PNAS
www.pnas.org

Supplementary Information for

Automatic Generation of Evolutionary Hypotheses Using Mixed Gaussian Phylogenetic Models

Venelin Mitov, Krzysztof Bartoszek and Tanja Stadler

Venelin Mitov.
E-mail: vmitov@gmail.com

This PDF file includes:

Supplementary text
Figs. S1 to S62
Tables S1 to S21
References for SI reference citations

www.pnas.org/cgi/doi/10.1073/pnas.1813823116

Supporting Information Text

Contents

A. Searching for an optimal MGPM.

A.1. Heuristics for reducing the MGPM search space. To reduce the search space of mixed phylogenetic models, $\mathcal{S}(\mathcal{T}, \mathcal{M})$, we use several "heuristics":

(A) Reducing the number of candidate shift-point configurations:

(A.1) Motivated by the usual lack of statistical power for inferring the precise location or the presence of multiple shifts within a branch (1, 2), we assume that a shift-point can only occur at the beginning of a branch. We call the end-node of such a branch a "shift-node". Following this assumption, the number of shift-point configurations is reduced to 2^{M-1}, with M denoting the number of nodes in the tree. Apart from this computational benefit, choosing the location of the shift to be the beginning of a branch is technically convenient for the pruning algorithm calculating the model likelihood, since there is no need to deal with shifts occurring at internal points of the branches. This heuristic can be relaxed if the tree has some very long branches and it is of interest to analyse the support for shifts occurring in the middle or at other internal points of the branches. In such cases, it is possible to introduce a finite number of singleton nodes at regular intervals within the branches. In other words, this would increase the time resolution for the inferred shift-points.

(A.2) We introduce a threshold, q, on the minimal number of tips "visible" from an ancestor shift-node, where visibility means that there is no other shift occurring on a path from the shift-node to any of these tips. Indirectly, this limits the maximum number of shifts to no more than N/q. However, specifying q instead of a maximum number of shifts has a performance benefit, because even a small value of q (e.g. less than 5% of the tree size) effectively reduces the space of possible shift configurations. As a downside, unlike a limit on the maximum number of shifts, specifying q hinders the detection of shifts visible from less than q tips. This is acceptable, given our goal is to detect patterns in big groups of tips, rather than outliers.

(A.3) The best configuration of a given size P (number of shift-nodes) can be obtained from the best configuration of size $P-1$ by adding one of the other possible shifts, namely, the one resulting in the best AIC score. This greedy assumption provides a stop criterion for the search procedure, namely, when a configuration has been reached, which's score cannot be improved by inserting a new shift. While not valid in general, this heuristic has proven useful in numerous previous implementations of stepwise AIC optimization on tree models, e.g. (3, 4).

(B) Reducing the number of possible model type mappings for a given shift-point configuration. For each candidate shift-point configuration comprising P shifts, i.e. $P+1$ regimes in total, there are $|\mathcal{M}|^{P+1}$ possible MGPMs. However, we notice that once a shift at a given node i is present, the best model type from $\mathcal{M}$ assigned to i is unlikely to affect the best model assignment for shift-nodes outside of i's clade. Intuitively, this follows from property 1 of the $\mathcal{G}_{LInv}$ family (see Definition in main text). As an example, consider Fig. 1A and assume for convenience that the encircled numbers are node identifiers. Think for a moment that we are at a step of the score optimization when we are considering placing a shift at node 10, all other shifts being placed as depicted. Placing a shift at node 10 reduces the set of tips visible from node 3 (visibility defined as in Heuristic A1). This is likely to affect the model type assigned to node 3, in particular, if the clade descending from 10 has evolved in a way different from the other part visible from 3. Due to property 1 of the $\mathcal{G}_{LInv}$-family (see Definition in main text), though, the model type that will be assigned to 10 has only a weak (indirect) effect on the model type assignment of the other shift-nodes. In particular, the only way in which the model for the clade descending from 10 affects the model assigned to node 3, is via the probability distribution it defines for the ancestral trait-value at the parent of node 10. This effect should be dominated by the cumulative effect from the other nodes visible from 3, which are more numerous (i.e. there are at least q of them, see Heuristic A.2). The effect on the other shift-nodes is only indirect, via changes in the model type and parameters for node 3. Therefore, when a new shift is inserted the optimal model type assignment for the nodes other than the ancestral shift-node is unlikely to change. Further in the text, we call "candidate shift-node" the node of the newly inserted shift (i.e. node 10 in the example), and we call "partition root" the ancestral shift-node (i.e. node 3 in the example). Following the above reasoning, we tested two possible ways to restrict the sets of candidate model types for a new shift-node configuration:

(B.1) For each shift-node j other than the partition root and the candidate shift-node, we test $\{M_{current-best-j}\}$, meaning the model types assigned to these nodes in the current best shift-node configuration; for the partition root we test the entire set $\mathcal{M}$, meaning no restriction on the possible model types; for the candidate shift-node i, we use $\{M_{clade-best-i}\}$, that is, the best model mapped to i in a single (non-mixed) model fit to i's clade. In this way, we reduce the above exponential complexity to $O(|\mathcal{M}|)$ in terms of number of ML fits.

(B.2) This is a "relaxed" variant of Heuristic B.1, in which the possible set of model types for all shift-nodes except the partition root is set to $\{M_{current-best}, M_{clade-best}\}$, whereas the whole set $\mathcal{M}$ is being tested for the partition root. This results in $O(|\mathcal{M}| \times 2^P)$ ML fits.

A.2. A recursive clade partition search algorithm. Using the above heuristics, we implemented a parallel recursive clade partition (RCP) algorithm searching for an optimal mixed Gaussian phylogenetic model (MGPM) over a finite subset $\mathcal{M} \subset \mathcal{G}_{LInv}$. This algorithm attempts to solve the inter-model shift problem by returning an (approximate) optimal inter-model shift configuration for a given tree and multivariate trait data at the tips (Algorithm S1, figs. S3-S4).

In Algorithm S1, we provide a pseudo-code description of the recursive clade partition search. To understand how the algorithm works, it may be useful to follow the search path for the optimal MGPM fit to the mammal tree shown on figs. S3-S4. We remind that, a shift occurs at the beginning of a branch leading to a so called "shift-node". We call "selected nodes" the nodes that have been selected as shift-nodes in the current best MGPM.

The algorithm starts with a ML fit of each model type to each clade of at least $q = 20$ tips, including the entire tree. The results of these model fits are stored in a data-table, which is used as a source for proposals of initial parameters in subsequent MGPM ML fits. Then, the algorithm initiates a queue of "partition root", starting with the root of the tree. Partition roots are equivalent to already selected shift-nodes. In each iteration of the main loop (line 12, algorithm S1), the partition root at the head of the queue is taken, and an attempt is made to improve the current best MGPM model by inserting a shift at one of its descendants. We call "candidates" the descendant nodes of a partition root, which have not been cut-out by (i.e. do not descend from) a previously selected shift-node and which satisfy the requirement that, after placing a shift on their corresponding branch, no regime (color) in the resulting tree would have less than q tips.

Each panel on Figs. S3-S4 shows the state at the beginning of a main-loop iteration. This state comprises the iteration number (number in parentheses), the AIC, log-likelihood and total number of parameters for the current best MGPM, the currently selected shift-nodes (colored points), the partition root (a colored point with a number equal to the iteration number), the candidate nodes (grey points) and the candidate model types for both, the selected and the candidate nodes (sets of capital letters in braces above the corresponding nodes). During the iteration, a maximum likelihood fit is performed for all MGPMs formed by adding one candidate (grey) node to the set of selected (colored) nodes and for all possible model mappings on this node and the currently selected shift-nodes. Note that this is a greedy step following Heuristic A.2. As an option it is possible to relax this heuristic by considering combinations of up to a user-specified number of candidate nodes. However, this would considerably slow down the search and was not tested. The set of possible model mappings for a configuration of shift-nodes can be formed as the Cartesian product of all candidate model types taken for each node. This, however, would result in an exponentially growing number of possible mappings. Thus, a reduction is made by using Heuristic B.1 or B.2. For the search-path example on Figs. S3-S4, the Heuristic B.2 is used. For the partition root, all model types are tested. For the other nodes, up to 2 model types are considered (the best model fit to the clade starting at the node vs the model assigned to the node in the current best fit). This effectively reduces the number of possible model mappings, although, in the worst case this number would still be exponential (see Heuristic B.2). If during the main-loop iteration the AIC has been improved by inserting a new shift-node, this shift-node, together with the partition root are inserted at the end of the queue, so that a further partition from these nodes can be explored in a next iteration. This step of the algorithm ends when the partition queue gets empty.

Optionally, it is possible to run an additional round-robin step (Step 3, Algorithm S1), in which the top-scoring one (or several) partition(s) are taken and for each position in each partition, each model type is tested in a row, keeping the remaining model types unchanged. If, after ML-fit of the MGPM so formed, the change of the model type for a given partition node results in a better score, this model type is retained and the algorithm continues with the next partition node. This step can be repeated several times, as long as the round-robin rounds over all partition nodes result in improved score. The goal of this step is to compensate for the possible negative effects of applying Heuristic B.2 during the recursive clade partition step.

Finally, we mention some potential drawbacks and directions for future work on the search algorithm:

- We did not consider other possible relaxations of Heuristic B.1. In particular, testing the entire set $\mathcal{M}$ for the candidate shift-node (in addition to the partition root), while restricting the set of models for each other shift-node j to $\{M_{current-best-j}\}$ could be a better choice than Heuristic B.2, because it has a polynomial complexity of $O(|\mathcal{M}|^2)$ ML fits while exploring more model types for the candidate shift-node.

- As pointed out by an anonymous reviewer, in its current version, the RCP algorithm does not allow to search for polyphyletic regimes on the tree. Polyphyletic regimes have been used in previous implementations, e.g. the SURFACE R-package (4) to model potential cases of convergent evolution. "Collapsing" several model regimes to form a polyphyletic regime (color) on the tree does not violate the $\mathcal{G}_{LInv}$ properties (see Definition in main text). Moreover, the PCMBase R-package has full support for calculating the log-likelihood and simulating MGPM models with polyphyletic regimes. Hence, at present, it is possible to test "collapsing" for some of the regimes in an inferred MGPM model by "adhoc" manipulation of the partition and regime assignment. An implementation of a "backward" step similar to SURFACE would be a useful enhancement.

A.3. A full search algorithm. Like every heuristic-based approach, the RCP algorithm comes with a potential risk of choosing a suboptimal configuration and model type assignment. To assess this risk, we implemented a "full" search algorithm that only uses the Heuristics A.1 and A.2. While this algorithm can be extremely slow on big trees, we were able to run it in reasonable time (within 12 hours) on small trees (N < 100), using the default setting for Heuristic A.2 ($q = 20$). We compare this full search algorithm to the RCP algorithm in Appendix I.

A.4. Likelihood optimization. We used calls of the R-function optim specifying the L-BFGS-B algorithm for optimizing the likelihood of each candidate MGPM. To reduce the risk of getting stuck in local optima, multiple runs have been performed starting from different locations in the parameter space. These starting locations were specified as follows:

- Clade fits: for the initial step of the algorithm, in which each model type is fit to each clade of not less than q tips, the likelihood of the MGPM was evaluated at a large number (in this case 150'000) of parameter points drawn at random

Algorithm S1 Recursive clade partition search (RCP) for an optimal MGPM

Input:
$\mathcal{T}$: a timed tree with M nodes of which N are tips;
$\mathcal{X} \in (\mathbb{R} \cup \{NA, NaN\})_{k \times N}$: data for k traits associated with the tips, missing values or non-existing traits allowed;
$\mathcal{M} \subset \mathcal{G}_k$, $|\mathcal{M}| < \infty$: a finite set of k-variate Gaussian phylogenetic models;
MLE: $\bigcup_{i \in \{0, N+1, \ldots, M-1\}} \mathcal{S}_i(\mathcal{T}_i, \mathcal{M}) \longrightarrow \{< \ell\ell^*, \Theta^* >\}$: a maximum likelihood estimator getting as input a MGPM on (a subtree of) $\mathcal{T}$ and returning the corresponding maximum likelihood, $\ell\ell^*$, and point estimate, Θ^*, of the parameters contained in S;
$SCORE$: $\{< S, \ell\ell >\} \longrightarrow \mathbb{R}$: a scoring function, penalizing a maximum likelihood value $\ell\ell$ based on the complexity (e.g. degrees of freedom) of S;
Output: A quasi-optimal MGPM, $S^* \in \{\mathcal{S}(\mathcal{T}, \mathcal{M})\}$, with respect to $SCORE$.

Data:
$TableFits$: a table with columns $tree$, $model$, Θ and q, containing the tree, the MGPM, the parameter-values and the penalized score for all MLEs produced during the search;
$QueuePartitionRoots$: a first-in-first-served list (queue) of the nodes used as clade-partition roots during the search;
S^*: the current MGPM on $\mathcal{T}$ with best score;

// ***Step 1. Initialization.*** *Fit each individual model to each clade in* $\mathcal{T}$.
foreach $i \in \{0, N+1, \ldots, M-1\}$ **do**
  foreach $m \in \mathcal{M}$ **do**
    $S_{i,m} \longleftarrow \{< i, m >\}$;
    $< \ell\ell^*_{i,m}, \Theta^*_{i,m} > \longleftarrow MLE(S_{i,m}; \mathcal{T}_i, \mathcal{X}_i, \mathcal{M})$;
    $q^*_{i,m} \longleftarrow SCORE(S_{i,m}, \ell\ell^*_{i,m})$;
    Add to $TableFits$ $\left\langle tree = \mathcal{T}_i, model = S_{i,m}, \Theta = \Theta^*_{i,m}, q = q^*_{i,m} \right\rangle$;

// ***Step 2. Recursive clade-partition search for the optimal MGPM on*** $\mathcal{T}$.
// ***Step 2.1.*** *Initialize* $QueuePartitionRoots$ *with root-node and the best individual model fit to* $\mathcal{T}$ *found in* $TableFits$.
Add to QueuePartitionRoots $< 0 >$;
$S^* \longleftarrow$ {model in $TableFits$ with the best score on the whole tree};
// Main loop
while $QueuePartitionRoots$ *is not empty* **do**
  // ***Step 2.2.*** *Get the node at the head of the queue: this node is the partition root for the iteration.*
  $j \longleftarrow PopFrontElement(QueuePartitionRoots)$;
  // ***Step 2.3.*** *Extract the subtree of* $\mathcal{T}$ *containing all tips descending from* j *without an intermediate node from* S^* *on their path to* j.
  $\mathcal{T}'_j \longleftarrow ExtractClade(\mathcal{T}, j)$;
  foreach $l \in Nodes(S^*) \backslash \{j\}$ **do**
    if $l \in Nodes(\mathcal{T}'_j)$ **then**
      $\mathcal{T}'_j \longleftarrow RemoveClade(\mathcal{T}'_j, l)$;
  $PartitionNodes \longleftarrow Nodes(\mathcal{T}'_j)$;
  // ***Step 2.4.*** *Make a list of all shift configurations including* $Nodes(S^*)$ *and a node from* $PartitionNodes$.
  $P \longleftarrow \phi$;
  foreach $p \in PartitionNodes$ **do**
    $P \longleftarrow P \bigcup \left\{ Nodes(S^*) \cup \{p\} \right\}$;
  // ***Step 2.5.*** *Restrict the sets of candidate models (the pseudo-code below implements Heuristic B.2)*
  foreach $l \in S^* \cup PartitionNodes \backslash \{j\}$ **do**
    $\mathcal{M}_l \longleftarrow$ {best model fit to clade l} $\cup$ {model assigned to l in S^*};
  $\mathcal{M}_j \longleftarrow \mathcal{M}$;
  // ***Step 2.6.*** *MLE fits to all shift configurations in* P *and possible model mappings using* $\mathcal{M}_l, l \in P$
  foreach $Sp \in P$ **do**
    foreach $Sm \in \prod_{l \in Sp} \mathcal{M}_l$ **do**
      $S \longleftarrow \{< Sp, Sm >\}$;
      $< \ell\ell^*_S, \Theta^* > \longleftarrow MLE(S; \mathcal{T}, \mathcal{X}, \mathcal{M})$;
      $q^* \longleftarrow SCORE(S, \ell\ell^*_S)$;
      Add to $TableFits$ $\left\langle tree = \mathcal{T}, model = S, \Theta = \Theta^*, q = q^* \right\rangle$;
  // ***Step 2.7.*** *If step 2.6 has found a fit with a better score, update* S^* *and add its nodes to the queue.*
  if $TableFits[tree == \mathcal{T}, Min(q)] < SCORE(S^*, \ell\ell_{S^*})$ **then**
    $S^* \longleftarrow BestModel(TableFits[tree == \mathcal{T}])$;
    Add to QueuePartitionRoots $Nodes(S^*)$;

// ***Step 3. Round robin search for the optimal model type assignment to*** $Nodes(S^*)$.
// This step is optional and can be repeated a user-specified number of times.
foreach $i \in Nodes(S^*)$ **do**
  foreach $m \in \mathcal{M}$ **do**
    $S^{*'} \longleftarrow S^* \backslash \{< i, \cdot >\} \cup \{< i, m >\}$;
    $< \ell\ell^{*'}, \Theta^{*'} > \longleftarrow MLE(S^{*'}; \mathcal{T}, \mathcal{X}, \mathcal{M})$;
    $q^{*'} \longleftarrow SCORE(S^{*'}, \ell\ell^{*'})$;
    if $q^{*'} < SCORE(S^*, \ell\ell^*)$ **then**
      $S^* \longleftarrow S^{*'}$;

return S^*;

from a uniform distribution defined by user-specified limits (see Parameter limits in the Model parametrizations section C). Then the points were sorted in order of decreasing likelihood and optim was run for the top 400 points.

- Main loop fits: for the main loop MGPM fits, we implement a similar procedure as for the clade fits, but with reduced number of likelihood evaluations (4000) and 20 optim calls. The starting locations have been chosen from a mixture of randomly drawn parameters and slightly modified (jittered) optimum points from the clade fits for each shift-node and mapped model type.

A.5. Parallel execution. We implemented parallel execution of the nested foreach loops in step 1 (line 2) and step 2.6 (line 29), and for the inner foreach loop in step 3 (line 43) in algorithm S1. Parallelization was implemented within the PCMFit R-package via calls to the R-packages foreach (5), iterators (6) and doMPI (7). The MGPM fit for both the mammal data and the simulated data (Appendix section I) was performed on the Euler cluster managed by the HPC team at ETH Zurich. For the analysis of the mammal dataset the search algorithm finished within 24 hours, running on 300 cores (299 MPI worker nodes).

B. Calculating the AIC of a MGPM ML fit. For a ML fit of the MGPM model, the Akaike information criterion is given by the formula

$$AIC = -2\ell\ell^* + 2p \qquad [S1]$$

where $\ell\ell^*$ denotes the maximum log-likelihood and p denotes the total number of model parameters. For the MGPM on a fixed tree and data, we define p as the total number of numerical model parameters, that is, the initial trait vector, $\vec{X}_0$ together with the parameters for each model regime, plus $[2 * (R-1) + 1]$, where R denotes the number of regimes. In this way, every shift counts as $(p_m + 2)$ added parameters, where p_m denotes the number of numerical parameters in the model mapped to the shift-point and 2 corresponding to the fact that the shift location and as well as the type of the mapped model are treated as free parameters (note, however, that only one parameter is counted for the root, because this node is present in every shift-configuration and only the model-type is a free parameter for that node).

In compliance with a reviewer's request, we've also considered a relaxed version, hereby denoted AIC2, in which the choice of model type at a shift-point is not penalized.

C. Model parametrizations.

C.1. The Ornstein-Uhlenbeck process is a $\mathcal{G}_{LInv}$- process. In particular, the elements $\vec{\omega}_{s,t}$, $\mathbf{\Phi}_{s,t}$ and $\mathbf{V}_{s,t}$ from property 2 of the $\mathcal{G}_{LInv}$-family (see Definition in main text) are given by (12):

$$\begin{aligned} \vec{\omega}_{s,t} &= \left(\mathbf{I} - \mathrm{Exp}\big(-(t-s)\mathbf{H}\big)\right)\vec{\theta} \\ \mathbf{\Phi}_{s,t} &= \mathrm{Exp}(-(t-s)\mathbf{H}) \\ \mathbf{V}_{s,t} &= \int_0^{t-s} \mathrm{Exp}(-v\mathbf{H})(\mathbf{\Sigma}_u\mathbf{\Sigma}_u^T)\mathrm{Exp}(-v\mathbf{H}^T)dv \end{aligned} \qquad [S2]$$

C.2. Transformations for the matrix parameters $\mathbf{\Sigma}$ and $\mathbf{H}$. We used transformations for the matrix parameters $\mathbf{\Sigma}$ and $\mathbf{H}$ to prevent the likelihood optimization from hitting on invalid parameter values (e.g. a matrix $\mathbf{\Sigma}$ which is not symmetric positive definite, or a matrix $\mathbf{H}$, which is not negative-definite). We note that analogical techniques described briefly below have been used in other OU implementations, e.g. (8, 9).

For the unit-time variance-covariance matrices, $\mathbf{\Sigma}$, which are symmetric positive definite by definition, we used the parametrization:

$$\mathbf{\Sigma} = \mathbf{\Sigma}_u\mathbf{\Sigma}_u^T, \qquad [S3]$$

where $\mathbf{\Sigma}_u$ denotes an upper triangular matrix with positive elements on its diagonal*. Note that, as long as the diagonal elements of $\mathbf{\Sigma}_u$ are positive, Eq. S3 guarantees that $\mathbf{\Sigma}$ is a symmetric positive definite matrix.

