has $p = 2$. Though the Wasserstein metric is not formally defined for non-integer p, it can still be calculated in certain cases. Panel (d) shows the case $p = 1/2$. These three cases demonstrate how to change p dependent upon whether the user wishes to penalize troughs in the middle of two peaks more than, less than, or the same as the peaks themselves.

Appendix C: literature search for movement models

Here we explain our search that led to the conclusion 'a search for the 20 highest cited papers that fit animal movement models to data reveals that none assess absolute quality' from the Main Text. We searched ISI Web of Knowledge for 'animal movement model' in the 'topic' field then refined our research results to incorporate just those research articles from the subjects 'Zoology' and 'Ecology' (search results are in the SI file 'MovementPapers.xlsx'). We discarded any papers that either did not fit a model to data, or did not include a model of animal movement. The 20 top cited papers according to these criteria were examined for assessment of absolute quality of the best fit movement model to reality. None of them made this assessment.

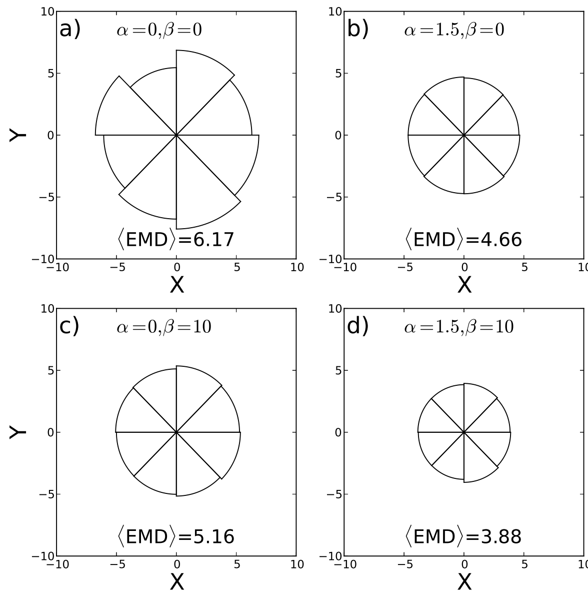

Fig. 2. Dharma wheels. The dharma wheels were all created using a simulated data set of the model in Equation (7) from the main text with $\alpha = 1.5$ and $\beta = 10$. The dharma wheels obtained by calculating the EMD from this data set to four different models of the form in Equation (7) are shown here. The mean EMDs, denoted $\langle \text{EMD} \rangle$, are given within the panels, together with the parameter values used. The latter correspond to models f_1, f_2, f_3, f_4 from the main text for panels (a,b,c,d) respectively.

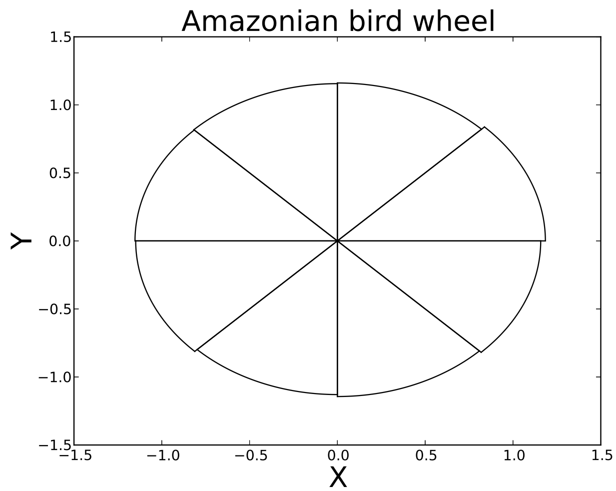

Fig. 3. Amazon bird dharma wheels. Stadardized dharma wheel of the best-fit model from Potts *et al.* (2014b) compared against the data from the same study.

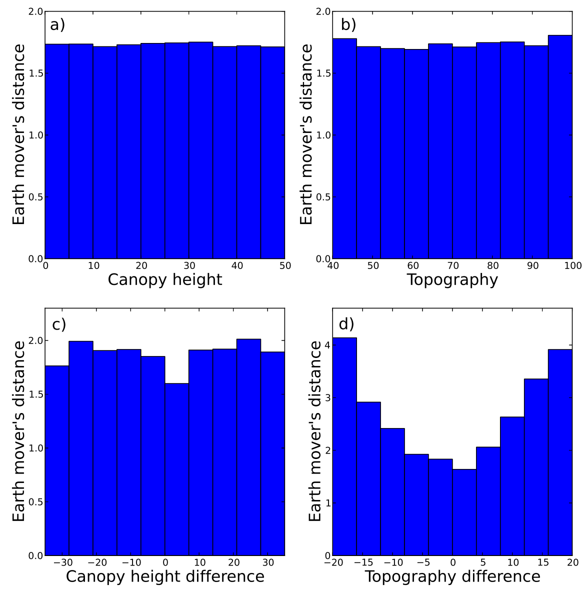

Fig. 4. EMD binned by values of environmental covariates. Figure (a) (resp. b) shows the EMDs of each step from the amazon bird data histogrammed by the canopy height (resp. topography) at the end of the step. Figure (c) (resp. d) shows the EMDs histogrammed by the canopy height (resp. topography) at the end of the step minus the canopy height (resp. topography) at the start of the step.

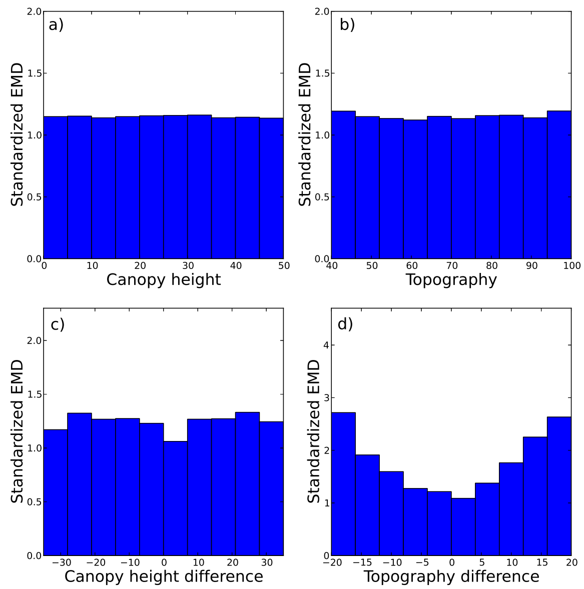

Fig. 5. SEMD binned by values of environmental covariates. Identical to SI Figure 4 but using Standardized EMD rather than EMD.

References

Ambrosio L, Gigli N, Savaré G (2005) *Gradient Flows in Metric Spaces and in the Space of Probability Measures*. Basel: ETH Zürich, Birkhäuser Verlag.

Ling H, Okada K (2007) An Efficient Earth Mover's Distance Algorithm for Robust Histogram Comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29:840-853

Potts, J. R., Mokross K., & Lewis M. A. (2014) A unifying framework for quantifying the nature of animal interactions. *J. Roy. Soc. Interface* (in press) Preprint at arXiv:1402.1802

Vasershtein LN (1969) Markov processes over denumerable products of spaces describing large system of automata. *Probl Inform Transm* 5:47-52.