For the OU selection strength matrices $\mathbf{H}$, which we require to have non-negative eigenvalues without necessarily being symmetric (note that negative eigenvalues result into repulsion from the $\vec{\theta}$), we used the Schur parametrization following (9). Specifically, we define a $k \times k$-dimensional matrix $\mathbf{H}_S$ as follows:

- the upper triangle of $\mathbf{H}_S$, excluding the diagonal, specifies $k(k-1)/2$ rotation angles for Givens rotations (10) to obtain a $k \times k$-dimensional orthoganal matrix $\mathbf{Q}$;
- the lower triangle of $\mathbf{H}_S$ including the diagonal defines a $k \times k$ triangular matrix $\mathbf{T}$.

Then, $\mathbf{H}$ is obtained from $\mathbf{Q}$ and $\mathbf{T}$ as follows (8, 9):

$$\mathbf{H} = \mathbf{Q}\mathbf{T}^T\mathbf{Q}^T \qquad [S4]$$

The matrix $\mathbf{H}$ calculated in this way has all of its eigenvalues equal to the elements on the diagonal of $\mathbf{H}_S$ (8, 9). Thus, by restricting the diagonal of $\mathbf{H}_S$ to non-negative values, we guarantee that $\mathbf{H}$ will have all of its eigenvalues non-negative. Further, if $\mathbf{H}_S$ is diagonal, then so is be the matrix $\mathbf{H}$; if $\mathbf{H}_S$ is upper triangular, then $\mathbf{T}$ is diagonal and the resulting matrix $\mathbf{H}$ is symmetric. Finally, if $\mathbf{H}_S$ is a full matrix, i.e. neither diagonal nor triangular, then the resulting matrix $\mathbf{H}$ is asymmetric.

*This parametrization is analogical but not identical with the Cholesky factorization ($\mathbf{\Sigma} = \mathbf{\Sigma}_l\mathbf{\Sigma}_l^T$, where $\mathbf{\Sigma}_l$ is a lower triangular matrix with positive elements on the diagonal, see, e.g., (8, 9)). In particular, the matrix $\mathbf{\Sigma}_u$ is not necessarily equal to the upper triangular Cholesky factor ($\mathbf{\Sigma}_l^T$) of $\mathbf{\Sigma}$. Note also that, due to historical reasons, the PCMBase R-package uses the name `$Sigma_x` instead of `$Sigma_u` for the parameter $\mathbf{\Sigma}_u$.

C.3. Parameter limits. Since we used the L-BFGS-B algorithm for gradient-descent optimization (11), we need to specify limits for the model parameters. We did this as follows:

- $0.0 \leqslant \mathbf{\Sigma}_{u,ii} \leqslant 1$, $i \in \{1,2\}$ for all model types;
- $0.0 \leqslant \mathbf{\Sigma}_{u,12} \leqslant 1$ for all model types;
- $0.0 \leqslant \mathbf{H}_{S,ii} \leqslant 10$, $i \in \{1,2\}$ for all OU model types;
- $-10.0 \leqslant \mathbf{H}_{S,12} \leqslant 10.0$ for the OU_E model type (keeping $\mathbf{H}_{S,21} = 0$ to ensure symmetry of the transformed matrix $\mathbf{H}$);
- $-10.0 \leqslant \mathbf{H}_{S,ij} \leqslant 10.0$, $i \neq j \in \{1,2\}$ for the OU_F model type;
- $0.0 \leqslant \theta_1 \leqslant 7.8$ according to the range of lg-body-mass in grams in the mammal dataset;
- $-1.2 \leqslant \theta_2 \leqslant 4.2$ according to the range of lg-brain-mass in grams in the mammal dataset.

D. Calculating expected trait distributions under the MGPM. We use the fact that, under a MGPM of the evolution of k traits along a tree $\mathcal{T}$, the expected distribution of the trait values at any time point, i, on any branch of $\mathcal{T}$ is a k-variate Gaussian distribution. The mean k-vector and the $k \times k$ variance-covariance matrix of this distribution are functions of the initial (root) trait vector, $\vec{X}_0$ and the model parameters and branch lengths for the sequence of regimes on the path from the root to i. These functions are calculated by applying the Definition of the $\mathcal{G}_{LInv}$-family (see Definition in main text) in the following recursive fashion:

1. Node 0 (the root of $\mathcal{T}$) is associated with a k-variate Dirac's δ with infinite density over the root-value and 0 density elsewhere:
$$\begin{aligned} \mathrm{E}\left[\vec{X}_0\right] &= \vec{X}_0, \\ \mathrm{Var}\left[\vec{X}_0\right] &= [0]_{k\times k}. \end{aligned} \qquad [S5]$$
2. For any other point i, let j be the closest ancestor of i and t and s be their corresponding time distances from the root. Then the expected distribution of the trait vector at i, $\vec{X}_i$ is a k-variate Gaussian with mean and variance given by:
$$\begin{aligned} \mathrm{E}\left[\vec{X}_i\right] &= \vec{\omega}_{s,t} + \mathbf{\Phi}_{s,t}\,\mathrm{E}\left[\vec{X}_j\right], \\ \mathrm{Var}\left[\vec{X}_i\right] &= \mathbf{V}_{s,t} + \mathbf{\Phi}_{s,t}\,\mathrm{Var}\left[\vec{X}_j\right]\mathbf{\Phi}_{s,t}^T, \end{aligned} \qquad [S6]$$
where $\vec{\omega}_{s,t}$, $\mathbf{\Phi}_{s,t}$ and $\mathbf{V}_{s,t}$ are defined as in the Definition of the $\mathcal{G}_{LInv}$-family (see Definition in main text).
3. For any pair of distinct points (m,n) on $\mathcal{T}$, let o be (m,n)'s most recent common ancestor node and let the sequences of nodes $< i_1 = m, i_2 = parent(i_1), ..., o = parent(i_{l_m}) >$ and $< j_1 = n, j_2 = parent(j_1), ..., o = parent(j_{l_n}) >$ denote (m,n)'s paths to o in root-wise direction. Then, following the properties 1 and 2 of the Definition of the $\mathcal{G}_{LInv}$-family (see Definition in main text), the covariance of the trait vectors $\vec{X}_m$ and $\vec{X}_n$ is given by the equation
$$\mathrm{Cov}\left[\vec{X}_m, \vec{X}_n\right] = \left(\prod_{r=i_1}^{i_{l_m}} \mathbf{\Phi}_{s_r,t_r}\right) \mathrm{Var}\left[\vec{X}_o\right] \left(\prod_{v=j_1}^{j_{l_n}} \mathbf{\Phi}_{s_v,t_v}\right)^T, \qquad [S7]$$
where for $w \in \{r,v\}$, s_w and t_w denote the times (summed up branch-distances to the root) for the nodes $parent(w)$ and w, and the matrices $\mathbf{\Phi}_{s_w,t_w}$ are defined as the matrices $\mathbf{\Phi}_{s,t}$ in the Definition of the $\mathcal{G}_{LInv}$-family (see Definition in main text).

Using Eqs. S5-S7 it is possible to calculate the moments, that is, the Mk-mean vector and the $(Mk \times Mk)$-variance-covariance matrix of the Mk-variate Gaussian distribution expected under an MGPM, M denoting the total number of nodes (root, tips and internal nodes) in $\mathcal{T}$. These have been implemented in the functions `PCMMean()` and `PCMVar()` from the PCMBase R-package (12). In SI Appendix, Section I, we use these properties to quantify the similarity between the Gaussian distributions expected under a true and and inferred MGPM.

Finally, we note that calculating the moments of the Mk-variate Gaussian distribution expected under a MGPM provides a way to calculate the model likelihood by applying the standard formula for the multivariate Gaussian density. While this approach has been used in previous PCM implementations (see, e.g. (9) and references therein), it does not scale to trees above several dozens to a hundred tips, due to its computational complexity exceeding $O(M^2)$, as well as numerical errors with the inversion of the resulting $(Mk \times Mk)$ variance covariance matrix. As a faster and numerically stable alternative, the our implementation relies on an $O(M)$ pruning algorithm for the likelihood calculation of $\mathcal{G}_{LInv}$-models, based on analytical integration over the unobserved trait values at the internal nodes of the tree (12). This algorithm is implemented in the accompanying PCMBase R-package (https://venelin.github.io/PCMBase) (12).

E. Ordinary least squares regressions. For calculating the linear regression lines of lg-brain-mass on lg-body-mass (fig.2), we use the fact that for a bivariate normal distribution of two variables x and y with mean vector $\vec{\mu} = [E(x), E(y)]^T$ and variance covariance matrix $\mathbf{V} = \begin{bmatrix} \sigma^2(x) & \sigma(x,y) \\ \sigma(x,y) & \sigma^2(y) \end{bmatrix}$, the linear regression of y on x, i.e. the linear model $y = a + bx + \epsilon$, has ordinary least squares (OLS) estimates for the slope (b) and intercept (a) given by the equations:

$$\begin{aligned} b &= \sigma(x,y)/\sigma^2(x) \\ a &= E(y) - bE(x). \end{aligned} \quad [S8]$$

Using eq. S8, we calculated the intercept and the slope of the OLS regressions of lg-brain-mass on lg-body-mass in the mammal data as follows (see also fig. 2):

(a) First, we calculated the expected distributions of the two traits under the inferred MPGM fit in the different backbone lineages (fig. 2) at seven past points in time located at regular intervals of 27 Ma, starting from -162 Ma and ending at the present time. For the calculation, we used the expected mean vector, $\vec{\mu} = [E(lg-body-mass), E(lg-brain-mass)]^T$ and variance-covariance matrix $\mathbf{V}$ under the MGPM fit (Eqs. S5- S7, Appendix D).

(b) We additionally calculated a "standard" ordinary least squares (OLS) regression based on the tip trait values, ignoring the phylogenetic relationship in the tree. We note that these OLS regressions assume independency of the tips, neglecting the correlation due to shared ancestry between the species in each regime. This is a known source of bias (13) motivating the use of PCMs. To do the OLS calculation, we used eq. S8, plugging in the empirical mean and variance covariance matrices. We cross-validated the resulting values for the coefficients with the slope and intercept obtained from calling the R-function lm.

(c) Third, we calculated the empirical measurements of the two traits for all 629 tips in the tree, using the empirical mean and variance-covariance matrices. The resulting OLS estimates matched the values used as a reference for the calculation of encephalization quotient (EQ) in (14). Again, we stress that this regression line (dashed grey regression lines on fig. S2) are calculated without accounting for the phylogenetic correlation between the tips.

F. Third party libraries. The software packages accompanying this article rely on a number of third party libraries: ape (15), Armadillo (16), expm (17), mvtnorm (18), data.table (19), ggplot2 (20), ggtree (21), ggimage (22), foreach (5), doMPI (7), iterators (6), digest (23), Rcpp (24). Additional tools used to generate the simulation data and to produce the figures and tables include phytools (25), cowplot (26), knitr (27), rmarkdown (28) and xtable (29).

G. Artistic images used in fig. 2 and SI Appendix, fig. S2. For Fig. 2 and SI Appendix, fig. S2, monochrome silhouette images of different species were downloaded from http://phylopic.org. The vectorized or raster images were re-colored using Microsoft Office or Adobe Illustrator. The images are licensed either under the Public Domain Mark 1.0 license (herebey abbreviated as PDM 1.0) available at https://creativecommons.org/publicdomain/mark/1.0/, or under the Public Domain Dedication 1.0 license (hereby abbreviated as PDD 1.0) available at https://creativecommons.org/publicdomain/zero/1.0/, or under the Creative Commons Attribution 3.0 Unported license (hereby abbreviated as CCAU 3.0) available at http://creativecommons.org/licenses/by/3.0/. Below, we list the images that were used, their authors, phylopic.org-ids and licenses:

- Cricetidae by Natasha Vitek: 81930c02-5f26-43f7-9c19-e9831e780e53, PDM 1.0;
- Hystricognathi (uncredited): 4c614ade-8710-400b-8045-ea1c9be4e7f2, PDM 1.0;
- Muridae by Daniel Jaron: 92989e35-4e68-4a2d-b3a2-191ba9da671a, PDD 1.0;
- Haplorrhini (1) by Gareth Monger: 24230275-1bfa-4ec2-a946-ca1ececdf216, CCAU 3.0;
- Haplorrhini (2, Homo sapiens sapiens) by T. Michael Keesey 2b4c32f6-99d0-43ba-9180-8013aa5bccd2, PDD 1.0;
- Cercopithecidae (uncredited), eccbb404-c99f-41f9-8785-01a7f57f1269, PDM 1.0;
- Marmotini by T. Michael Keesey 61440e34-7d24-4607-8479-2708ac45663f, PDD 1.0;
- Cetartiodactyla (1, Artiodactyla) by T. Michael Keesey (after C. De Muizon) 407f51d5-aa40-4e71-a5a7-7a6d6f328b5d, PDD 1.0;
- Cetartiodactyla (2, Cetacea) by Scott Hartman e68270c1-3091-4aee-92ae-51341a40e94a, PDD 1.0;
- Cetartiodactyla (3, Hippopotamus amphibius) by Jan A. Venter, Herbert H. T. Prins, David A. Balfour & Rob Slotow (vectorized by T. Michael Keesey) , 6336f90c-8f02-48f5-94d1-1d85c0100473, CCAU 3.0;
- Microchiroptera by Yan Wong, 18bfd2fc-f184-4c3a-b511-796aafcc70f6, PDD 1.0;
- Soricidae by Becky Barnes, 822c549b-b29b-47eb-9fe3-dc5bbb0abccb, PDD 1.0;

- Sciuridae by Catherine Yasuda, 5ebe5f2c-2407-4245-a8fe-397466bb06da, PDD 1.0;
- Feliformia (uncredited), ec56fa32-947b-4f0c-976b-c456132f2d6e, PDD 1.0;
- Diprotodontia by Michael Scroggie, f5592cab-cc61-4aab-b1dd-fba7cd2df7c9, PDD 1.0;
- Euarchonta by T. Michael Keesey (after Joseph Wolf), 88a07585-846a-405d-9195-c15c010e7443, PDD 1.0;
- Elephantidae by T. Michael Keesey, a15244a4-ecaa-4891-b870-31e5c8d9b5b3, PDD 1.0;

At the time of writing this document, each of the silhouette images could be accessed from the web-address http://phylopic.org/image/<image-phylopic.org-id>/, upon substituting the term "<image-phylopic.org-id>" with the image phylopic.org-id, e.g. http://phylopic.org/image/ec56fa32-947b-4f0c-976b-c456132f2d6e/.

H. Analysis of the mammal tree and data. All programming scripts and datasets for the mammal data analysis were collected within an R-package called MGPMMammals, available at https://github.com/venelin/MGPMMammals. Below, we describe the steps in the analysis.

H.1. Preparation of the mammal tree. As a preprocessing step, we repeatedly split, through insertion of singleton nodes, all branches in the tree longer than 16 Ma, until all branches were shorter than 16 Ma. This enabled the detection of shifts at internal points of the long branches in the tree, alleviating the possible negative effect of heuristic A.1, which restricts the location of shift-points to the beginning of each branch. The final tree used in all analyses of mammal data was an ultrametric tree of 629 tips, 1063 internal nodes, of which 494 were annotated ancestral bifurcating or multi-furcating nodes (30), and 569 were artificially inserted singletons. This tree is stored as an object of class "phylo" in the file MGPMMammals/data/tree.rda.

H.2. Preparation of the brain- and body-mass data. We collected the mammal trait values from Boddy et al. 2012 (14) available at https://onlinelibrary.wiley.com/action/downloadSupplement?doi=10.1111%2Fj.1420-9101.2012.02491.x&file=JEB_2491_sm_TableS1.xlsx.

The original analysis in (14) did not account for measurement error in the brain- and body-mass measurements. Following a suggestion from a reviewer that neglecting measurement error could potentially bias the inferred models towards OU model types, we enriched the original dataset with measurement error for each trait/species. To that end, we used the sample sizes and sample standard deviations that were available for some of the species in the original studies used as primary data-source by (14, 31). At the time of writing this article, the raw datafile for (31) could be downloaded from this URL: https://datadryad.org/bitstream/handle/10255/dryad.37960/brain_body_database_v2.txt?sequence=1. In order to estimate the measurement error for each trait/species, we performed the following preprocessing steps (the programming code for this preprocessing is located in the file MGPMMammals/data-raw/PreprocessMammalData.Rmd):

Filtering the data in (31). Given that the record ids from (31) included in the main analysis in (14) were not available[†], we repeated the filtering procedure as described in (14). In particular, we included only the records in (31) for which:

- all measurements were from adult individuals;
- published data were obtained from its original source;
- there was no note for emaciation.

After applying this filtering, we noticed that some measurements without a note for emaciation were still excluded from the main analysis in (14). We could not find an instruction how this was done in (14). Hence, to achieve a maximum match between the data records included in the main analysis in (14) and the records used to estimate measurement error, we applied a secondary filtering step:

1. Denote by data.BoddyEtAl.1 the aggregated dataset from (14) (Supplementary data file: JEB_2491_sm_TableS1.xlsx). Denote by data.BoddyEtAl.2 the data file from (31) resulting after applying the filtering described above.
2. for each species in data.BoddyEtAl.2, we sort the records in decreasing order of their values for body-mass (column Body.Mass..g.).
3. starting from the first record in that order, we consecutively add the next records, until either all records for the species have been added or the mean brain mass and the mean body-mass from the records currently added becomes equal to the brain mass and body-mass values reported in data.BoddyEtAl.1.

Taking the arithmetic means for the two traits for each of the records selected in this way resulted in matching brain- and body-mass values in grams for 599 out of 630 species (absolute difference smaller than 1e–6 for brain mass in grams and 1e–4 for body-mass in grams). For the remaining species, the difference was slightly bigger but negligible on the logarithmic scale[‡]

[†] We failed in our attempt to contact the authors.

[‡] We note that, technically, it would be more accurate to use a weighted instead of an arithmetic mean with weights corresponding to the sample sizes for the different records for a given species. We did not find an indication that this was done in (14) and we did not do this either, because our goal was to perform the analysis on the same trait values as in (14).

Aggregating multiple records for the same species. For the records where standard deviation and sample size were available, we calculated the standard error for using the formula $e_i = SD_i/\sqrt{(n_i)}$, i denoting a species/record/trait SD_i denoting the standard deviation and n_i denoting the sample size. Given that (31) can contains multiple records for the same species/trait an aggregation was needed to obtain a single standard error per species/trait. Baker 1963 (32) derived a formula for calculating the combined standard error from the standard errors, the means and the sizes of two samples:

$$e^2 = \frac{1}{n(n-1)}\left[n_1(n_1-1)e_1^2 + n_2(n_2-1)e_2^2 + \frac{n_1 n_2}{n}(m_1-m_2)^2\right],$$

where $n = n_1 + n_2$ is the sum of the sample sizes, m_1 and m_2 are the means of the two samples, e_1 and e_2 are the standard errors in the two samples and e is the standard error of the combined mean $m = (n_1 m_1 + n_2 m_2)/n$.

After applying this aggregation, we obtained standard errors (SE) of the body-mass in grams for 144 species and of the brain-mass in grams for 87 species.

Using linear regression to impute the missing standard errors. Using the function lm from the package stats in R, we did linear regression of the body-mass SE on the body-mass as shown in the following listing summarizing the linear model:

```
##
## Call:
## lm(formula = Body.Mass.SE ~ 0 + Body.Mass..g.)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -188936    -601     -54      -1  249762
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## Body.Mass..g.  0.16537    0.00622    26.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34800 on 143 degrees of freedom
## Multiple R-squared:  0.832,  Adjusted R-squared:  0.83
## F-statistic:  706 on 1 and 143 DF,  p-value: <2e-16
```

Note that this linear regression model had a considerably high $R^2_{adj} = 0.83$, suggesting that standard error for body-mass scales linearly with the body-mass itself. We fit a similar linear model for the brain-mass SE, this time including both, the brain- and the body-mass as predictor variables (this resulted in even higher $R^2_{adj} = 0.93$):

```
##
## Call:
## lm(formula = Brain.Mass.SE ~ 0 + Brain.Mass..g. + Body.Mass.SE)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -58.03  -2.89  -0.16   2.10  43.21
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## Brain.Mass..g. 4.23e-02   2.15e-03    19.7   <2e-16 ***
## Body.Mass.SE   1.69e-04   9.14e-06    18.5   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.4 on 85 degrees of freedom
## Multiple R-squared:  0.938,  Adjusted R-squared:  0.936
## F-statistic:  642 on 2 and 85 DF,  p-value: <2e-16
```

Then, we used these two models to compute the missing estimates of standard errors. We did this by plugging-in the brain- and body-mass measurements for the corresponding species in the two models.

Transforming the trait values and standard errors to the log-10 scale. Given that the original values for brain- and body-mass as well as their corresponding standard errors were in grams, we performed a log-10 transformation. This is a commonly used transformation for quantitative traits to achieve approximate normality of the trait distribution. Boddy et al. (14) used a simple log-10 transformation for the trait values. While this is acceptable for the trait values, it involves a non-trivial transformation for the standard errors (33). To understand this, consider the simple fact that a standard error should always be non-negative (even on the logarithmic scale), whereas the 10-th logarithm of a value between 0 and 1 is negative. Hence, a trivial log-10 transformation for a standard error is invalid. Following (33), we assume that for a given species, each of the two traits considered on the scale of grams has a sample mean that is log-normally distributed with mean equal to the true mean value for the species and standard deviation, equal to the true standard error for the species. Let's denote by Y any one of the two traits measured in a sample of organisms from a given species on the scale of grams. Assuming that the sample mean $\bar{Y}$ is log-normally distributed means that the variable $Z = \ln(\bar{Y})$ is normally distributed with mean and variance equal to μ_Z and σ_Z^2 for some values of these two parameters. We note that, since the coefficient of variation $CV_{\bar{Y}} = \frac{\sigma_{e,\bar{Y}}}{\mu_{\bar{Y}}}$ is small, the log-normal distribution has a bell-shaped density very similar to a normal distribution (see (33)). Further in the text, we use the symbol lg to denote the log-10 logarithm. Let $X = \lg(\bar{Y}) = \lg(e)\ln(\bar{Y})$. Following Eqs. 1-5 in (33), it is possible to estimate μ_X and σ_X^2 from estimates $\hat{\mu}_{\bar{Y}}$, $\hat{\sigma}^2_{e,\bar{Y}}$ of $\mu_{\bar{Y}}$, $\sigma^2_{e,\bar{Y}}$ as follows:

$$\hat{\sigma}^2_Z = \ln\left(1 + \frac{\hat{\sigma}^2_{e,\bar{Y}}}{\hat{\mu}^2_{\bar{Y}}}\right) \quad [S9]$$

$$\hat{\sigma}^2_X = \left[\lg(e)\right]^2 \hat{\sigma}^2_Z \quad [S10]$$

$$\hat{\mu}_X = \lg(e)\ln(\hat{\mu}_{\bar{Y}}) - \overbrace{\lg(e)\frac{1}{2}\hat{\sigma}^2_Z}^{\approx 0} \quad [S11]$$

Using Eqs. S9 and S10 above, we estimated the standard errors for brain- and body-mass on the log-10 scale. The resulting estimates are shown on Fig. S1. Noticing that the term $-\lg(e)\frac{1}{2}\hat{\sigma}^2_Z$ in Eq. S11 was very small and nearly constant among all species, we neglected this term. In this way, $\hat{\mu}_X$ were exactly equal to the log-10 transformed trait values. These values were identical with the trait values used in (14). These values are stored as a matrix called "values" in the file MGPMMammals/data/values.rda. The corresponding standard errors are stored as a matrix called SEs in the file MGPMMammals/data/SEs.rda. Each of these matrices has two rows corresponding to the lg-body- and lg-brain-mass, respectively. The order of columns in these matrices corresponds to the order of tips in the tree.

H.3. Model inference.

MGPM Inference. First, we performed a total of seven MGPM fits on the mammal data excluding measurement error, setting different random generator seeds. Then, we selected the best of these 7 fits and performed a manual round-robin step on it. The search history and final score for this fit are depicted on Figs. S3-S4. This model object is stored in the binary file MGPMMammals/data/bestModelToDataWithoutSEs.rda. The program code for running this inference is located in the file MGPMMammals/data-raw/DetectShiftsMammalData_MixedGaussian.R.

In compliance with a reviewer's suggestion to account for measurement error, we reran two times the fit on the mammal data, including the estimated standard error for each trait/species. The program code for running this MGPM inference is located in the file MGPMMammals/data-raw/DetectShifts_MGPM_A_F_best_clade_2.R. In addition, we ran a ML fit of the best MGPM inferred from the first session (without measurement error) on the mammal data with measurement error. The program code for running this ML fit is located in the file MGPMMammals/data-raw/FitFinalModel_t6_toMammalDataWithSEs.R. In this ML fit, the shift-point configuration and model mapping was kept fixed as inferred from the mammal data without measurement error. This ML fit resulted in a model with a slightly better (lower) AIC ($\Delta AIC \ -2$) and minimal difference in the shift-point configuration, compared to the best model from the two other runs on the mammal data with measurement error. Due to various constraints, e.g. limited parallel capacity on the ETH Euler cluster, we were not able to run more runs on the mammal data with measurement error. Hence, we retained the ML fit as the best model on data with measurement error and this model is referred to as MGPM* in the main text. The model object for MGPM* is stored in the binary file MGPMMammals/data/bestFitToDataWithSEs.rda.

Single-regime model inference. The inference of the single-regime models A-F on the mammal data with measurement errors was performed during Step 1 of the RCP algorithm (see Algorithm S1). The corresponding model objects can be retrieved from the binary data file MGPMMammals/data/fitMappings_MGPM_A_F_best_clade_2_DataWithSEs.rda.

SURFACE OU, SCALAR OU and RATEMATRIX BM inference. We performed inference of the SURFACE OU (4), the SCALAR OU models (2), and the RATEMATRIX BM (39) models. All three of these fits were implemented as MGPM fits over sets of one candidate model (section C). All other settings of the RCP algorithm were the same as for the MGPM fits over the models A-F. These fits were performed on the mammal data including measurement errors. The scripts for these fits are located in the files MGPMMammals/data-raw/DetectShifts_SURFACEOU_best_clade.R, MGPMMammals/data-raw/DetectShifts_SCALAROU_best_clade.R and MGPMMammals/data-raw/DetectShifts_MGPM_B_best_clade_2.R. The resulting model objects are stored in the binary files MGPMMammals/data/fitMappings_SURFACEOU_best_clade_DataWithSEs.rda, MGPMMammals/data/fitMappings_SCALAROU_best_clade_DataWithSEs.rda, and MGPMMammals/data/fitMappings_MGPM_B_best_clade_2_DataWithSEs.rda.

H.4. Parametric bootstrap of the MGPM model.*

Simulating the bootstrap datasets. We performed model parametric bootstrap based on MGPM*. A total of 200 random trait datasets were generated by simulating MGPM* on the mammal tree (tree with shift-point configuration and model type assignment as depicted on Fig. 1A, main text). Measurement error at the tips of the tree was simulated by adding to the simulated trait values white (Gaussian) noise with mean 0 and standard deviation equal to the standard error for each trait/species. The code for generating these bootstrap datasets is in the file MGPMMammals/data-raw/SimulateParametricBootstrapData.R. The resulting simulated datasets are stored in the binary data file MGPMMammals/data/valuesBootstrapBestFitToDataWithSEs.rda.

MGPM inference on the bootstrap datasets. Due to parallel execution constraints of the Euler cluster, MGPM inference was limited to the first 50 datasets, using heuristic B.1 (instead of B.2) and a round robin final step of up to 5 iterations. Each of these inferences was run using 50 parallel CPUs (compared to 200 CPUs for runs using the B.2 heuristic). 49 out the 50 model inferences finished within 3 days and were retained for the bootstrap analysis. The program code for running the bootstrap fits is located in the file MGPMMammals/data-raw/DetectShifts_bootstrap_MGPM_A_F_best_clade_RR.R. The resulting best MGPMs from the bootstrap datasets are located in the file MGPMMammals/data/fits_bootstrap_MGPM_A_F_best_clade_RR_HD.rda.

The code used to collect the bootstrap fits and generate this data file is located in the file MGPMammals/data-raw/CollectFits_bootstrap_MGPM_A_F_best_clade_RR.R.

Each of the bootstrap MGPM fits generated a shift-point configuration and model type assignment. These are shown on Figs. S5-S7. A summary comparing these trees with the optimal shift-point configuration and model assignment in MGPM* is provided in Fig. S8.

Bootstrap support for the shift-point configuration in MGPM*. We performed a visual comparison of the shift-point configurations inferred from the bootstrap datasets (Figs. S5-S7) with the tree used to simulate the datasets (Fig. 1A, main text). We observed a tendency for the bootstrap MGPMs to have fewer regimes than MGPM* with weakest bootstrap support for the shift between regimes 1 and 2 (Fig. 1A, main text). There was strong bootstrap support for the shift-points defining regimes 3 (Haplorrhini), 4 (Microchiroptera), 5 (Cetartiodactyla), 6 (Soricidae), 7 (HystricognathiG1), 10 (Cercopithecidae) and 11 (MuridaeG1) (numbers from Fig. 1A). Weaker support was observed for regime 2, 8, 9 and 12.

Bootstrap support for the model type assignment in MGPM*. Except for regime 5, for the regimes with strong bootstrap support, the bootstrap MGPMs correctly discriminated between simulated BM and OU types. For regime 5, we observed a frequent assignment of model type B, instead of the simulated model type F (Figs. S5-S7). The exact type of OU model, was rarely identified by the MGPM fit even if the shift-locations were correctly identified. This could be due to the relatively small sizes of the groups associated with these models. A notable exception was regime 1, for which all bootstrap MGPMs assigned model type F, same as MGPM*. This identifiability for the model type in regime 1 could be explained by the fact that this regime contained the biggest group of species both, in MGPM* as well as most of the bootstrap MGPMs. We note also the good overlap of the tips in regime 1 in MGPM* vs the tips in regime 1 in the bootstrap MGPMs (Fig. S8C,D).

Bootstrap boxplots for X_0 and for the parameters in regime 1. We caution that it is generally impossible to compare the parameters of the bootstrap MGPMs against those in MGPM*, because there is not an exact correspondence between these parameters (see also section I). This is possible to do only for the global parameter X_0 which is present in all models, and for the parameters of regimes/model types which were correctly identified by most of the bootstrap MGPMs. This was the case for regime 1.F containing the largest group of species in MGPM* as well as most of the bootstrap MGPMs. Fig. S9 shows the bootstrap box-plots for X_0 and the OU_F parameters in regime 1.

Calculating the expected brain-body-mass regression slope through time in the bootstrap MGPMs. In the middle panel of Fig. 2 in main text, we show the expected evolution for the brain-body-mass regression slope for MGPM* and for each bootstrap MGPM. To that end, we first discretized the time interval since the root of the tree into epochs at each 2 Ma. Then, for each epoch, we inserted singleton nodes on all branches of the tree intersecting with this epoch. In doing this, we preserved the regime assignment (colouring) of the trees, both for MGPM* (Fig. 1A, main text) and for the coloured trees resulting from the parametric bootstrap inferences (Figs. S5-S7). For each epoch we considered each regime in MGPM* that intersects with that epoch separately. For all nodes assigned to such a regime and located at that epoch in time, we calculated the mean vector and the variance covariance matrix using MGPM* and the bootstrap MGPMs (see sections D and E). We note that in the case of MGPM* all nodes at a given epoch in a given regime have the same expected regression slope (the traits have followed the same evolutionary path in terms of model types and model parameters since the root of the tree). For the bootstrap MGPMs though, these expected regression slopes can differ, because a bootstrap MGPM can assign different model types along the path from the root to these nodes. Hence, for the bootstrap MGPMs, we took the average of the expected regression slope over all nodes at the given epoch within the given regime.

H.5. Model fits to phylogenetic principal component scores of the mammal data. A recent work by Adams et al. (40) has evoked the need for phylogenetic comparative methods (PCMs) to be invariant to rigid linear transformations of the trait data, meaning *"summary measures and statistical tests based on them (PCMs) should be invariant to the orientation of the multivariate dataspace, so long as all trait dimensions containing variation are treated simultaneously"* (40). An example is the rigid rotation, in which the design $(N \times k)$-matrix $\mathbf{X}$ with row-vectors containing the trait values for each tip in the tree is projected onto a different coordinate system. Principal component analysis (PCA) is a prominent practical application of such a rigid rotation, optionally, preceded by shifting (e.g. centering) the rows of the design matrix by a constant vector (40–42). The axes of this new coordinate system are usually called "principal component scores" or "PC scores".

Before we go on with this report, we should mention several concerns arising with the PCA transformation of the data prior to applying a phylogenetic comparative method:

1. The standard PCA ignores the phylogenetic correlation between species (i.e. their common history). It assumes that all trait measurements associated with different tips are statistically uncorrelated. As shown in (41, 42), applying standard PCA with this false assumption can be *"positively misleading"* for the further PCM analysis.

2. It is possible to perform a phylogenetically aware PCA by taking into account a correlation matrix of the data derived from applying a model of evolution on the phylogeny (41, 42). However, as demonstrated in (42), this procedure is not robust with respect to the choice of the model. *"Researchers must assume a model for the evolution of the traits in order to obtain the pPC scores and then perform model-based inference on these scores. This introduces some circularity into the analysis: it seems likely that the choice of a model for the evolution of the traits will influence the apparent macroevolutionary dynamics of the resulting pPC scores"* (42).

3. The correlation between PC scores is different from the correlation of the original trait values. Thus, any estimates involving the trait correlation (e.g. the allometry between two traits) should be mapped back to the original trait space. This task could interfere with other steps of the analysis (see point 2. above).

In the following sub-sections, we summarize the definition and most important properties of the PCA transformation and study the invariance of the maximum likelihood and the AIC-based model selection for MGPM models on the mammal dataset.

Principal component analysis. Formally, the $(N \times k)$-matrix $\mathbf{S}$ of PC scores can be obtained using the formula

$$\mathbf{S} = (\mathbf{X} - \vec{1}\vec{a}^{\,T})\mathbf{V} = \mathbf{X}\mathbf{V} - \vec{1}\vec{a}^{\,T}\mathbf{V}, \qquad \text{[S12]}$$

where $\mathbf{V}$ is a $(k \times k)$-orthogonal matrix with determinant equal to 1[§], $\vec{1}$ is a N-dimensional column vector of 1's, and $\vec{a}$ is k-dimensional column vector (see also Eqs. 1-5 in (42)). Further in the text, we will use the notation $\vec{X}_i$ and $\vec{S}_i$ to denote the k-dimensional column vectors corresponding to the i-th rows of the matrices $\mathbf{X}$ and $\mathbf{S}$, i.e. $\vec{S}_i = \left[(\vec{X}_i - \vec{a})^T\mathbf{V}\right]^T = (\vec{X}_i^T\mathbf{V} - \vec{a}^T\mathbf{V})^T$.
Depending on the choice of $\vec{a}$ and $\mathbf{V}$, we distinguish between two types of PCA (42):

- standard PCA (sPCA): $\vec{a}$ is the arithmetic mean of the row-vectors of $\mathbf{X}$; $\mathbf{V}$ is the matrix of eigenvectors of the $k \times k$ empirical variance-covariance matrix of the column vectors of $\mathbf{X}$.

- phylogenetic PCA (pPCA): $\vec{a}$ is the *"phylogenetic mean"* of the row-vectors of $\mathbf{X}$ ((42):

$$\vec{a} = \left[(\vec{1}^T\mathbf{C}^{-1}\vec{1})^{-1}\vec{1}^T\mathbf{C}^{-1}\mathbf{X}\right]^T, \qquad \text{[S13]}$$

where the $(N \times N)$-matrix $\mathbf{C}$ is the matrix representation of the tree, i.e. $\mathbf{C}_{ij}$ equals the distance from the root to the most recent common ancestor of any pair of tips (i, j); $\mathbf{V}$ is the matrix of eigenvectors of the $(k \times k)$-matrix obtained using the formula (see also Eqs. 3 and 4 in (42)):

$$\mathbf{R} = (N-1)^{-1}\left[\mathbf{X} - \vec{1}\vec{a}^T\right]^T\mathbf{C}^{-1}\left[\mathbf{X} - \vec{1}\vec{a}^T\right]. \qquad \text{[S14]}$$

We note that, in choosing the matrix $\mathbf{C}$ above, pPCA makes the assumption that the traits have evolved following a BM process. Furthermore, the vector $\vec{a}$ and the matrix $\mathbf{R}$ (Eqs. S13 and S14) represent the maximum likelihood estimates for the root vector $\vec{X}_0$ and the unit-time variance-covariance matrix $\mathbf{\Sigma}$ of the BM model (see, e.g. (42, 43)).

We finish this summary of the PCA with an important property that we use to transform the measurement error in empirical datasets. Let $\vec{X} = \vec{g} + \vec{e}$ is k-dimensional measured trait vector, where $\vec{g}$ denotes the vector of exact trait values and $\vec{e}$ denotes a k-vector of measurement errors, which assumed to be uncorrelated with $\vec{g}$. Let $\mathrm{Cov}\left[\vec{\cdot}\right]$ denotes the $(k \times k)$ variance covariance matrix of $\vec{\cdot}$ for any $\vec{\cdot} \in \{\vec{g}, \vec{e}, \vec{X}\}$. Then, by Eq. S12 and the properties of variance-covariance matrices, the variance covariance matrix of the k-vector $\vec{S}$ obtained after applying the transformation to $\vec{X}$ is is given by

$$\mathrm{Cov}\left[\vec{S}\right] = \mathbf{V}^T\,\mathrm{Cov}\left[\vec{X}\right]\mathbf{V} = \mathbf{V}^T\left(\mathrm{Cov}\left[\vec{g}\right] + \mathrm{Cov}\left[\vec{e}\right]\right)\mathbf{V} = \mathbf{V}^T\,\mathrm{Cov}\left[\vec{g}\right]\mathbf{V} + \mathbf{V}^T\,\mathrm{Cov}\left[\vec{e}\right]\mathbf{V}. \qquad \text{[S15]}$$

Fitting models with shifts to the pPC scores of the mammal data. Applying the procedure described previously in SI Appendix, Section H.5, we applied the linear transformation Eq. S12 using the phylogenetic mean vector $\vec{a}$ (Eq. S13) and the rotation matrix $\mathbf{V}$ formed from the eigenvectors of the inferred rate parameter matrix $\mathbf{\Sigma}$ of the optimal global BM_B model. To transform the measurement standard errors, we used the right-most term in Eq. S15, namely, for each species i in the mammal tree, the transformed standard error was calculated as the Cholesky factor of the matrix $\mathbf{V}^T diag(\vec{SE}_i) diag(\vec{SE}_i)\mathbf{V}$, where $\vec{SE}_i$ denotes the standard error for species i (see SI Appendix, Section H.2) and $diag(\vec{SE}_i)$ denotes the corresponding diagonal 2×2 matrix. The above procedure was implemented in the R-script: MGPMMammals/data-raw/TransformPPCA_MammalData.R; the transformed dataset is available in MGPMMammals/data/valuesPPCA.rda and the transformed measurement error data is available in MGPMMammals/data/SEsPPCA.rda.
We reran the inference for the Global, SURFACE OU, SCALAR OU, RATEMATRIX BM and MGPM (A-F) models on the pPCA transformed version of the data. The optimal scores found for these model are listed in table S11, the inferred parameter values are listed in tables S12–S21 and, for the models with shifts, the inferred regimes on the original and the pPCA transformed data are shown on Fig. S10.
Analysing these results, we identify the following cases:

- The model types BM_A, OU_C and SURFACE OU are not invariant to rigid rotations of the data. This is caused by the fact that these model types, representing special cases of the BM and OU processes, assume phylogenetic independence between the traits. Thus, such models tend to have higher maximum log-likelihoods upon a pPCA transformation, which, by definition, eliminates the phylogenetic correlation between the traits (assuming a global BM_B process, see also SI Appendix, table S11 and SI Appendix, Section K for a theoretical proof and a simulation-based example). Importantly, this lack of invariance for some of the candidate model types compromises the invariance of the AIC-based model selection for the corresponding models with shifts. In particular, the inference of the model MGPM (A-F) to pPCA transformed

[§] We remind that, by definition, an orthogonal $(k \times k)$ matrix $\mathbf{V}$ satisfies the property $\mathbf{V}\mathbf{V}^T = \mathbf{I}$, where $\mathbf{I}$ is the $k \times k$ identity matrix. Each orthogonal matrix $\mathbf{V}$ has a determinant $\det(\mathbf{V}) \in \{1, -1\}$. It is known that the k eigenvectors of a symmetric real $k \times k$ matrix, are orthogonal. Thus, the matrix of the eigenvectors of any symmetric positive definite covariance matrix is orthogonal.

data tends to select model types BM_A and OU_C, therefore favouring models with more shift points (SI Appendix, Fig. S10E). Therefore, whenever invariance to rigid rotations is required, we recommend limiting the set of candidates to models which do not impose constraints on the trait variance-covariance matrix.

- For the Global model types, which do not restrict the correlation between the traits (i.e. don't assume diagonal rate matrix $\mathbf{\Sigma}$), we observe a nearly exact match of the optimal log-likelihood and AIC score values before and after transformation (SI Appendix, table S11).

- The models with shifts SCALAR OU and RATEMATRIX BM do not restrict the phylogenetic correlation between the traits and should be invariant. For these models, we observe relatively similar but not exactly identical optimal scores and shift-point configurations (SI Appendix, table S11, SI Appendix, Fig. S10 B,C). These differences in the model fits can be explained as follows:

  - the optimal shift-point configuration for a given model can be nearly unidentifiable. In particular, different shift-point configurations can have very similar optimal scores;
  - the ML inference procedure is prone to getting trapped in local optima and depends strongly on the initial parameter values, which are drawn at random. At present, no numerical or analytical procedure exists that is guaranteed to always find the global maximum likelihood for a PCM beyond the (trivial) global BM_B model.
  - the "greedy" nature of the RCP algorithm – for big trees beyond 100 tips, it is computationally hard to perform a full search over all possible shift-point configurations (see SI Appendix, section A).

H.6. Interpretation of the global BM_A, global OU_C and SURFACE OU fits to the mammal data. The SURFACE OU fit to the mammal dataset found no shifts in the evolution of mammal body- and brain-mass. Furthermore, the log-likelihood value for the optimal model fit was as low as -540, i.e. about 700 log-units less than the log-likelihood of MGPM* (Table 1, main text). Noticing that this log-likelihood value nearly matched the values for the models global BM_A and global OU_C, we checked the inferred parameter values for the three models (SI Appendix, tables S1, S3 and S7). These revealed a convergence of the three model fits to the same BM_A model. We conducted a further validation of this result. In particular, we used the R-package diversitree (34) to perform univariate OU-inference for each of the two traits. Given the fact that SURFACE OU and OU_C assume independence of the two traits, performing the univariate OU inference with diversitree is equivalent to inferring a single-regime SURFACE OU model or a global OU_C model. This test confirmed our result: the inferred parameter values for the (univariate) selection strength OU parameter α were very close to 0 ($\alpha < 10e-5$, corresponding to a BM-process); the values for the drift parameter σ^2 nearly matched the diagonal elements of the $\mathbf{\Sigma}$ in the ML fit for the global BM_A model; the optimal log-likelihood values for the two traits summed up to the maximum log-likelihood value for the BM_A model (see program listing from the R-script ValidateLogLikSURFACEOU.R below and SI Appendix, tables S1, S3 and S7). We explain these poor fits of the global BM_A, global OU_C and SURFACE OU by the wrong assumption of trait independence in these three models.

```
my-computer$ R -f ValidateLogLikSURFACEOU.R

R version 3.5.3 (2019-03-11) -- "Great Truth"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library(diversitree)
Loading required package: ape
> library(MGPMMammals)
> library(ape)
>
> # Diversitree does not accept singleton nodes and polytomies.
> treeMammals <- collapse.singles(MGPMMammals::tree)
> treeMammals <- multi2di(treeMammals)
>
> # Create an OU log-likelihood function for each mammal trait
> likOU.lg_BodyMass <- make.ou(
+   tree = treeMammals,
+   states = MGPMMammals::values[1,],
+   states.sd = MGPMMammals::SEs[1,])
> likOU.lg_BodyMass
Ornstein-Uhlenbeck likelihood function:
  * Parameter vector takes 2 elements:
     - s2, alpha
  * Function takes arguments (with defaults)
     - pars: Parameter vector
     - root [ROOT.MAX]: Type of root treatment
     - root.x [NULL]
     - intermediates [FALSE]: Also return intermediate values?
  * Phylogeny with 629 tips and 628 nodes
     - Taxa: Tachyglossus_aculeatus, Zaglossus_bruijni, ...
  * Reference:
     - I really don't know
R definition:
function (pars, root = ROOT.MAX, root.x = NULL, intermediates = FALSE)
>
> likOU.lg_BrainMass <- make.ou(
+   tree = treeMammals,
+   states = MGPMMammals::values[2,],
+   states.sd = MGPMMammals::SEs[2,])
> likOU.lg_BrainMass
Ornstein-Uhlenbeck likelihood function:
  * Parameter vector takes 2 elements:
     - s2, alpha
  * Function takes arguments (with defaults)
     - pars: Parameter vector
     - root [ROOT.MAX]: Type of root treatment
     - root.x [NULL]
     - intermediates [FALSE]: Also return intermediate values?
  * Phylogeny with 629 tips and 628 nodes
     - Taxa: Tachyglossus_aculeatus, Zaglossus_bruijni, ...
  * Reference:
     - I really don't know
R definition:
function (pars, root = ROOT.MAX, root.x = NULL, intermediates = FALSE)
>
> # Fit ML-fit of the log-likelihood function.
> # x.init specifies initial values for the parameters alpha and sigma
> # Trying with different x.init values can lead to different estimates
> # due to non-convex likelihood function shape, i.e. local optima.
> fitOU.lg_BodyMass <- find.mle(likOU.lg_BodyMass, x.init = c(0.1, 0.01))
> fitOU.lg_BrainMass <- find.mle(likOU.lg_BrainMass, x.init = c(0.1, 0.01))
>
> # Inferred parameters (sigma^2):
> fitOU.lg_BodyMass$par[1]
        s2
0.00866394
> fitOU.lg_BrainMass$par[1]
         s2
0.003582764
>
> # Inferred parameters (alpha):
> fitOU.lg_BodyMass$par[2]
       alpha
2.075195e-05
> fitOU.lg_BrainMass$par[2]
       alpha
2.441406e-06
>
> # ML-values:
> fitOU.lg_BodyMass$lnLik
[1] -413.2419
> fitOU.lg_BrainMass$lnLik
[1] -127.6642
>
> # The joint ML-value:
> # We use the fact that log-likelihood of the two traits is the sum of
> # the two log-likelihoods (independent traits). This value equals the
> # log-likelihood of the models A, C, and SURFACEOU to the mammal data:
> fitOU.lg_BodyMass$lnLik + fitOU.lg_BrainMass$lnLik
[1] -540.9061
>
```

To provide further evidence that the assumption of trait independence has been the root cause for the poor fit of the global BM_A, global OU_C and the SURFACE OU models, we performed an additional test in which all of the above models were fit to a phylogenetic principal component analysis (pPCA) transformation of the mammal trait data (see SI Appendix, Section H.5 and (41)). Assuming a global BM_B model, the two phylogenetic principal component scores (pPCA scores) resulting from the pPCA transformation are phylogenetically independent. Thus, after the transformation, the global BM_A model has an equal maximum log-likelihood to the maximum log-likelihood of the global BM_B model (we prove this formally in SI Appendix, Section K); as a supermodel of BM_A, the global OU_C fit to the pPC scores should have a maximum log-likelihood at least as high as the maximum log-likelihood for the global BM_A model; again, being a supermodel of the global OU_C model, the SURFACE OU should fit at least as well as the global OU_C model.

The results from model inference on the pPCA transformed mammal data are listed in SI Appendix, table S11. We notice that, in contrast with table 1 in the main text, table S11 shows matching log-likelihood values for BM_A, and BM_B models. The global OU_C is matching the values for the global OU_A model. The SURFACE OU model has identified 2 model shifts and has outperformed the global BM_B model in terms of ΔAIC. This result confirms that the assumption of trait independence is the root cause for the poor fit of the global BM_A, global OU_C and SURFACE OU models to the original mammal data.

H.7. Interpretation of the global OU_D and the SCALAR OU fit to the mammal data. The common feature of the models OU_D and the SCALAR OU models is that they assume a diagonal selection strength matrix $\mathbf{H}$ (with SCALAR OU assuming all elements on the diagonal being equal). We notice that the inference of the model OU_D has converged to the best global BM_B model (table 1 and SI Appendix, tables S2 and S4). Furthermore, with $\Delta\text{AIC} = 110.97$, the SCALAR OU fit to the mammal data was the third best fit after the RATEMATRIX BM and the MGPM* (Table 1, main text). Looking at the inferred model parameters, though, reveals a zero matrix $\mathbf{H}$, meaning that the SCALAR OU model converged to a BM_B model with shifts, which was suboptimal with respect to the RATEMATRIX BM process due to the added penalty for the additional parameters $\mathbf{H}$ and $\vec{\theta}$ (SI Appendix, table S8). This shows that assuming an OU process with a single selection strength parameter α shared by all traits and all regimes in the tree is not appropriate for the mammal data.

H.8. Interpretation of the RATEMATRIX BM fit to the mammal data. With $\Delta\text{AIC} = 73.40$, the RATEMATRIX BM fit to the mammal data was the second best after the MGPM* (Table 1, main text). There was a remarkable agreement for some of the identified shift points between the two models (compare Fig. 1 A in the main text versus SI Appendix, Fig. S10 C). The large difference in the AIC as well as the much lower maximum log-likelihood confirm a strong statistical support for the MGPM (A-F) model versus a simpler BM_B model with shifts.

I. A simulation based comparison of different phylogenetic models and implementations. To test and compare our implementation of the MGPM against other multivariate phylogenetic models and implementations, we performed a benchmark on two-trait simulated data. These benchmarks are organized in an accompanying R-package called MGPMSimulations.

I.1. Simulated data. We simulated ultrametric and non-ultrametric birth-death trees of four different sizes (N=80, N=159, N=318, and N=638). The trees were generated using calls to the function pbtree from the R-package phytools as follows:

- treeFossil80 <- pbtree(n=48, scale=1, b = 1, d = 0.4) : generated a non-ultrametric tree of size N=80 (the size depends on the random generator seed).
- treeExtant80 <- pbtree(n=80, scale=1, b = 1, d = 0.4, extant.only = TRUE) : generated an ultrametric tree of size N=80.
- treeFossil159 <- pbtree(n=106, scale=1, b = 1, d = 0.4) : generated a non-ultrametric tree of size N=159 (the size depends on the random generator seed).
- treeExtant159 <- pbtree(n=159, scale=1, b = 1, d = 0.4, extant.only = TRUE) : generated an ultrametric tree of size N=159.
- treeFossil318 <- pbtree(n=200, scale=1, b = 1, d = 0.4) : generated a non-ultrametric tree of size N=318 (the size depends on the random generator seed).
- treeExtant318 <- pbtree(n=318, scale=1, b = 1, d = 0.4, extant.only = TRUE) : generated an ultrametric tree of size N=318.
- treeFossil638 <- pbtree(n=374, scale=1, b = 1, d = 0.4) : generated a non-ultrametric tree of size N=638 (the size depends on the random generator seed).
- treeExtant638 <- pbtree(n=638, scale=1, b = 1, d = 0.4, extant.only = TRUE) : generated an ultrametric tree of size N=318.

To match the time-scale of the mammal tree, we rescaled the branch-lengths in the trees so that their total height would be equal to 166.2. This allowed to perform trait simulation and ML-inference on the same scale for the parameters as in the analysis of the mammal data.

For each tree, we assigned two shift-point configurations as follows:

- R=2: 1 shift point;
- R>2: R=3 (2 shift points) for N=80, R=5 (4 shift points) for N=159, R=8 (7 shift points) for N=318 and N=638.

The 16 trees with shift-point configurations generated in this way are shown on Fig. S29.

For each shift-point configuration, we generated 4 random model type mappings drawing random models from the set $\{BM_A, BM_B, OU_C, OU_D, OU_E, OU_F\}$. For each model mapping, we generated eight random MGPM parameter sets denoted by the integers 1, ..., 8. Following an anonymous reviewer's suggestion, these parameter sets were split in two groups, differing by the similarity between the values of the OU long-term optimum $\vec{\theta}$ between different OU regimes. Parameter sets 1, ..., 4 were assigned to the group "DistinctThetaOU" and were drawn from the following uniform distributions:

- $0.05 \leqslant \mathbf{\Sigma}_{u,ii} \leqslant 0.5$, $i \in \{1, 2\}$ for all model types;
- $0.0 \leqslant \mathbf{\Sigma}_{u,12} \leqslant 0.2$ for all model types;
- $0.1 \leqslant \mathbf{H}_{S,ii} \leqslant 4.0$, $i \in \{1, 2\}$ for all OU model types;
- $-4.0 \leqslant \mathbf{H}_{S,12} \leqslant 4.0$ for the OU_E model type (keeping $\mathbf{H}_{S,21} = 0$ to ensure symmetry of the transformed matrix $\mathbf{H}$);
- $-4.0 \leqslant \mathbf{H}_{S,ij} \leqslant 4.0$, $i \neq j \in \{1, 2\}$ for the OU_F model type;
- $3.0 \leqslant \theta_1 \leqslant 6.0$ for all OU model types;
- $2.0 \leqslant \theta_2 \leqslant 4.0$ for all OU model types;

The parameter sets 5, ..., 8 were assigned to the group "SimilarThetaOU" and were drawn from the same uniform distributions as above, except for $\vec{\theta}$ which were drawn from:

- $3.0 \leqslant \theta_1 \leqslant 3.3$ for all OU model types;
- $2.0 \leqslant \theta_2 \leqslant 2.2$ for all OU model types;

Fixing the starting point to $X_0 = (1.0, -1.0)^T$, for each randomly drawn parameter set, we simulated four random datasets, using the function PCMSim from the package PCMBase (12). This resulted in a total of (4 tree sizes) × (2 tree types) × (2 shift-point configurations) × (4 model mappings) × (8 parameter sets) × (4 simulations) = 2048 simulated datasets. Due to various constraints, we tested the models and inference methods only on half of these datasets, choosing at random two out of the four model mappings for each scenario. The R-code generating this data is written in the file "MGPMSimulations/data-raw/GenerateTestData_t5.R". All simulated datasets are stored in the file "MGPMSimulations/data/testData_t5.rda". The row ids for the 1024 datasets, on which model inference was tested are designated by an integer vector stored in "MGPMSimulations/data/testData_t5/testData_t5_fittedIds.rda". Scatter plots for these 1024 datasets are shown in Figs. S30-S61.

I.2. Tested models and inference methods. We tested a total of eleven inference methods for three types of MGPM models as described below:

- **SURFACE FWD AICc q=n.a.**: Here we use the function surfaceForward from the R-package SURFACE (4), which performs forward step-wise AICc algorithm to infer the Surface OU model (Materials and Methods, (4)). This method uses a corrected AICc criterion for model selection as specified in (4). Unlike the RCP algorithm, this method does not implement a threshold on the minimal number of tips visible from a shift-node (thus, q=n.a.). After the inference was done, the fit objects were converted to MGPM fit objects from the PCMFit package, aiming to compare the SURFACE fit against the true model (criteria for this comparison described in the next subsection). During this conversion, we checked that the likelihood and information score values of the original fit matched with those in the converted MGPM fit. Due to the long runtime, we ran this implementation on the trees of size N=80 only. The script for running this implementation is found in "MGPMSimulations/data-raw/DetectShifts_t5_surfaceFwdBwd.R"; the script for converting the fit objects is found in "MGPMSimulations/data-raw/CollectFits_surfaceFwd_t5.R". The optimal model objects are stored in "MGPMSimulations/data/fits_surface_fwd_t5.rda".
- **SURFACE FWD-BWD AICc q=n.a.**: Here we call the function surfaceBackward from the SURFACE R-package (4) on the fit object output from the previous call to surfaceForward. This results in a fit combining forward and backward step-wise AICc algorithm to infer the SURFACE model using the AICc as a model selection criterion (4). Unlike the RCP algorithm, the backward step-wise AICc algorithm allows to collapse some of the regimes found in the forward step into a polyphyletic group of branches sharing the same regime (color) on the tree. Biologically, polyphyletic regimes could be associated with occurrences of convergent evolution for separate clades in the tree. Unlike the RCP algorithm, this method does not implement a threshold on the minimal number of tips visible from a shift-node (thus, q=n.a.). We used the same post-processing to convert the resulting fits into PCMFit objects. We ran this implementation on the trees of size N=80 only. The script for running this implementation is found in "MGPMSimulations/data-raw/DetectShifts_t5_surfaceFwdBwd.R"; the script for converting the fit objects is found in "MGPMSimulations/data-raw/CollectFits_surfaceFwdBwd_t5.R". The optimal model objects are stored as a data.table in "MGPMSimulations/data/fits_surface_bwd_t5.rda".

- **SURFACE RCP AICc q=10**: We implemented the SURFACE model as a candidate model type for the MGPM fit and ran the RCP algorithm specifying q=10 and AICc as model selection information criterion. Since for bigger trees, the relatively small setting for q results in big numbers of shift-point configuration, we ran this implementation on the trees of N=80 only. The script for running this implementation is found in "MGPMSimulations/data-raw/DetectShifts_t5_SURFACE_best_clade_2_mcs10.R". The resulting optimal models are found in the data.table "MGPMSimulations/data/fits_SURFACE_best_clade_2_AICc_mcs10_t5.rda".

- **SURFACE RCP AICc q=20**: This is the same SURFACE implementation as above, except for the setting q=20, which enables fast execution on all four tree sizes. The script for running this implementation is found in "MGPMSimulations/data-raw/DetectShifts_t5_SURFACE_best_clade_2.R". The resulting optimal models are stored as a data.table in "MGPMSimulations/data/fits_SURFACE_best_clade_2_AICc_t5.rda".

- **SCALAR OU RCP AIC**: We implemented the SCALAR OU model as a candidate model type for the MGPM fit and ran the RCP algorithm specifying q=20 and the first definition of AIC as model selection criterion (Appendix B). The script for running this implementation is found in "MGPMSimulations/data-raw/DetectShifts_t5_SCALAROU_best_clade_2.R". The resulting optimal models are stored in "MGPMSimulations/data/fits_SCALAROU_best_clade_2_AIC_t5.rda".

- **MGPM A-F RCP B.1 RR AIC q=20**: This is the inference of the MGPM model over the model types A-F using Heuristic B.1 and up to 5 additional round-robin iterations (Step 3, Algorithm S1, Appendix A). The script for running this implementation is found in "MGPMSimulations/data-raw/DetectShifts_t5_MGPM_A_F_best_clade_RR.R". The resulting optimal models are stored in "MGPMSimulations/data/fits_MGPM_A_F_best_clade_RR_AIC_t5.rda".

- **MGPM A-F RCP B.2 AIC q=20**: This is the default MGPM implementation over the models $\{BM_A, ..., OU_F\}$ applying the RCP-algorithm with the Heuristic B.2 (see Appendix A). The script for running this implementation is found in "MGPMSimulations/data-raw/DetectShifts_t5_MGPM_A_F_best_clade_2.R". The resulting optimal models are stored in "MGPMSimulations/data/fits_MGPM_A_F_best_clade_2_AIC_t5.rda".

- **MGPM A-F RCP B.2 RR AIC q=20**: This is the default MGPM implementation (as above) with up to 5 additional round-robin iterations (Step 3, Algorithm S1, Appendix A). The script for running this implementation is found in "MGPMSimulations/data-raw/DetectShifts_t5_MGPM_A_F_best_clade_2_RR.R". The resulting optimal models are stored in "MGPMSimulations/data/fits_MGPM_A_F_best_clade_2_AIC_t5.rda".

- **MGPM A-F FULL AIC q=20**: This is the implementation of a full search over the models A-F limiting the heuristics to Heuristic A.1 and A.2 only. This implementation was run on trees of 80 tips only. The script for running this implementation is found in "MGPMSimulations/data-raw/DetectShifts_t5_MGPM_A_F_all.R". The resulting optimal models are stored in "MGPMSimulations/data/fits_MGPM_A_F_all_AIC_t5.rda".

- **MGPM A-F FULL AIC2 q=20**: This is the same implementation as above, except that it uses AIC2 instead of AIC as optimization score, so that the choice of a different type of model with every shift is not penalized. For this implementation, it was possible to convert the fit objects from **MGPM A-F FULL AIC q=20** above (reusing their log-likelihood values to calculate the AIC2), rather than re-executing the model inference. The script for this conversion is found in 'MGPMSimulations/data-raw/ConvertToAIC2_t5_MGPM_A_F_all.R". The resulting optimal models are stored in "MGPMSimulations/data/fits_MGPM_A_F_all_AIC2_t5.rda".

- **MGPM (A-F) TRUE MLE q=n.a.**: This should not be considered a real inference method, because it does not infer the shift-point configuration and model type assignment. Rather, this is the result from fitting through maximum likelihood the parameters of the MGPM with the true shift-point configuration and model type mapping to the data simulated using the true parameter values of this MGPM. We use this MLE model as a reference when assessing the performance of the other model fits. In particular, this model is expected to be the "closest", yet not identical, to the true MGPM used to simulate the data, because it is informed with the true shift-point configuration and model type assignment. Given that the parameter values of this model are a maximum likelihood estimate (MLE) from a random dataset (simulated with the true parameter values), we expect that these parameter estimates would be "slightly" different from the true values. The likelihood and any information score of this MLE should be equal or higher than the likelihood (or information score) of the true MGPM calculated on the same simulated dataset. A substantial difference between some or all parameters of this model with respect to the corresponding parameter in the true MGPM indicates that these parameters are unidentifiable with respect to likelihood maximization. The script for this ML estimation is 'MGPMSimulations/data-raw/FitTrueModel_t5.R". The resulting optimal models are stored in "MGPMSimulations/data/fits_MGPM_A_F_all_AIC2_t5.rda".

I.3. Execution. Excluding SURFACE FWD AICc q=n.a. and SURFACE FWD-BWD AICc q=n.a., for all model inference methods, we used the same boundaries for the parameters as in the mammal data analysis (see Appendix C). These boundaries are specified in the file "MGPMSimulations/R/PCMParamLimits.R", duplicating the code from the file "MGPMMammals/R/PCMParamLimits.R". We did not find an appropriate way to control these boundaries for SURFACE FWD AICc q=n.a. and SURFACE FWD-BWD AICc q=n.a..

Due to constraints of parallel execution, for the biggest trees (N=638), the inference of MGPM A-F models was done on up to two of the four simulated datsets for each scenario.

All inferences were executed on the ETH Zürich cluster Euler.

I.4. Performance criteria. Comparing an inferred model with shifts to a true model used to simulate the input dataset is a complex task due to the fact that the two models are unlikely to have the same regimes on the tree as well as the same model type assignment to the different regimes. For example, it is inappropriate to compare a parameter $\mathbf{H}$ from a model type OU_C versus a parameter $\mathbf{H}$ from a model OU_F, which mapped to the same part of the tree in an inferred and a true model used to simulate a given dataset, nor, is it appropriate to compare the parameter $\mathbf{\Sigma}$ from a BM_B process versus a parameter $\mathbf{\Sigma}$ from an OU_D process. Due to issues of parameter identifiability and correlation between different parameters such as $\mathbf{\Sigma}$ and $\mathbf{H}$ in OU models, it is fairly possible that two models with differing parameter types define nearly identical expected distributions at the tips of the tree. When this happens, we say that the true model is unidentifiable. Understanding when and with what accuracy a given numerical parameter for a given model type assigned to a given regime in a tree is identifiable is, in our current understanding, an extremely hard task that goes far beyond the scope of this work. Rather than searching for complicated methods to compare the parameters of models with differing sets of parameters, we consider a different approach. In particular we focus on derived features of the models such as the Gaussian distributions for the trait values at the tips of a tree. Following this idea, we define a general set of criteria that can be evaluated for any model with a shift-point configuration for a given tree. Hence, these criteria allow to compare any two Gaussian phylogenetic models with shifts as long as they are applied to the same phylogenetic tree.

- Numeric distances

  1. ΔS: This is the score difference between the inferred model and MGPM (A-F) TRUE MLE q=n.a.. If the score of the best fit is bigger (worse) than the score of the simulated model ($\Delta S > 0$), we know for sure that the optimum of the score surface over multi-parameter plane could not be found during the search. Very likely the fit has been stuck in a local optimum, away from the true model. Conversely, if the found score is nearly equal or smaller (better) than the true model's score, there is a chance that the global optimum has been found. Still, this does not imply that the true model parameters are located in the same valley of the score surface. Note here, that for model inferences based on the AICc or AIC2, these scores have been taken for MGPM (A-F) TRUE MLE q=n.a..
  2. ΔB: This is the Bhattacharyya distance between the $k \times N$-variate normal distribution expected at the tips of the tree under the inferred model and the $k \times N$-variate normal distribution expected under the true model used to simulate the data. The Bhattacharyya distance is defined as follows:

$$\Delta B = \frac{1}{8}(\vec{\mu}_1 - \vec{\mu}_2)^T \mathbf{\Sigma}^{-1}(\vec{\mu}_1 - \vec{\mu}_2) + \frac{1}{2}\ln\left(\frac{\det \mathbf{\Sigma}}{\sqrt{\det \mathbf{\Sigma}_1 \det \mathbf{\Sigma}_2}}\right), \qquad \text{[S16]}$$

     where $\vec{\mu}_i$ and $\mathbf{\Sigma}_i$ are the mean vector and variance covariance matrix for each of the two distributions and $\mathbf{\Sigma} = \frac{\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2}{2}$. If ΔB is close to 0, this means that replacing the true model with the inferred model for simulating the data would be nearly invisible for a side observer – any sample of $k \times N$- trait vectors simulated with the inferred model would be indistinguishable from a sample of such vectors simulated with the true model. We used the function bhattacharyya.dist from the R-package fpc (35) for numerically stable calculation of ΔB.
  3. ΔM: This is the Mahalanobis distance of the mean $(k \times N)$-vector of the $(k \times N)$-variate normal distribution expected at the tips of the tree under the inferred model to the center (mean $(k \times N)$-vector) of the $k \times N$-variate normal distribution expected under the true model used to simulate the data. The Mahalanobis distance is defined as follows:

$$\Delta M = \sqrt{(\vec{\mu}_1 - \vec{\mu})^T \mathbf{\Sigma}^{-1}(\vec{\mu}_1 - \vec{\mu})}, \qquad \text{[S17]}$$

     where $\vec{\mu}_1$ denotes the mean vector expected under the inferred model and $\left\{\vec{\mu}, \mathbf{\Sigma}\right\}$ denote the mean vector and the variance covariance matrix expected under the true model. ΔM can be interpreted as a transformation of the Euclidian distance of $\vec{\mu}_1$ to μ measured in units of multi-dimensional standard deviations of the true normal distribution. In the special case of identity matrix $\mathbf{\Sigma}$, the Mahalanobis distance is equivalent to the Euclidian distance. We used the function mahalanobis from the R-package stats, which returns $(\Delta M)^2$. Then, we report the squared root, that is ΔM.
  4. ΔR: difference in the number of regimes between the inferred and the true model.

- True positive and false positive rates for binary criteria. We defined five binary criteria, each one representing a question with a binary (positive or negative) answer that can be asked either for every pair of nodes in the tree or for every branch in the tree. We compare the answers to these questions given by the inferred model against the known true answers. The true positive rate (tpr, also known as *sensitivity*) is calculated as the proportion of actual positive cases (pairs of nodes or branches, depending on the criterion) that are correctly identified as positive by the inferred model. The false positive rate (fpr, also known as $1 - specificity$) is calculated as the proportion of negative cases that are wrongly identified as positive by the inferred model. The perfect fit to a given data and tree has fpr=0 and tpr=1 for each criterion. The worse fit to a given data and tree has fpr=1 and tpr=0 for each criterion. Equality between the tpr and the fpr for some criterion corresponds to a random guess. The five criteria are listed below:

5. Cluster: for each pair of nodes in the tree (internal and tip nodes), we ask if the branches leading to these nodes evolve under the same regime (i.e. have the same color). The test is positive if the two branches do belong to the same regime and negative otherwise.
6. OU process: for each branch in the tree, we ask whether it evolves under an OU model, i.e. one of the models C, D, E, F. Note that this criterion can not be evaluated for model mappings where none of the mapped model types was among C, D, E, F (all negative, so impossible to calculate tpr) or all of the model types were among C, D, E, F (all positive, so impossible to calculate fpr).
7. Correlated traits: for each branch in the tree, we ask whether the regime assigned to that branch supports correlated traits, that is, the model type mapped to that regime is among B, D, E, F. Similar to criterion 2, this criterion could not be evaluated for model mappings where none of the mapped model types was among B, D, E, F or all of the model types were among B, D, E, F.
8. NonDiagonal H: for each branch in the tree, we ask whether its regime has a non-diagonal matrix $\mathbf{H}$, that is, the model mapped to that regime is among the model types E and F. Similar to criteria 2 and 3, this criterion could not be evaluated for model mappings where none of the mapped model types was among E, F, or all of the model types were among E, F.
9. Asymmetric H: for each branch in the tree, we ask whether its regime has an asymmetric matrix $\mathbf{H}$, that is, the model type mapped to that regime is F. Same considerations as above apply when all or none of the mapped model types are equal to F.

We evaluated the above criteria for each inferred model. The scripts for this evaluation are "MGPMSimulations/data-raw/CalculateDistanceToTrueModelDistributions_t5.R" for the numeric distance criteria and "MGPMSimulations/data-raw/CalculateBinaryCriteria_t5.R" for the binary criteria. The results for each individual simulated dataset and inference method are stored in "MGPMSimulations/data/dtSimulationFits.rda". Aggregated results taking the mean values of an inference method over the group of 4 simulated datasets for each parameter set (panel rows on Figs. S30-S61) are stored in a separate data.table found in "MGPMSimulations/data/dtFig.rda". This aggregated data was used as a data-source for Figs. S11-S28.

I.5. Evaluation. We performed a visual analysis of the performance criteria for each test scenario and inference method (Figs. S11-S28). Also, we used linear regression of the criteria measures on the different inference methods. Below, we comment our most important observations.

The big picture. Considering criteria 1 to 5, which are common for all models and inference methods and test cases, the simulation results suggest a general consensus that the MGPM A-F models have an advantage with respect to the other methods (Figs. S11-S20). This confirms that the two simpler models, SURFACE and SCALAR OU, cannot, in general, fit the patterns exhibited by data simulated under more complex OU models with shifts in all parameters.

Heuristic A.2 can prevent overfitting. It is noteworthy that SURFACE FWD AICc q=n.a and SURFACE FWD-BWD AICc q=n.a (N=80) have a highly negative ΔS, which would suggest an advantage of these two models to the other ones. In fact though, this is a manifestation of model overfitting that was not penalized by the AICc score. Looking at the other criteria shows that SURFACE FWD AICc q=n.a and SURFACE FWD-BWD AICc is dominated by all other methods for all criteria, except for the Mahalanobis distance. This can also be seen from listings summarizing linear models of criteria 1-5 with respect to the type of method. For this purpose, we use the function summary.lm from the R-package stats to summarize the regression of criteria 1-5 on the method used to infer the models. For readers who are not familiar with the output of the summary.lm function, we clarify that the most important section of each listing is the Coefficients table. This table shows the average contrast for each value of the predictor variable "fitType" with respect to the the first row called Intercept. The first row (Intercept) corresponds to the reference method MGPM A-F TRUE MLE AIC q=20. For each estimated contrast, a p-value is calculated measuring the probability of observing such a contrast under the hypothesis that the true contrast is 0. Therefore, a low p-value indicates that the contrast is significantly different from 0. Another important entry in the summary is the value of the statistic called "Adjusted R-squared", hereby abbreviated as R^2_{adj}. This statistic taking values between 0 and 1 measures the proportion of variance in the response variable, that is attributable to variation in the predictor variable. A low value of R^2_{adj} suggests that variables other than the one included in the model might have strong influence on the response.

The listings (see next page) reveal a significant positive bias in ΔR for the methods SURFACE FWD AICc q=n.a and SURFACE FWD-BWD AICc q=n.a. This means that without imposing Heuristic A.2, even a very simple model assuming trait independence and global values for the OU parameters $\mathbf{H}$ and $\mathbf{\Sigma}$ can have a better AICc then the true model used to generate the data. Conversely, imposing Heuristic A.2 with q=10 cancels this bias for the SURFACE model. Therefore, we think that a heuristic like A.2 that forbids the fit of model regimes to single tips in the tree should be mandatory, even for small trees, where the computational complexity is not an issue.

```
> dtSimulationFits[TreeSize == "N=80", summary(lm(deltaScore ~ fitType))]

Call:
lm(formula = deltaScore ~ fitType)

Residuals:
   Min     1Q Median     3Q    Max
-228.8   -8.2    0.0    3.3  366.8

Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                     -1.35e-12   2.79e+00    0.00     1.00
fitTypeMGPM A-F FULL AIC q=20   -3.68e+00   4.83e+00   -0.76     0.45
fitTypeMGPM A-F FULL AIC2 q=20  -1.17e+00   4.83e+00   -0.24     0.81
fitTypeMGPM A-F RCP B.2 RR AIC q=20 -3.58e+00   4.83e+00   -0.74     0.46
fitTypeMGPM A-F RCP B.2 AIC q=20    -1.95e+00   4.83e+00   -0.40     0.69
fitTypeMGPM A-F RCP B.1 RR AIC q=20 -3.34e+00   4.83e+00   -0.69     0.49
fitTypeSCALAR OU RCP AIC q=20    8.43e+01   4.83e+00   17.47  < 2e-16 ***
fitTypeSURFACE RCP AICc q=20     1.66e+02   4.83e+00   34.43  < 2e-16 ***
fitTypeSURFACE RCP AICc q=10     1.59e+02   4.83e+00   33.02  < 2e-16 ***
fitTypeSURFACE FWD-BWD AICc q=n.a.  -4.78e+01   4.83e+00   -9.91  < 2e-16 ***
fitTypeSURFACE FWD AICc q=n.a.   -2.19e+01   4.83e+00   -4.53  6.1e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 63 on 3061 degrees of freedom
Multiple R-squared:  0.532,Adjusted R-squared:  0.53
F-statistic:  348 on 10 and 3061 DF,  p-value: <2e-16

> dtSimulationFits[TreeSize == "N=80", summary(lm(dBhattacharyya ~ fitType))]

Call:
lm(formula = dBhattacharyya ~ fitType)

Residuals:
   Min     1Q Median     3Q    Max
-44.36  -2.44  -0.64   1.73  73.48

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                        2.134      0.661    3.23   0.0013 **
fitTypeMGPM A-F FULL AIC q=20      1.016      1.145    0.89   0.3748
fitTypeMGPM A-F FULL AIC2 q=20     1.037      1.145    0.91   0.3650
fitTypeMGPM A-F RCP B.2 RR AIC q=20    1.147      1.145    1.00   0.3163
fitTypeMGPM A-F RCP B.2 AIC q=20       1.126      1.145    0.98   0.3254
fitTypeMGPM A-F RCP B.1 RR AIC q=20    1.126      1.145    0.98   0.3252
fitTypeSCALAR OU RCP AIC q=20         16.552      1.145   14.46   <2e-16 ***
fitTypeSURFACE RCP AICc q=20          43.550      1.145   38.04   <2e-16 ***
fitTypeSURFACE RCP AICc q=10          42.633      1.145   37.24   <2e-16 ***
fitTypeSURFACE FWD-BWD AICc q=n.a.    43.595      1.145   38.08   <2e-16 ***
fitTypeSURFACE FWD AICc q=n.a.        44.776      1.145   39.11   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15 on 3061 degrees of freedom
Multiple R-squared:  0.636,Adjusted R-squared:  0.635
F-statistic:  536 on 10 and 3061 DF,  p-value: <2e-16

> dtSimulationFits[TreeSize == "N=80", summary(lm(dMahalanobis ~ fitType))]

Call:
lm(formula = dMahalanobis ~ fitType)

Residuals:
   Min     1Q Median     3Q    Max
 -49.8   -1.6   -0.4    0.6  645.7

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                        1.992      1.743    1.14    0.253
fitTypeMGPM A-F FULL AIC q=20      0.231      3.018    0.08    0.939
fitTypeMGPM A-F FULL AIC2 q=20     0.234      3.018    0.08    0.938
fitTypeMGPM A-F RCP B.2 RR AIC q=20    0.389      3.018    0.13    0.897
fitTypeMGPM A-F RCP B.2 AIC q=20       0.520      3.018    0.17    0.863
fitTypeMGPM A-F RCP B.1 RR AIC q=20    0.367      3.018    0.12    0.903
fitTypeSCALAR OU RCP AIC q=20         31.108      3.018   10.31   <2e-16 ***
fitTypeSURFACE RCP AICc q=20          47.987      3.018   15.90   <2e-16 ***
fitTypeSURFACE RCP AICc q=10          44.394      3.018   14.71   <2e-16 ***
fitTypeSURFACE FWD-BWD AICc q=n.a.     6.074      3.018    2.01    0.044 *
fitTypeSURFACE FWD AICc q=n.a.         6.059      3.018    2.01    0.045 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 39 on 3061 degrees of freedom
Multiple R-squared:  0.168,Adjusted R-squared:  0.165
F-statistic: 61.6 on 10 and 3061 DF,  p-value: <2e-16

> dtSimulationFits[TreeSize == "N=80", summary(lm(deltaNumRegimes ~ fitType))]

Call:
lm(formula = deltaNumRegimes ~ fitType)

Residuals:
    Min      1Q  Median      3Q     Max
-12.051  -0.027  -0.012   0.000  15.762

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                     1.16e-13   9.69e-02    0.00     1.00
fitTypeMGPM A-F FULL AIC q=20   1.95e-02   1.68e-01    0.12     0.91
fitTypeMGPM A-F FULL AIC2 q=20  2.73e-02   1.68e-01    0.16     0.87
fitTypeMGPM A-F RCP B.2 RR AIC q=20  1.56e-02   1.68e-01    0.09     0.93
fitTypeMGPM A-F RCP B.2 AIC q=20     7.81e-03   1.68e-01    0.05     0.96
fitTypeMGPM A-F RCP B.1 RR AIC q=20  1.17e-02   1.68e-01    0.07     0.94
fitTypeSCALAR OU RCP AIC q=20       -1.48e-01   1.68e-01   -0.88     0.38
fitTypeSURFACE RCP AICc q=20        -8.13e-01   1.68e-01   -4.84  1.3e-06 ***
fitTypeSURFACE RCP AICc q=10        -2.50e-01   1.68e-01   -1.49     0.14
fitTypeSURFACE FWD-BWD AICc q=n.a.   9.24e+00   1.68e-01   55.07  < 2e-16 ***
fitTypeSURFACE FWD AICc q=n.a.       1.41e+01   1.68e-01   83.76  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.2 on 3061 degrees of freedom
Multiple R-squared:  0.809,Adjusted R-squared:  0.808
F-statistic: 1.29e+03 on 10 and 3061 DF,  p-value: <2e-16

> dtSimulationFits[TreeSize == "N=80", summary(lm(perf_Cluster_tpr ~ fitType))]

Call:
lm(formula = perf_Cluster_tpr ~ fitType)

Residuals:
    Min      1Q  Median      3Q     Max
-0.4179  0.0000  0.0130  0.0192  0.3239

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                      1.00000    0.00420  238.21  < 2e-16 ***
fitTypeMGPM A-F FULL AIC q=20   -0.01592    0.00727   -2.19  0.02861 *
fitTypeMGPM A-F FULL AIC2 q=20  -0.01894    0.00727   -2.60  0.00925 **
fitTypeMGPM A-F RCP B.2 RR AIC q=20 -0.01381    0.00727   -1.90  0.05760 .
fitTypeMGPM A-F RCP B.2 AIC q=20    -0.01305    0.00727   -1.79  0.07285 .
fitTypeMGPM A-F RCP B.1 RR AIC q=20 -0.01304    0.00727   -1.79  0.07307 .
fitTypeSCALAR OU RCP AIC q=20       -0.07468    0.00727  -10.27  < 2e-16 ***
fitTypeSURFACE RCP AICc q=20        -0.02492    0.00727   -3.43  0.00062 ***
fitTypeSURFACE RCP AICc q=10        -0.09750    0.00727  -13.41  < 2e-16 ***
fitTypeSURFACE FWD-BWD AICc q=n.a.  -0.34374    0.00727  -47.27  < 2e-16 ***
fitTypeSURFACE FWD AICc q=n.a.      -0.35382    0.00727  -48.66  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.095 on 3061 degrees of freedom
Multiple R-squared:  0.628,Adjusted R-squared:  0.626
F-statistic:  516 on 10 and 3061 DF,  p-value: <2e-16

> dtSimulationFits[TreeSize == "N=80", summary(lm(perf_Cluster_fpr ~ fitType))]

Call:
lm(formula = perf_Cluster_fpr ~ fitType)

Residuals:
    Min      1Q  Median      3Q     Max
-0.4985 -0.0527 -0.0066  0.0000  0.8625

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                     -3.91e-15   8.67e-03    0.00     1.00
fitTypeMGPM A-F FULL AIC q=20    6.64e-03   1.50e-02    0.44     0.66
fitTypeMGPM A-F FULL AIC2 q=20   6.64e-03   1.50e-02    0.44     0.66
fitTypeMGPM A-F RCP B.2 RR AIC q=20  6.20e-03   1.50e-02    0.41     0.68
fitTypeMGPM A-F RCP B.2 AIC q=20     8.15e-03   1.50e-02    0.54     0.59
fitTypeMGPM A-F RCP B.1 RR AIC q=20  6.75e-03   1.50e-02    0.45     0.65
fitTypeSCALAR OU RCP AIC q=20        1.38e-01   1.50e-02    9.16  < 2e-16 ***
fitTypeSURFACE RCP AICc q=20         4.98e-01   1.50e-02   33.19  < 2e-16 ***
fitTypeSURFACE RCP AICc q=10         4.09e-01   1.50e-02   27.25  < 2e-16 ***
fitTypeSURFACE FWD-BWD AICc q=n.a.   1.10e-01   1.50e-02    7.30  3.7e-13 ***
fitTypeSURFACE FWD AICc q=n.a.       9.56e-02   1.50e-02    6.36  2.2e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2 on 3061 degrees of freedom
Multiple R-squared:  0.409,Adjusted R-squared:  0.407
F-statistic:  212 on 10 and 3061 DF,  p-value: <2e-16
```

Performance of the RCP algorithm. Considering the MGPM A-F methods, the simulations show little difference in the performance for RCP versus FULL implementations. Unfortunately we were unable to validate this observation on trees bigger than 80 tips, due to very long execution times for the FULL search. The results for ΔS and the other criteria suggest that the RCP algorithm is performing very well on trees with small number of regimes. For trees with N=318 and N=638 tips, we observed a noteworthy phenomenon. In particular, there was a tendency for big ΔS on specific model type mappings, regardless of the parameter sets for these mappings (see e.g. mapping CCAAEACD for Ultram. trees of N=318, and mapping AFBDAFBE for Ultram. tree of N=638, Figs. S11, S12). To investigate this phenomenon, we consider the scatter plots for the datasets corresponding to these two cases shown respectively on Figs. S49 S57. We notice that in both of these cases the log-likelihood values of the true models used to simulate these datasets tend to be very low (in particular, for AFBDAFBE, we observe log-likelihoods below -1000). Conversely, in the case of the mapping FBEEFDEC for Ultram. tree of N=638, we observe small or negative ΔS and a tendency for high log-likelihood values (see Figs. S11, S12 and Fig. S57). Based on this, we speculate that the success of the RCP algorithm in inferring a model with a nearly optimal score depends on the likelihood surface of the true model – a likelihood surface that is nearly flat over large regions of the $k \times N$ space of trait values, is prone to cause highly suboptimal models. Finally, we compared the different MGPM A-F implementations using a linear regression analysis. This analysis reveals a small but significant advantage for the methods performing the final round-robin (RR) step of the algorithm. In terms of the average effect on ΔS, the best method was MGPM A-F RCP B.2 RR AIC (see listing on the next page).

```
> dtSimulationFits[sapply(fitType, function(.) {
      startsWith(as.character(.), "MGPM")
    }),

    summary(lm(deltaScore ~ fitType))]

Call:
lm(formula = deltaScore ~ fitType)

Residuals:
   Min     1Q Median     3Q    Max
 -86.6   -9.0    0.0    0.0  657.6

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     -8.17e-14   6.19e-01    0.00  1.00000
fitTypeMGPM A-F FULL AIC q=20   -3.68e+00   1.86e+00   -1.98  0.04796 *
fitTypeMGPM A-F FULL AIC2 q=20  -1.17e+00   1.86e+00   -0.63  0.52905
fitTypeMGPM A-F RCP B.2 RR AIC q=20  4.36e+00   1.15e+00    3.78  0.00016 ***
fitTypeMGPM A-F RCP B.2 AIC q=20     8.29e+00   1.15e+00    7.20  7.0e-13 ***
fitTypeMGPM A-F RCP B.1 RR AIC q=20  7.59e+00   1.13e+00    6.74  1.7e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 28 on 5109 degrees of freedom
  (2053 observations deleted due to missingness)
Multiple R-squared:  0.019,Adjusted R-squared:  0.0181
F-statistic: 19.8 on 5 and 5109 DF,  p-value: <2e-16

> dtSimulationFits[sapply(fitType, function(.) {
     startsWith(as.character(.), "MGPM")
   }) & TreeSize == "N=80",

   summary(lm(deltaScore ~ fitType))]

Call:
lm(formula = deltaScore ~ fitType)

Residuals:
   Min     1Q Median     3Q    Max
-19.34  -1.42   0.00   1.68 144.34

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     -3.03e-14   3.32e-01    0.00  1.00000
fitTypeMGPM A-F FULL AIC q=20   -3.68e+00   5.76e-01   -6.39  2.2e-10 ***
fitTypeMGPM A-F FULL AIC2 q=20  -1.17e+00   5.76e-01   -2.03  0.04229 *
fitTypeMGPM A-F RCP B.2 RR AIC q=20 -3.58e+00   5.76e-01   -6.23  6.0e-10 ***
fitTypeMGPM A-F RCP B.2 AIC q=20    -1.95e+00   5.76e-01   -3.39  0.00072 ***
fitTypeMGPM A-F RCP B.1 RR AIC q=20 -3.34e+00   5.76e-01   -5.80  8.0e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.5 on 1786 degrees of freedom
Multiple R-squared:  0.0385,Adjusted R-squared:  0.0358
F-statistic: 14.3 on 5 and 1786 DF,  p-value: 9.72e-14

> dtSimulationFits[sapply(fitType, function(.) {
     startsWith(as.character(.), "MGPM")
     }) & TreeSize == "N=159",

     summary(lm(deltaScore ~ fitType))]

Call:
lm(formula = deltaScore ~ fitType)

Residuals:
   Min     1Q Median     3Q    Max
-45.53  -6.29   0.00   0.00 141.81

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                     2.12e-14   9.41e-01    0.00  1.00000
fitTypeMGPM A-F RCP B.2 RR AIC q=20 1.97e+00   1.63e+00    1.21  0.22692
fitTypeMGPM A-F RCP B.2 AIC q=20    5.48e+00   1.63e+00    3.36  0.00081 ***
fitTypeMGPM A-F RCP B.1 RR AIC q=20 1.80e+00   1.63e+00    1.10  0.27103
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 21 on 1276 degrees of freedom
  (512 observations deleted due to missingness)
Multiple R-squared:  0.00877,Adjusted R-squared:  0.00644
F-statistic: 3.76 on 3 and 1276 DF,  p-value: 0.0105

> dtSimulationFits[sapply(fitType, function(.) {
     startsWith(as.character(.), "MGPM")
  }) & TreeSize == "N=318",

  summary(lm(deltaScore ~ fitType))]

Call:
lm(formula = deltaScore ~ fitType)

Residuals:
   Min     1Q Median     3Q    Max
 -73.4  -18.4    0.0    0.0  226.5

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                     1.18e-13   1.55e+00    0.00        1
fitTypeMGPM A-F RCP B.2 RR AIC q=20 1.30e+01   2.68e+00    4.85  1.4e-06 ***
fitTypeMGPM A-F RCP B.2 AIC q=20    1.90e+01   2.68e+00    7.10  2.1e-12 ***
fitTypeMGPM A-F RCP B.1 RR AIC q=20 1.42e+01   2.68e+00    5.30  1.4e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 35 on 1276 degrees of freedom
  (512 observations deleted due to missingness)
Multiple R-squared:  0.0476,Adjusted R-squared:  0.0453
F-statistic: 21.3 on 3 and 1276 DF,  p-value: 1.96e-13

> dtSimulationFits[sapply(fitType, function(.) {
     startsWith(as.character(.), "MGPM")
  }) & TreeSize == "N=638",
  summary(lm(deltaScore ~ fitType))]

Call:
lm(formula = deltaScore ~ fitType)

Residuals:
   Min     1Q Median     3Q    Max
-107.1   -7.3    0.0    0.0  636.4

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     -2.93e-14   2.00e+00    0.00   1.0000
fitTypeMGPM A-F RCP B.2 RR AIC q=20  1.11e+01   6.01e+00    1.85   0.0640 .
fitTypeMGPM A-F RCP B.2 AIC q=20     1.74e+01   5.97e+00    2.91   0.0037 **
fitTypeMGPM A-F RCP B.1 RR AIC q=20  2.88e+01   4.57e+00    6.30    5e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 45 on 759 degrees of freedom
  (1029 observations deleted due to missingness)
Multiple R-squared:  0.0551,Adjusted R-squared:  0.0514
F-statistic: 14.8 on 3 and 759 DF,  p-value: 2.37e-09
```

For the MGPM A-F inference methods, there is a positive correlation between criteria 1-7 In particular, when ΔS is small, we observe very good performance with respect to all criteria 2-7 (Figs. S11-S24). We did not notice any difference between the parameter sets 1-4 and 5-8, except for the "Correlated traits" criterion, for which the performance was notably worse in the case of parameter sets 5-8. One potential bias that was observable was that the inferred models tended to have 1 or 2 regimes more than the true model. This was reflected also by tpr < 0 for the Cluster criterion. These results suggest that the MGPM A-F RCP B.2 RR AIC q=10 inference method can reliably identify the clusters in a tree up to a slight positive bias in the number of shift points. Also, the simulation results suggest that the method reliably discriminates between branches evolving under OU and branches evolving under a BM process, as well as branches in which the two traits evolve in a correlated fashion and branches, in which the two traits evolve independently.

The MGPM A-F inference methods perform poorly with respect to criteria 8 and 9 In particular, for small trees, we observe poor performance for the NonDiagonal H (Figs. S25-S26), and the Asymmetric H criterion (Figs. S27-S28). Because ΔS is small for most of these cases, we speculate that the failure of the method to infer these properties correclty should be due to very weak signal in the data for these criteria. This seems to be the case, also from the fact that we start observing an improvement for some of the big trees (e.g. Fig. S27, N=638, Ultram., DF). We suppose that the presence of successful inference for some specific parameter sets on some specific mappings and trees, indicates that the identifiability for these properties depends strongly on the particular parameter values. This justifies the inclusion of complex OU models in the MGPM.

There was no significant difference between ultrametric and non-ultrametric trees While this could be a particular observation related to birth-death parameters used to generate the trees, we did not notice any significant difference between the performance on ultrametric vs non-ultrametric trees.

Concluding notes We should admit that despite our effort to provide an exhaustive analysis, many other trends and patterns might be hidden in the simulation results. Readers who have particular questions, e.g. about the identifiability of some parameters, are encouraged to try mining in the data.table-objects stored in the MGPMSimulations package. Furthermore, despite our will to cover numerous existing tools, a fair comparison was only possible against two models – the SCALAR OU model and the SURFACE model, and one model implementation – the SURFACE R-package. Below, we briefly comment on several recent tools that deserve equal attention but could not be included in our benchmark:

- In a recent extension of the R-package bayou, Uyeda et al. analysed the allometry between metabolic rate and body-mass for a tree of 600 species spanning the animal kingdom (36). While the Bayesian (reversible jump) inference method in this study is innovative and highly valuable for future work, the inferred univariate OU model allowing for shifts only in the long-term optimum and the regression slope is far from realistic. In particular, this model excludes possible evolution of the body-mass. Instead it treats body-mass as a constant value measured for each extant species. In this way, changes of the body-mass through time are modelled as fluctuations of the regression slope β_P (see Eq. 9, p. 4, Appendix in (36)). With that respect, a MGPM allowing for co-evolution of the two traits is more realistic and easier to interpret.

- Another example is the R-package Rphylopars developed by Goolsby et al. (37). Rphylopars implements a multiple trait generalisation of a 3-point linear time algorithm for likelihood calculation (38). This enables fast inference of OU models with varying evolutionary rates on different parts of the tree, sharing the selection strength matrix. However, for OU models Rphylopars requires that the tree is ultrametric (37). Moreover, extending this approach to support shifts over different types of models (beyond BM) is complicated, because each model type must satisfy a generalized 3-point property and involves the implementation of a complex transformation of the branch lengths in the tree (37, 38).

- A third example is the R-package ratemetrix developed by Caetano et al. (39). This study allowed for Bayesian estimation of different parameters on different parts of the tree, but restricted the model to a multivariate BM process. We inferred this model on the mammal data but did not include it in the simulations.

Finally, our simulation results support the general conclusion that data simulated using a complex MGPM is not amenable to analysis with a simpler model – the ML inference of a simpler model, such as SURFACE, results in a bias for some of the most important parameters, e.g. the number of different regimes. Such biases can be prevented by imposing "regularization" rules, such as heuristic A.2. While this speaks in favour of complex models such as the MGPM, the application of such models is far from straightforward – a caution for possible overfitting in the case of small datasets, as well as parameter identifiability issues should always be in mind.

J. Type I error of the AIC-based MGPM selection for varying number of traits. As mentioned in the main text, optimizing an information score function, such as the AIC, is a widely used but often disputed approach for model selection. Previous works have demonstrated the proneness of AIC to selecting a more complex model, in particular, when the number of traits increases (40). In the context of MGPMs, this phenomenon, further referred to as type I error, manifests as selecting a model with more regimes than the true number of regimes and/or selection of a complex OU model when the actual true model is BM. To assess these risks for the designated best MGPM implementation (MGPM A-F RCP B.2 RR AIC q=20) we conducted additional simulations of 2-, 4-, 6- and 8-trait data using a single-regime BM_A or a single-regime BM_B models. In particular, the goal of this test was to answer the following questions:

- What is the risk of our method selecting a multiple-regime model if the true model has been single-regime? We refer to this kind of error as "type I error for regimes".

- What is the risk of incorrectly selecting an OU process when the true process has been BM? We refer to this type of error as “type I error for model type”.
- What is the risk of incorrectly selecting a model with correlating traits when the true model assumes trait independence? We refer to this type of error as “type I error for correlation”.
- Do the above risks aggravate with higher numbers of traits?
- Does adding more data (i.e. increasing the number of tips in the tree) alleviate the above risks?

J.1. Simulated data. We simulated trait data on the eight trees described in Appendix, Section I (Fig. S29). We simulated single regime BM_A and BM_B models of $k \in \{2, 4, 6, 8\}$ traits. For each tree, model type and trait number, we generated four random parameter sets by drawing from uniform distributions specified by the same boundaries as in Appendix, Section I after extrapolation to k traits ($k \in \{2, 4, 6, 8\}$):

- $0.05 \leqslant \mathbf{\Sigma}_{u,ii} \leqslant 0.5$, $i \in \{1, ..., k\}$ for both model types;
- $0.0 \leqslant \mathbf{\Sigma}_{u,ij} \leqslant 0.2$, $i, j \in \{1, ..., k\}, i < j$ for both model types;

For each tree, model type, trait number and parameter set, we simulated two datasets, fixing the initial trait vector at the root to a k-column vector of 1’s and -1’s, i.e. $X_0 = (1, -1, ..., 1, -1)^T$. This resulted in $8 \times 2 \times 4 \times \times 4 \times 2 = 512$ simulations. The script for simulating the above datasets is located in the file MGPMSimulations/data-raw/GenerateTestData_t5_NULL.R. The data-file containing all datasets is located in MGPMSimulations/data/testData_t5_NULL_fittedIds.rda.

J.2. MGPM inference. Due to various constraints, we performed MGPM model inference on a subset of the above simulated datasets. In particular, we limited the tree size to $N = 80$ and $N = 318$ (Fig. S29 A, C, I, G) and the parameters to parameter sets 1 and 3, resulting in $4 \times 2 \times 4 \times \times 2 \times 2 = 128$ MGPM inferences.

The MGPM model inference was performed over the model types BM_A, BM_B, OU_C, OU_D, OU_E and OU_F, adapting the boundaries specified in Appendix, Section C to k traits as follows:

- $0.0 \leqslant \mathbf{\Sigma}_{u,ii} \leqslant 1$, $i \in \{1, , , k\}$ for all model types;
- $0.0 \leqslant \mathbf{\Sigma}_{u,ij} \leqslant 1$, $i, j \in \{1, ..., k\}, i < j$ for all model types;
- $0.0 \leqslant \mathbf{H}_{S,ii} \leqslant 10$, $i \in \{1, ..., k\}$ for all OU model types;
- $-10.0 \leqslant \mathbf{H}_{S,ij} \leqslant 10.0$, $i, j \in \{1, ..., k\}, i < j$, for the OU_E model type (keeping $\mathbf{H}_{S,ji} = 0$ to ensure symmetry of the transformed matrix $\mathbf{H}$);
- $-10.0 \leqslant \mathbf{H}_{S,ij} \leqslant 10.0$, $i \neq j \in \{1, ..., k\}$ for the OU_F model type;
- $\min(\mathbf{X}_{i,\cdot}) \leqslant \theta_i \leqslant \max(\mathbf{X}_{i,\cdot})$, $i \in \{1, ..., k\}$, for each OU model type according to the range of trait i in each simulated dataset $\mathbf{X} \in \mathbb{R}^{k \times N}$.

Using the above settings, we ran the model inference “MGPM A-F RCP B.2 RR AIC q=20” (see also Appendix, Section I for a detailed description). The R-script for running the above inference is located in the file MGPMSimulations/data-raw/DetectShifts_t5_NULL_MGPM_A_F_best_clade_2_RR.R. The raw results from these model inferences are stored in the directories MGPMSimulations/data-raw/Results_t5_NULL_MGPM_A_F_best_clade_2_RR_N80_2 and MGPMSimulations/data-raw/Results_t5_NULL_MGPM_A_F_best_clade_2_RR_N318_2 . Instructions for accessing these directories are available at the package web-page https://github.com/venelin/MGPMSimulations.

J.3. Evaluation. We formulated performance criteria similar to the criteria for the simulations with 2 trait-models (see also Appendix, Section I):

- Criterion 1. ΔS: For each model inference, this is the difference between the AIC score of the model inferred using the procedure "MGPM A-F RCP B.2 RR AIC q=20" and the score of the true model used to simulate the trait values. $\Delta S > 0$ indicates a failure of the “MGPM A-F RCP B.2 RR AIC q=20” procedure to find the global optimum. Note that this criterion is similar to criterion 1 in Appendix, Section I, with the only difference that we evaluate $\Delta S > 0$ with respect to the true parameter values instead of “MGPM (A-F) TRUE MLE q=n.a.”; the latter was not done due to various constraints.
- Criterion 4. ΔR: equivalent to Criterion 4 in Appendix, Section I.
- Criterion 5. Cluster: equivalent to Criterion 5 in Appendix, Section I; Here, we only report the true positive rate (tpr), given that, in the true model, there is only one cluster (single regime), and, therefore, the false positive rate is not defined.
- Criterion 6. OU process: equivalent to Criterion 6 in Appendix, Section I; Here, we only report the false positive rate (fpr), given that, the true model is a single-regime BM, and the true positiver rate (tpr) is not defined.

- Criterion 7. Correlated traits: equivalent to Criterion 7 in Appendix, Section I; Here, we report the true positive rate (tpr) for the simulations where the true model has been BM_B (false positive rate not defined), and the false positive rate (fpr) for the simulations where the true model has been BM_A (true positive rate not defined).

The code calculating the values of the above criteria is found in MGPMSimulations/data-raw/CalculateBinaryCriteria_t5_NULL.R. The values for the above criteria for each inferred model are stored in a data.table object in the file MGPMSimulations/data/fits_MGPM_A_F_best_clade_2_RR_t5_NULL.rda and shown visually on Fig. S62.

Type I error for regimes The inferred models had more than one regime in 19 out of 128 cases, respectively, or an overall 15% type I error rate (Fig. S62B,C). The simulations show that this kind of error tends to occur more frequently in the case k=2, regardless of the tree size, and in the case k=8, N=318. Considering the ΔS values for these two cases (Fig. S62A), we notice that for k=8, the error is associated with failures to identify the global optimal model (see, e.g. S62A,B, N=318, k=8, Parameter:Simulation 1:2, 1:1, 3:1). In the case of k=2, the error is explained by a tendency of overfitting, that is, the added value to the model likelihood from adding new regimes tends to be bigger than the penalty for increased number of parameters (see Eq. S1). In agreement with the simulation of two-trait models in Appendix, Section I, Figs. S17 and S18, this risk seems to aggravate with bigger tree size (Fig. S62B). This motivates the use of more conservative scoring functions when the number of traits is small.

Type I error for model type False positive detection of an OU process was present in 21 out of 128 cases, or an overall 16% type I error rate. Of these 21 cases, there were 13 where the fpr was close or above 0.5 and 11 cases where it was exactly 1, meaning that an OU process has been assigned to each branch in the tree. This latter case was observed in simulations of model BM_B only, in particular for bigger trait numbers, k, and regardless of the tree size. Again, we notice an association of the cases of OU fpr=1 with the cases of $\Delta S > 0$ (see, e.g. S62A,D, N=318, k=8, Parameter:Simulation 3:1, 3:2).

Type I and II error for correlation False positive identification of correlated traits was present in 5 out of 64 cases (simulations of BM_A model only), all from the case k=2 (Fig. S62F). Failure to identify that the traits were actually correlated (type II error for correlation) occurred in 7 out 64 cases, again all belonging to the case of k=2 (Fig. S62E). For bigger number of traits the presence or absence of correlation between the traits was correctly detected. Both, type I and II errors seem to appear less often with bigger tree size (compare N=80 vs N=318 on Fig. S62E,F).

Conclusion The results from our simulations confirm the observation stated in (40) that the rate of type I errors tends to increase with the number of traits k. In addition, our results show that such errors can occur frequently even in the simplest multivariate case k=2. The cause of these errors seem to differ between the cases of small k versus big k, small tree versus big tree, and BM_A- versus BM_B-true model. Hence, as long as the model selection is based on numerical optimization of an information score function, such as AIC, our results suggest differentiating the strategies to minimize the above errors. For instance:

- Regularize (i.e. add additional penalty terms for) the number of regimes in the cases of smaller numbers of traits and bigger tree sizes.

- Consider more exhaustive, analytical or heuristic-based search to minimize the risk of getting stuck in local optima, in the case of numerous traits causing high dimensionality of the search space.

In a broader perspective, our results confirm the conclusion from (40) that the AIC is, in general, not a reliable information score function for phylogenetic comparative models. New probabilistic methods for model selection and inference are needed.

K. On the invariance of PCMs to rigid linear transformations of the trait data. Mathematically, rigid linear transformations are known to preserve the Euclidian distance between the trait vectors associated with any pair of tips in the tree. This implies that, upon a rigid transformation, the pattern of dispersion of the trait vectors in the k-dataspace would be preserved exactly and so would be the shape (and, therefore, the maximum value) of the optimal (Nk)-variate density, representing the likelihood function of a phylogenetic comparative model after ML fit to the data. This "geometrical" reasoning is the argument for claiming that the ML value of any phylogenetic model should be invariant to a rigid transformation of the data (40). A formal proof, though, would require the following two conditions to hold for any pair of a phylogenetic model and a rigid transformation:

1. There exists a point in the parameter space, such that the likelihood of the model with respect to the transformed data equals the ML value of the model with respect to the original data;

2. There does not exist a point in the parameter space, for which the likelihood of the model with respect to the transformed data is higher than the ML value of the model with respect to the original data.

In what follows, we show that the above two conditions do not hold for each model in $\mathcal{G}_{LInv}$ and, therefore ML- and AIC-invariance cannot be claimed, in general, for MGPM models. Briefly, model types imposing constraints on the covariance between the traits, such as BM_A, OU_C and SURFACE OU, are not invariant with respect to such rotations.

In particular, we will show that the BM_B model is indeed ML-invariant but the BM_A model is not. Consider the example of fitting a BM_B model, given a tree of N tips with a $(N \times N)$ phylogenetic variance-covariance matrix $\mathbf{C}$ and a $(N \times k)$ design matrix $\mathbf{X}$. Let the k-vector $\vec{a}$ and the $(k \times k)$ orthogonal matrix $\mathbf{V}$ define a pPCA transformation of $\mathbf{X}$ (see Eqs. S13

and S14), and let $\mathbf{S}$ be the corresponding transformed design matrix of pPC scores (see Eq. S12). We know from (43) that $\vec{a}$ and $\mathbf{R}$ defined as in Eqs. S13 and S14 represent the ML estimate for the root vector $\vec{X}_0$ and $\mathbf{\Sigma}$ of the BM_B model fit to the tree and $\mathbf{X}$. We now apply the same equations to find the ML estimates $\vec{a}'$ and $\mathbf{R}'$ of these model parameters with respect to the transformed data $\mathbf{S}$:

$$\begin{aligned}
\vec{a}' &= \left[(\vec{1}^T\mathbf{C}^{-1}\vec{1})^{-1}\vec{1}^T\mathbf{C}^{-1}\mathbf{S}\right]^T \\
&= \left[(\vec{1}^T\mathbf{C}^{-1}\vec{1})^{-1}\vec{1}^T\mathbf{C}^{-1}(\mathbf{X}-\vec{1}\vec{a}^T)\mathbf{V}\right]^T \\
&= \left[(\vec{1}^T\mathbf{C}^{-1}\vec{1})^{-1}\vec{1}^T\mathbf{C}^{-1}\mathbf{X}\mathbf{V} - (\vec{1}^T\mathbf{C}^{-1}\vec{1})^{-1}(\vec{1}^T\mathbf{C}^{-1}\vec{1})\vec{a}^T\mathbf{V}\right]^T \\
&= (\vec{a}^T\mathbf{V} - \vec{a}^T\mathbf{V})^T \\
&= \vec{0}
\end{aligned} \qquad \text{[S18]}$$

$$\begin{aligned}
\mathbf{R}' &= (N-1)^{-1}\mathbf{S}^T\mathbf{C}^{-1}\mathbf{S} \\
&= (N-1)^{-1}\left[\mathbf{X}\mathbf{V} - \vec{1}\vec{a}^{\,T}\mathbf{V}\right]^T\mathbf{C}^{-1}\left[\mathbf{X}\mathbf{V} - \vec{1}\vec{a}^{\,T}\mathbf{V}\right] \\
&= (N-1)^{-1}\left[\left(\mathbf{X} - \vec{1}\vec{a}^{\,T}\right)\mathbf{V}\right]^T\mathbf{C}^{-1}\left[\left(\mathbf{X} - \vec{1}\vec{a}^{\,T}\right)\mathbf{V}\right] \\
&= \mathbf{V}^T(N-1)^{-1}\left[\mathbf{X} - \vec{1}\vec{a}^{\,T}\right]^T\mathbf{C}^{-1}\left[\mathbf{X} - \vec{1}\vec{a}^{\,T}\right]\mathbf{V} \\
&= \mathbf{V}^T\mathbf{R}\mathbf{V}
\end{aligned} \qquad \text{[S19]}$$

We proceed further to express the maximum log-likelihood values of the model BM_B with respect to $\mathbf{X}$ and $\mathbf{S}$. For that purpose we use the notation $\text{rep}(\vec{\cdot})$ to denote the Nk-dimensional vector obtained by repeating a k-dimensional vector $\vec{\cdot}$ N times, and the notation $\text{vec}(\mathbf{Y})$ to denote the Nk-dimensional vector obtained by stacking the columns of a $k \times N$ times matrix $\mathbf{Y}$ on top of each other. Further, we remind that, in the original trait space, the log-likelihood function corresponding to the optimal BM_B model is a Nk-variate Gaussian density function with a mean vector $\text{rep}(\vec{a})$ and $(Nk \times Nk)$ variance-covariance matrix equal to the Kronecker product $\mathbf{C} \otimes \mathbf{R}$ (43).[¶] The maximum log-likelihood of the BM_B model in the original trait space is given by (43):

$$\begin{aligned}
\ln L &= -0.5\left[\text{vec}(\mathbf{X}^T) - \text{rep}(\vec{a})\right]^T(\mathbf{C}\otimes\mathbf{R})^{-1}\left[\text{vec}(\mathbf{X}^T) - \text{rep}(\vec{a})\right] - 0.5\ln|\mathbf{C}\otimes\mathbf{R}| - 0.5Nk\ln(2\pi) \\
&= -0.5\left[\text{vec}(\mathbf{X}^T) - \text{rep}(\vec{a})\right]^T(\mathbf{C}^{-1}\otimes\mathbf{R}^{-1})\left[\text{vec}(\mathbf{X}^T) - \text{rep}(\vec{a})\right] - 0.5\ln|\mathbf{C}\otimes\mathbf{R}| - 0.5Nk\ln(2\pi) \\
&= -0.5\sum_{i,j\in\{1,\ldots,N\}}\left[\mathbf{C}^{-1}_{ij}(\vec{X}_i - \vec{a})^T\mathbf{R}^{-1}(\vec{X}_j - \vec{a})\right] - 0.5\ln(|\mathbf{C}|^k\,|\mathbf{R}|^N) - 0.5Nk\ln(2\pi)
\end{aligned} \qquad \text{[S20]}$$

In analogy, the maximum log-likelihood of the BM_B model after the pPCA transformation is given by:

$$\begin{aligned}
\ln L' &= -0.5\left[\text{vec}(\mathbf{S}^T)\right]^T(\mathbf{C}\otimes\mathbf{R}')^{-1}\left[\text{vec}(\mathbf{S}^T)\right] - 0.5\ln|\mathbf{C}\otimes\mathbf{R}'| - 0.5Nk\ln(2\pi) \\
&= -0.5\left[\text{vec}(\mathbf{S}^T)\right]^T(\mathbf{C}^{-1}\otimes(\mathbf{V}^T\mathbf{R}\mathbf{V})^{-1})\left[\text{vec}(\mathbf{S}^T)\right] - 0.5\ln\left[|\mathbf{C}|^k\left(\left|\mathbf{V}^T\right|\,|\mathbf{R}|\,|\mathbf{V}|\right)^N\right] - 0.5Nk\ln(2\pi) \\
&= -0.5\sum_{i,j\in\{1,\ldots,N\}}\left[\mathbf{C}^{-1}_{ij}(\vec{S}_i)^T(\mathbf{V}^T\mathbf{R}\mathbf{V})^{-1}(\vec{S}_j)\right] - 0.5\ln(|\mathbf{C}|^k\,|\mathbf{R}|^N) - 0.5Nk\ln(2\pi) \\
&= -0.5\sum_{i,j\in\{1,\ldots,N\}}\left\{\mathbf{C}^{-1}_{ij}\left[(\vec{X}_i^T\mathbf{V} - \vec{a}^T\mathbf{V})^T\right]^T(\mathbf{V}^T\mathbf{R}\mathbf{V})^{-1}\left[(\vec{X}_i^T\mathbf{V} - \vec{a}^T\mathbf{V})^T\right]\right\} - 0.5\ln(|\mathbf{C}|^k\,|\mathbf{R}|^N) - 0.5Nk\ln(2\pi) \\
&= -0.5\sum_{i,j\in\{1,\ldots,N\}}\left\{\mathbf{C}^{-1}_{ij}(\vec{X}_i - \vec{a})^T\mathbf{V}(\mathbf{V}^T\mathbf{R}\mathbf{V})^{-1}\mathbf{V}^T(\vec{X}_i - \vec{a})\right\} - 0.5\ln(|\mathbf{C}|^k\,|\mathbf{R}|^N) - 0.5Nk\ln(2\pi) \\
&= -0.5\sum_{i,j\in\{1,\ldots,N\}}\left\{\mathbf{C}^{-1}_{ij}(\vec{X}_i - \vec{a})^T\mathbf{V}\mathbf{V}^T\mathbf{R}^{-1}\mathbf{V}\mathbf{V}^T(\vec{X}_i - \vec{a})\right\} - 0.5\ln(|\mathbf{C}|^k\,|\mathbf{R}|^N) - 0.5Nk\ln(2\pi) \\
&= -0.5\sum_{i,j\in\{1,\ldots,N\}}\left[\mathbf{C}^{-1}_{ij}(\vec{X}_i - \vec{a})^T\mathbf{R}^{-1}(\vec{X}_j - \vec{a})\right] - 0.5\ln(|\mathbf{C}|^k\,|\mathbf{R}|^N) - 0.5Nk\ln(2\pi) \\
&= \ln L
\end{aligned} \qquad \text{[S21]}$$

By the equality $\ln L = \ln L'$, we have shown that, for the BM_B model and the pPCA transformation, the invariance condition 1. holds. Further, following (43), this is the maximum log-likelihood value, hence, satisfying the invariance condition 2. We presume that the same approach can be used to prove the equality $\ln L = \ln L'$ for any other rigid linear transformation (i.e. other values for the vector $\vec{a}$ and the orthogonal matrix $\mathbf{V}$). However, in these cases, the vector $\vec{a}'$ will be non-zero (Eq. S18), and the mathematical derivation will be more difficult.

Now, we realize that, while the BM_B model is indeed invariant with respect to rigid linear transformations, the same is not true for its nested BM_A model. This becomes clear if we consider the fact that the matrix $\mathbf{R}' = \mathbf{V}^T\mathbf{R}\mathbf{V}$ (Eq. S19) is diagonal with diagonal elements, equal to the eigenvalues of $\mathbf{R}$. This follows from the definition of $\mathbf{V}$ in the pPCA case as the matrix of eigenvectors of $\mathbf{R}$. Therefore, the optimal BM_B model after the transformation is effectively a BM_A model. If we consider the inverse transformation ($\mathbf{X} = (\mathbf{S} - \vec{1}\vec{b}^T)\mathbf{W}$, $\mathbf{W} = \mathbf{V}^T$; $\vec{b} = -(\vec{a}^T\mathbf{V})^T$), the BM_A model would have a lower maximum log-likelihood, as $\mathbf{R}$ is non-diagonal. By this, it becomes clear that the ML estimator for the model BM_A is, in general, not invariant to a rigid linear transformation. Furthermore, it is clear that the AIC criterion for an MGPM fit over

[¶] Note that the original paper (43) uses a slightly different (but equivalent) representation, in which the Nk-dimensional trait vector is obtained by stacking the columns of the $N \times k$ design matrix $\mathbf{X}$ on top of each other. Thus, in the original paper, the covariance matrix is equal to $\mathbf{R} \otimes \mathbf{C}$. For our purpose, though, it is more convenient to use the equivalent representation where the Nk-dimensional trait vector is equal to the stacked columns of the $(k \times N)$ transposed design matrix $\mathbf{X}$. Hence, the variance-covariance matrix is equal to $\mathbf{C} \otimes \mathbf{R}$.

the candidate model-types $\{BM_A, BM_B\}$ would, in general, not be invariant either. We demonstrate this by the following programming example, using one of the simulations of the single-regime BM_B model with $k = 4$ traits (simulation setup described in SI Appendix, Section J).

```
/MGPMSimulations/data-raw vmitov$ R -f ExamplePPCASimulation_BM_B_Id41.R

R version 3.5.3 (2019-03-11) -- "Great Truth"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library(PCMBase)
> library(PCMBaseCpp)
Loading required package: Rcpp
> library(PCMFit)
> library(MGPMSimulations)
> library(ape)
> library(data.table)
>
> # id simulation for
> id <- 41
> testData_t5_NULL[
+   id,
+   list(
+     N = nobs,
+     k = numTraits, p = df,
+     `model-type` = LETTERS[1:6][unlist(mapping)],
+     `logLik(true model)` = unlist(logLik),
+     `AIC(true model)` = unlist(AIC)
+   )]
    N k  p model-type logLik(true model) AIC(true model)
1: 80 4 15          B          -684.5758        1399.152
>
> # phylogeny
> tree <- testData_t5_NULL$treeWithRegimes[[id]]
> # number of tips in tree
> N <- PCMTreeNumTips(tree)
>
> # X: design matrix (each column vector corresponds to one trait)
> X <- t(testData_t5_NULL$X[[id]][, seq_len(N)])
> colnames(X) <- paste0("V", seq_len(ncol(X)))
>
> # True model B used to simulate the data
> trueModelOnOriginalData <- testData_t5_NULL$model[[id]]
>
> # Parameters of the true model:
> # Initial vector at the root:
> round(trueModelOnOriginalData$X0, 2)
[1]  1 -1  1 -1
attr(,"class")
[1] "VectorParameter" "_Global"         "numeric"
attr(,"description")
[1] "trait values at the root"
>
> # Note that the PCMBase R-package uses a slightly different notation,
> # namely Sigma_x stays for the Cholesky factor of the matrix Sigma
> # (rather than Sigma_C).
> # Sigma = Sigma_x %*% t(Sigma_x):
> round(trueModelOnOriginalData$`1`$Sigma_x[,,1] %*%
+         t(trueModelOnOriginalData$`1`$Sigma_x[,,1]), 2)
     [,1] [,2] [,3] [,4]
[1,] 0.23 0.05 0.02 0.01
[2,] 0.05 0.15 0.08 0.02
[3,] 0.02 0.08 0.16 0.05
[4,] 0.01 0.02 0.05 0.17
>
> # number of traits
> k <- PCMNumTraits(trueModelOnOriginalData)
>
> # Best model on the original data found using the RCP algorithm
> bestModelOnOriginalData <-
+   fits_MGPM_A_F_best_clade_2_RR_t5_NULL[IdGlob == id]$model[[1]]
> # Parameters of the best model on the original data:
> # Initial vector at the root:
> round(bestModelOnOriginalData$X0, 2)
[1]  2.88  1.57  3.43 -5.23
attr(,"class")
[1] "VectorParameter" "_Global"         "numeric"
attr(,"description")
[1] "trait values at the root"
> # Sigma:
> round(bestModelOnOriginalData$`1`$Sigma_x[,,1] %*%
+         t(bestModelOnOriginalData$`1`$Sigma_x[,,1]), 2)
     [,1] [,2] [,3] [,4]
[1,] 0.24 0.05 0.03 0.04
[2,] 0.05 0.15 0.10 0.03
[3,] 0.03 0.10 0.20 0.05
[4,] 0.04 0.03 0.05 0.14
>
>
>
> # matrix representation of the tree (using the ape::vcv function)
> C <- vcv(tree)
> # column vector of N 1's
> one <- rep(1, N)
>
> # column vector b
> b <- as.vector(
+   t(solve(t(one) %*% solve(C) %*% one) %*% t(one) %*% solve(C) %*% X))
> round(b, 2)
[1]  2.89  1.57  3.45 -5.26
>
> # the R matrix
> R <- (N-1)^(-1) * t(X - one %*% t(b)) %*% solve(C) %*% (X - one %*% t(b))
> round(R, 2)
     V1   V2   V3   V4
V1 0.24 0.05 0.03 0.04
V2 0.05 0.15 0.10 0.03
V3 0.03 0.10 0.20 0.05
V4 0.04 0.03 0.05 0.15
>
> # the V matrix
> V <- eigen(R)$vectors
>
> # Validating that V is orthogonal with determinant 1:
> round(V %*% t(V), 2)
     [,1] [,2] [,3] [,4]
[1,]    1    0    0    0
[2,]    0    1    0    0
[3,]    0    0    1    0
[4,]    0    0    0    1
> det(V)
[1] 1
>
> # Do the pPCA transformation: obtain the pPC scores matrix
> S <- X%*%V - one%*%(t(b)%*%V)
>
> # 'Rotate' the best model (a model B fit on the original data).
> # We do this by transforming the initial vector X0 and by 'rotating'
> # the unit-time variance covariance matrix parameter.
> rotatedBestModelOnOriginalData <- bestModelOnOriginalData
> rotatedBestModelOnOriginalData$X0[] <-
+   rotatedBestModelOnOriginalData$X0[] %*% V - 1 %*% (t(b) %*% V)
>
> rotatedBestModelOnOriginalData$`1`$Sigma_x[,,1] <-
+   chol(t(V)%*% rotatedBestModelOnOriginalData$`1`$Sigma_x[,,1] %*%
+          t(rotatedBestModelOnOriginalData$`1`$Sigma_x[,,1]) %*% V)
> # Parameters of the rotated best model:
> # Initial vector at the root:
> round(rotatedBestModelOnOriginalData$X0, 2)
[1] 0.01 0.00 0.03 0.02
attr(,"class")
[1] "VectorParameter" "_Global"         "numeric"
attr(,"description")
[1] "trait values at the root"
> # Sigma (should be a diagonal matrix, up to numerical precision error):
> round(rotatedBestModelOnOriginalData$`1`$Sigma_x[,,1] %*%
+         t(rotatedBestModelOnOriginalData$`1`$Sigma_x[,,1]), 2)
     [,1] [,2] [,3] [,4]
[1,] 0.34  0.0 0.00 0.00
[2,] 0.00  0.2 0.00 0.00
[3,] 0.00  0.0 0.12 0.00
[4,] 0.00  0.0 0.00 0.06
>
>
> # Create a model of type A, and set its parameters to those from the
> # rotatedBestModelOnOriginalData.
> modelAFromRotatedBestModelOnOriginalData <- MixedGaussian(
+   k = k,
+   modelTypes = MGPMDefaultModelTypes(),
+   mapping = c(`1`=1),
+   Sigmae_x = structure(
+     0, class = c("MatrixParameter", "_Omitted"),
+     description =
+       "Zero upper triangular factor of the error term"))
>
> modelAFromRotatedBestModelOnOriginalData$X0[] <-
+   rotatedBestModelOnOriginalData$X0
> # note that we only copy the diagonal, the other elements being 0 by the
> # definition of model-type A:
> diag(modelAFromRotatedBestModelOnOriginalData$`1`$Sigma_x[,,1]) <-
+   diag(rotatedBestModelOnOriginalData$`1`$Sigma_x[,,1])
>
> # Calculating the log-likelihood of the different models reveals equal
> # values for the latter three models on both the transformed and the
> # rotated data. This confirms the invariance of the ML estimator for
> # model B. Notice, however that the model A having exactly the same
> # log-likelihood has fewer parameters (p=8, because the matrix Sigma
> # is diagonal). Hence, this model would be selected by a better AIC
> # score.
> #
> # The log-likelihood of the true model on the original data is slightly
> # lower, due to the fact that this model is not fit to the data (sampling
```

```
> # from a normal distribution is usually close to but rarely matching the
> # optimum of the density function).
> report1 <- data.table(
+   model = c("true model (used for the simulation)",
+             "best model B (fit on original data)",
+             "rotated best model B",
+             "model A from rotated best model B"),
+   data = c("original (X)",
+            "original (X)",
+            "pPCA (S)",
+            "pPCA (S)"),
+   p = c(
+     PCMParamCount(trueModelOnOriginalData),
+     PCMParamCount(bestModelOnOriginalData),
+     PCMParamCount(rotatedBestModelOnOriginalData),
+     PCMParamCount(modelAFromRotatedBestModelOnOriginalData)),
+   logLik = c(
+     PCMLik(X = t(X), tree = tree,
+            model = trueModelOnOriginalData),
+     PCMLik(X = t(X), tree = tree,
+            model = bestModelOnOriginalData),
+     PCMLik(X = t(S), tree = tree,
+            model = rotatedBestModelOnOriginalData),
+     PCMLik(X = t(S), tree = tree,
+            model = modelAFromRotatedBestModelOnOriginalData)
+   ))
>
> report1[, AIC:=-2*logLik + 2*p]
>
> report1
                                  model         data  p    logLik      AIC
1: true model (used for the simulation) original (X) 14 -684.5758 1397.152
2:  best model B (fit on original data) original (X) 14 -677.9245 1383.849
3:                 rotated best model B     pPCA (S) 14 -677.9245 1383.849
4:    model A from rotated best model B     pPCA (S)  8 -677.9245 1371.849
>
> # Now let's see if a ML fit of the model A and model B to the original
> # and the rotated data would recover the above pattern.
>
> # We create stubs for the models to be fit, which we pass to PCMFit()
> modelAOnOriginalData <- modelAOnRotatedData <- MixedGaussian(
+   k = k,
+   modelTypes = MGPMDefaultModelTypes(),
+   mapping = c(`1`=1),
+   Sigmae_x = structure(
+     0, class = c("MatrixParameter", "_Omitted"),
+     description =
+       "Zero upper triangular factor of the error term"))
>
> modelBOnOriginalData <- modelBOnRotatedData <- MixedGaussian(
+   k = k,
+   modelTypes = MGPMDefaultModelTypes(),
+   mapping = c(`1`=2),
+   Sigmae_x = structure(
+     0, class = c("MatrixParameter", "_Omitted"),
+     description =
+       "Zero upper triangular factor of the error term"))
>
> # Make results reproducible:
> set.seed(1, kind = "Mersenne-Twister", normal.kind = "Inversion")
>
> # Fit model B to the original data
> fitModelBOnOriginalData <- PCMFit(
+   X = t(X), model = modelBOnOriginalData, tree = tree,
+   metaI = PCMInfoCpp,
+   numGuessInitVecParams = 50000, numCallsOptim = 50)
There were 50 or more warnings (use warnings() to see the first 50)
>
> # Log-likelihood of the inferred model:
> cat(
+   "logLik(fitModelBOnOriginalData$modelOptim) =",
+   round(
+     PCMLik(X = t(X), tree = tree,
+            model = fitModelBOnOriginalData$modelOptim),
+     4), "\n")
logLik(fitModelBOnOriginalData$modelOptim) = -677.9247
>
> # Parameters of the inferred model:
> # Initial vector at the root:
> round(fitModelBOnOriginalData$modelOptim$X0, 2)
[1]  2.96  1.62  3.47 -5.23
attr(,"class")
[1] "VectorParameter" "_Global"         "numeric"
attr(,"description")
[1] "trait values at the root"
> # Sigma:
> round(fitModelBOnOriginalData$modelOptim$`1`$Sigma_x[,,1] %*%
+         t(fitModelBOnOriginalData$modelOptim$`1`$Sigma_x[,,1]), 2)
     [,1] [,2] [,3] [,4]
[1,] 0.24 0.05 0.03 0.04
[2,] 0.05 0.15 0.10 0.03
[3,] 0.03 0.10 0.20 0.05
[4,] 0.04 0.03 0.05 0.14
>
>
> # Fit model A to the original data
> fitModelAOnOriginalData <- PCMFit(
+   X = t(X), model = modelAOnOriginalData, tree = tree,
+   metaI = PCMInfoCpp,
+   numGuessInitVecParams = 50000, numCallsOptim = 50)
There were 50 or more warnings (use warnings() to see the first 50)
>
> # Log-likelihood of the inferred model:
> cat(
+   "logLik(fitModelAOnOriginalData$modelOptim) =",
+   round(
+     PCMLik(X = t(X), tree = tree,
+            model = fitModelAOnOriginalData$modelOptim),
+     4), "\n")
logLik(fitModelAOnOriginalData$modelOptim) = -703.1626
>
> # Parameters of the inferred model:
> # Initial vector at the root:
> round(fitModelAOnOriginalData$modelOptim$X0, 2)
[1]  2.88  1.57  3.45 -5.26
attr(,"class")
[1] "VectorParameter" "_Global"         "numeric"
attr(,"description")
[1] "trait values at the root"
> # Sigma:
> round(fitModelAOnOriginalData$modelOptim$`1`$Sigma_x[,,1] %*%
+         t(fitModelAOnOriginalData$modelOptim$`1`$Sigma_x[,,1]), 2)
     [,1] [,2] [,3] [,4]
[1,] 0.24 0.00  0.0 0.00
[2,] 0.00 0.15  0.0 0.00
[3,] 0.00 0.00  0.2 0.00
[4,] 0.00 0.00  0.0 0.14
>
>
> # Fit model B to the rotated data
> fitModelBOnRotatedData <- PCMFit(
+   X = t(S), model = modelBOnRotatedData, tree = tree,
+   metaI = PCMInfoCpp,
+   numGuessInitVecParams = 50000, numCallsOptim = 50)
There were 50 or more warnings (use warnings() to see the first 50)
>
> # Log-likelihood of the inferred model:
> cat(
+   "logLik(fitModelBOnRotatedData$modelOptim) =",
+   round(
+     PCMLik(X = t(S), tree = tree,
+            model = fitModelBOnRotatedData$modelOptim),
+     4), "\n")
logLik(fitModelBOnRotatedData$modelOptim) = -677.9245
>
> # Parameters of the inferred model:
> # Initial vector at the root:
> round(fitModelBOnRotatedData$modelOptim$X0, 2)
[1] -0.02 -0.02  0.03 -0.01
attr(,"class")
[1] "VectorParameter" "_Global"         "numeric"
attr(,"description")
[1] "trait values at the root"
> # Sigma:
> round(fitModelBOnRotatedData$modelOptim$`1`$Sigma_x[,,1] %*%
+         t(fitModelBOnRotatedData$modelOptim$`1`$Sigma_x[,,1]), 2)
     [,1] [,2] [,3] [,4]
[1,] 0.34  0.0 0.00 0.00
[2,] 0.00  0.2 0.00 0.00
[3,] 0.00  0.0 0.12 0.00
[4,] 0.00  0.0 0.00 0.06
>
>
>
> # Fit model A to the rotated data
> fitModelAOnRotatedData <- PCMFit(
+   X = t(S), model = modelAOnRotatedData, tree = tree,
+   metaI = PCMInfoCpp,
+   numGuessInitVecParams = 50000, numCallsOptim = 50)
There were 50 or more warnings (use warnings() to see the first 50)
>
> # Log-likelihood of the inferred model:
> cat(
+   "logLik(fitModelAOnRotatedData$modelOptim) =",
+   round(
+     PCMLik(X = t(S), tree = tree,
+            model = fitModelAOnRotatedData$modelOptim),
+     4), "\n")
logLik(fitModelAOnRotatedData$modelOptim) = -677.9243
> # Parameters of the inferred model:
> # Initial vector at the root:
> round(fitModelAOnRotatedData$modelOptim$X0, 2)
[1]  0.01 -0.01 -0.01  0.00
attr(,"class")
[1] "VectorParameter" "_Global"         "numeric"
attr(,"description")
[1] "trait values at the root"
> # Sigma:
> round(fitModelAOnRotatedData$modelOptim$`1`$Sigma_x[,,1] %*%
+         t(fitModelAOnRotatedData$modelOptim$`1`$Sigma_x[,,1]), 2)
     [,1] [,2] [,3] [,4]
[1,] 0.34  0.0 0.00 0.00
```

```
[2,] 0.00  0.2 0.00 0.00
[3,] 0.00  0.0 0.12 0.00
[4,] 0.00  0.0 0.00 0.06
>
>
> # Summary:
> report2 <- data.table(
+   model = c("model B",
+             "model A",
+             "model B",
+             "model A"),
+   data = c("original (X)",
+            "original (X)",
+            "pPCA (S)",
+            "pPCA (S)"),
+   p = c(
+     PCMParamCount(fitModelBOnOriginalData$modelOptim),
+     PCMParamCount(fitModelAOnOriginalData$modelOptim),
+     PCMParamCount(fitModelBOnRotatedData$modelOptim),
+     PCMParamCount(fitModelAOnRotatedData$modelOptim)),
+   logLik = c(
+     PCMLik(X = t(X), tree = tree,
+            model = fitModelBOnOriginalData$modelOptim),
+     PCMLik(X = t(X), tree = tree,
+            model = fitModelAOnOriginalData$modelOptim),
+     PCMLik(X = t(S), tree = tree,
+            model = fitModelBOnRotatedData$modelOptim),
+     PCMLik(X = t(S), tree = tree,
+            model = fitModelAOnRotatedData$modelOptim)
+   ))
>
> report2[, AIC:=-2*logLik + 2*p]
>
> report2
      model         data  p    logLik      AIC
1: model B original (X) 14 -677.9247 1383.849
2: model A original (X)  8 -703.1626 1422.325
3: model B     pPCA (S) 14 -677.9245 1383.849
4: model A     pPCA (S)  8 -677.9243 1371.849
>
> # Note: there are small differences in the logLik values for rows
> # 1, 3 and 4 (4th digit after the decimal point). We presume
> # that these are due to the tolerance settings for the convergence
> # criteria of the stats::optim function in R.
>
```

The above programming example confirms that the maximum likelihood of a BM_B model is invariant to a pPCA transformation of the trait data. However, the optimal model fit to the transformed data has a diagonal matrix $\mathbf{\Sigma}'$ and is effectively a model BM_A. Thus, the AIC model selection criterion would select the model BM_B when fit to the original data $\mathbf{X}$ versus the model BM_A when fit to the pPCA scores.

We conclude that $\mathcal{G}_{LInv}$-models, which constrain the phylogenetic covariance between the traits would, in general, have different optimal likelihoods and scores upon rigid linear transformations of the trait data. Thus, for MGPMs, invariance should only be expected when none of the candidate model types imposes such constraints, e.g. an MGPM over $\{BM_B\}$ or $\{OU_F\}$. In practice, though, invariance to rigid transformations will not always be observed, even with such models, due to the fact that, at present, no numerical or analytical procedure exists that is guaranteed to always find the global maximum likelihood.

L. Supplementary Figures.

L.1. Supplementary figures for the mammal data and analysis.

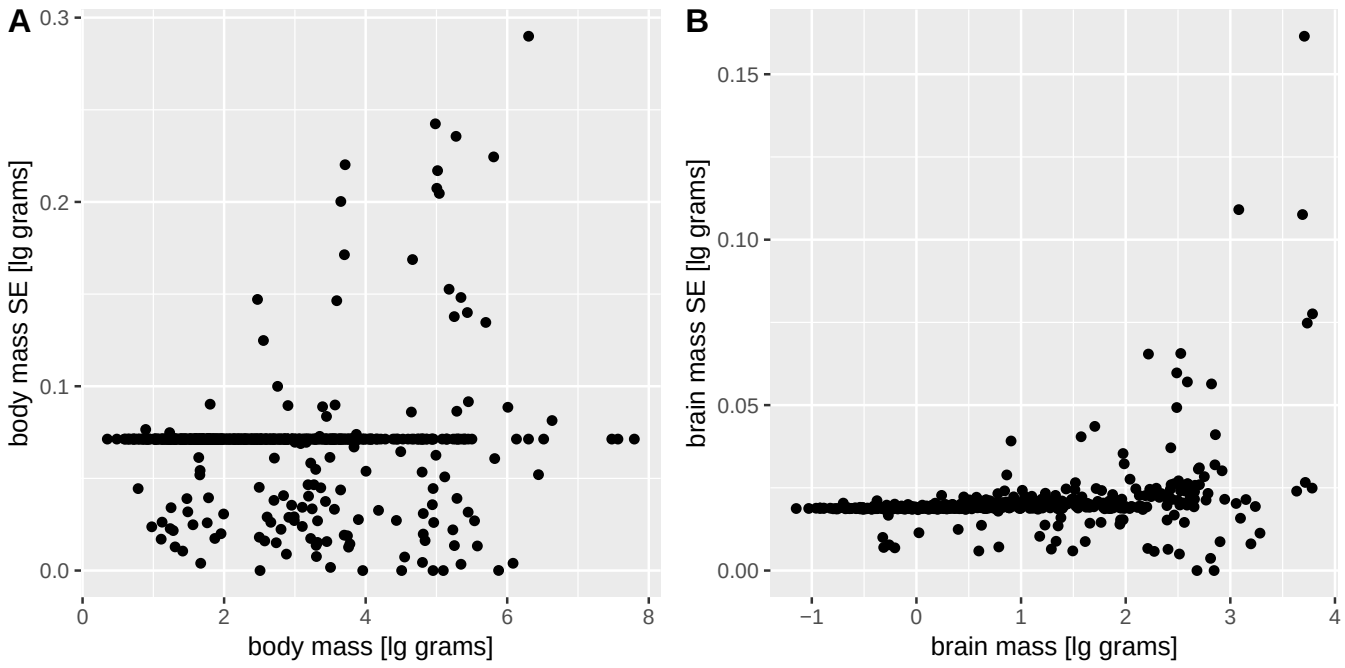


Fig. S1. Estimated standard error for lg-body-mass and lg-brain-mass in mammal species.

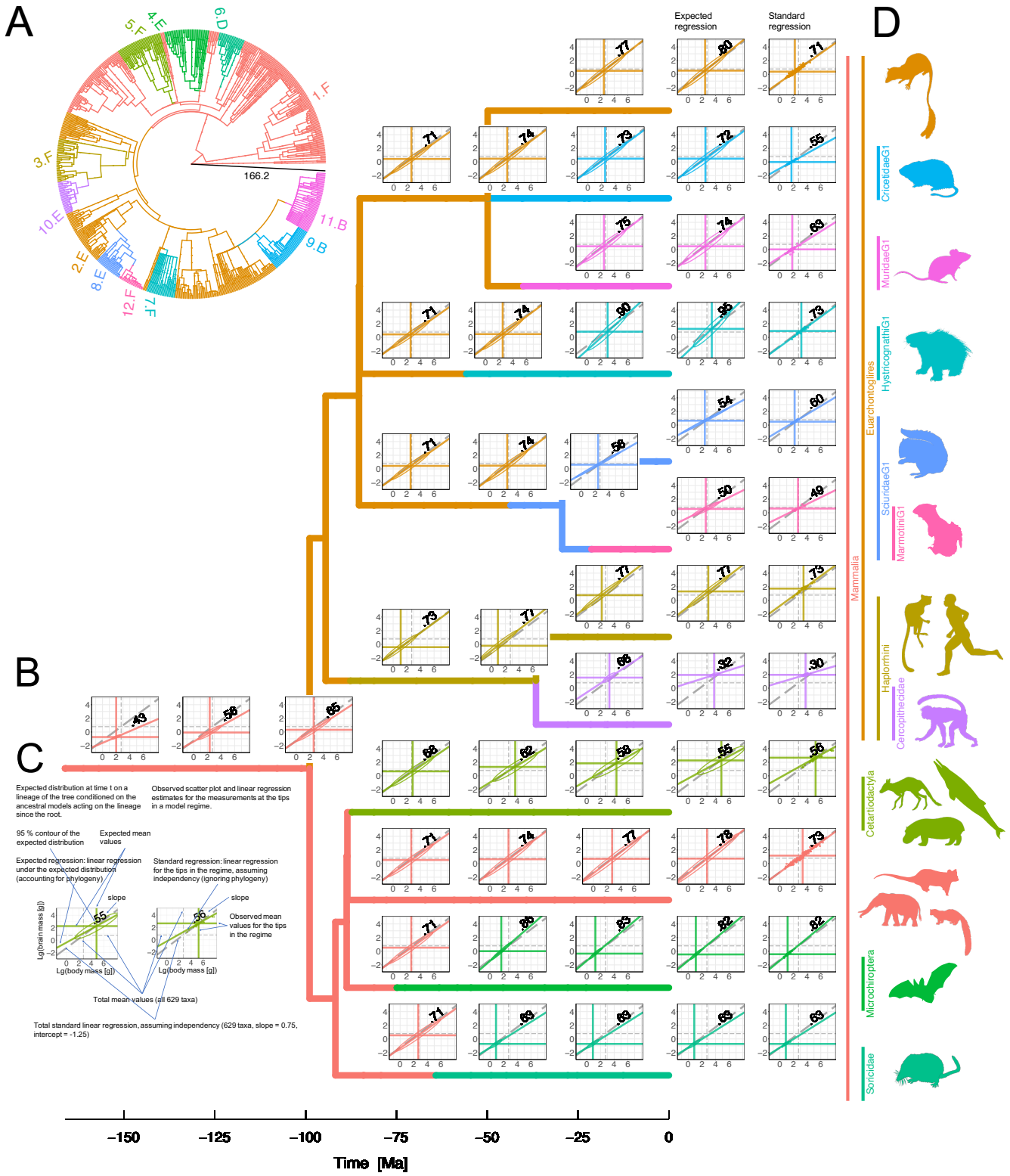
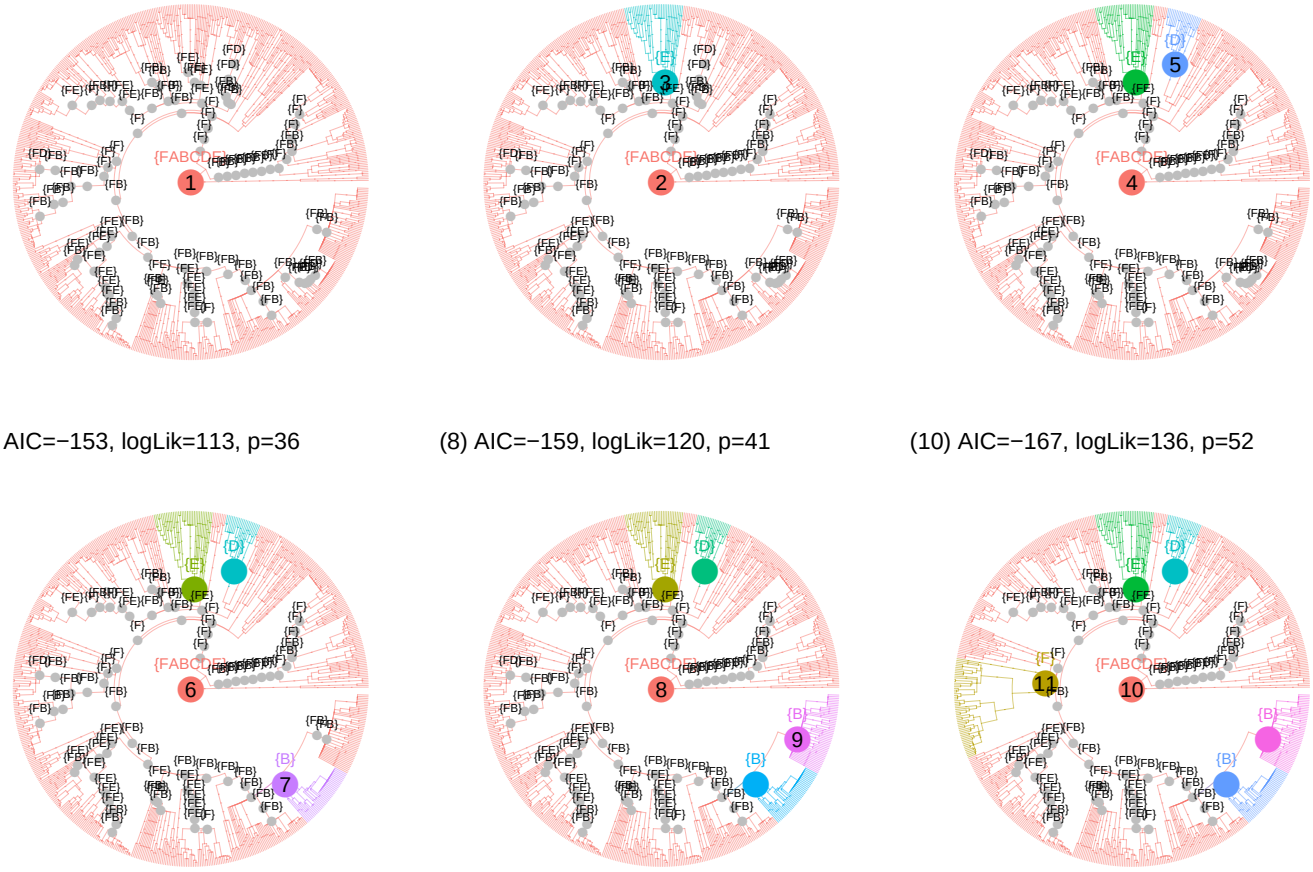


Fig. S2. Evolution of the lg-brain- vs. lg-body-mass regression line in mammals according to MGPM* A: A copy of Fig. 1A. B: A pruned (back-bone) variant of the tree in A. The plots above the lineages depict the evolution of the trait distribution and regression line on each backbone lineage, conditioned on the inferred ML parameters in MGPM*. These inferred distributions should be interpreted as the expectation for the corresponding ancestral species under the hypothesis that MGPM* is the true model. At the tips, this inferred distributions are compared to the empirical trait distributions (scatter plots) localised over the tips belonging to each regime. C: Description for the distribution plots in B. D: A visual hint showing some of the mammal species under each regime. Silhouette images courtesy of Phylopic/T. Michael Keesey, Joseph Wolf, Natasha Vitek, Daniel Jaron, Catherine Yasuda, Allis Markham, Gareth Monger, Jan A. Venter, Herbert H.T. Prins, David A. Balfour, Rob Slotow, C. De Muizon, Scott Hartman, Michael Scroggie, Yan Wong, and Becky Barnes (see also SI Appendix, Section G for full credit details).

(1) AIC=−35, logLik=29, p=12

(2) AIC=−129, logLik=87, p=22

(4) AIC=−149, logLik=105, p=31



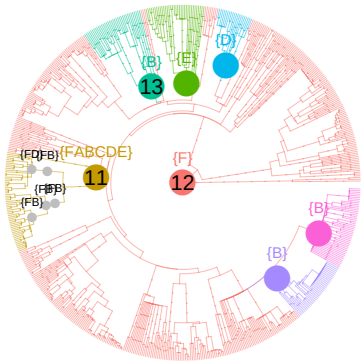
(6) AIC=−153, logLik=113, p=36

(8) AIC=−159, logLik=120, p=41

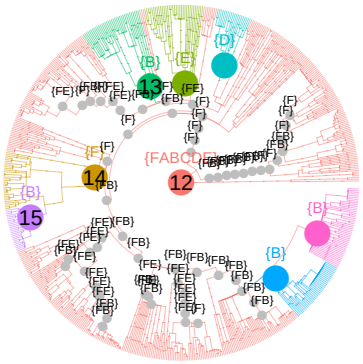
(10) AIC=−167, logLik=136, p=52

Fig. S3. Search path of the recursive clade-partition algorithm in the mammal tree. As initialization step, each model-type is fit to each clade not smaller than a user-defined threshold, q (here, $q = 20$). Each panel denoted by a number in parentheses (i) describes iteration i of the main loop (line 11 in algorithm S1). The coloured node with a number i is the partition root for the iteration. ColoNodes in grey represent the potential shift points - these are descendants from the partition root, which have not been "cut out" by a shift and have at least q descendants, themselves. Letters in braces denote the candidate model-types for each shift-node. For the partition root (i), these are all model-types; for every other node, this is the set $\{XY\}$, where X is the model-type assigned to the node in the best fit on the entire tree found so far, and Y is the best model-type fit to the node's clade during the initial step.

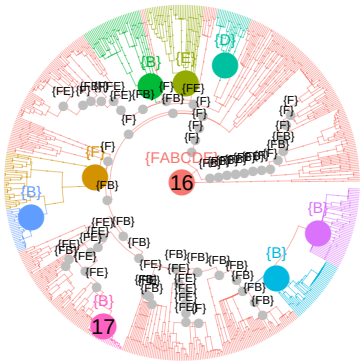
(11) AIC=−188, logLik=151, p=57



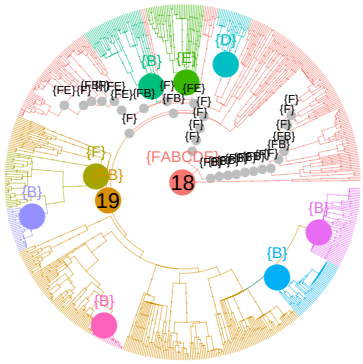
(12) AIC=−189, logLik=156, p=62



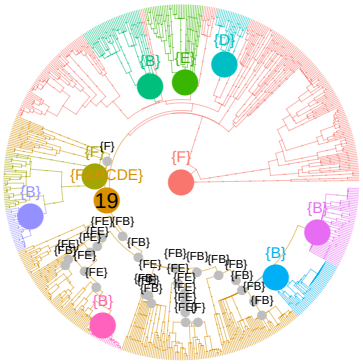
(16) AIC=−202, logLik=168, p=67



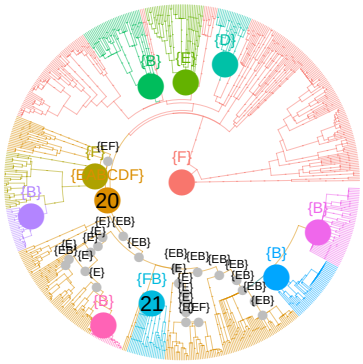
(18) AIC=−222, logLik=189, p=78



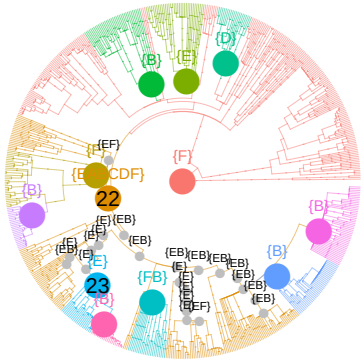
(19) AIC=−222, logLik=189, p=78



(20) AIC=−234, logLik=205, p=88



(22) AIC=−237, logLik=217, p=98



FINAL: AIC=−241, logLik=235, p=115

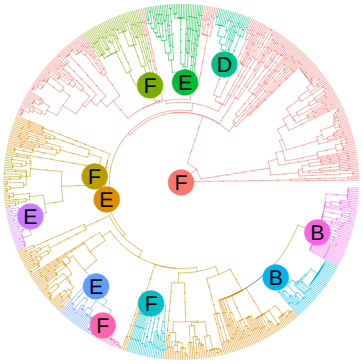


Fig. S4. Search path of the recursive clade-partition algorithm in the mammal tree.

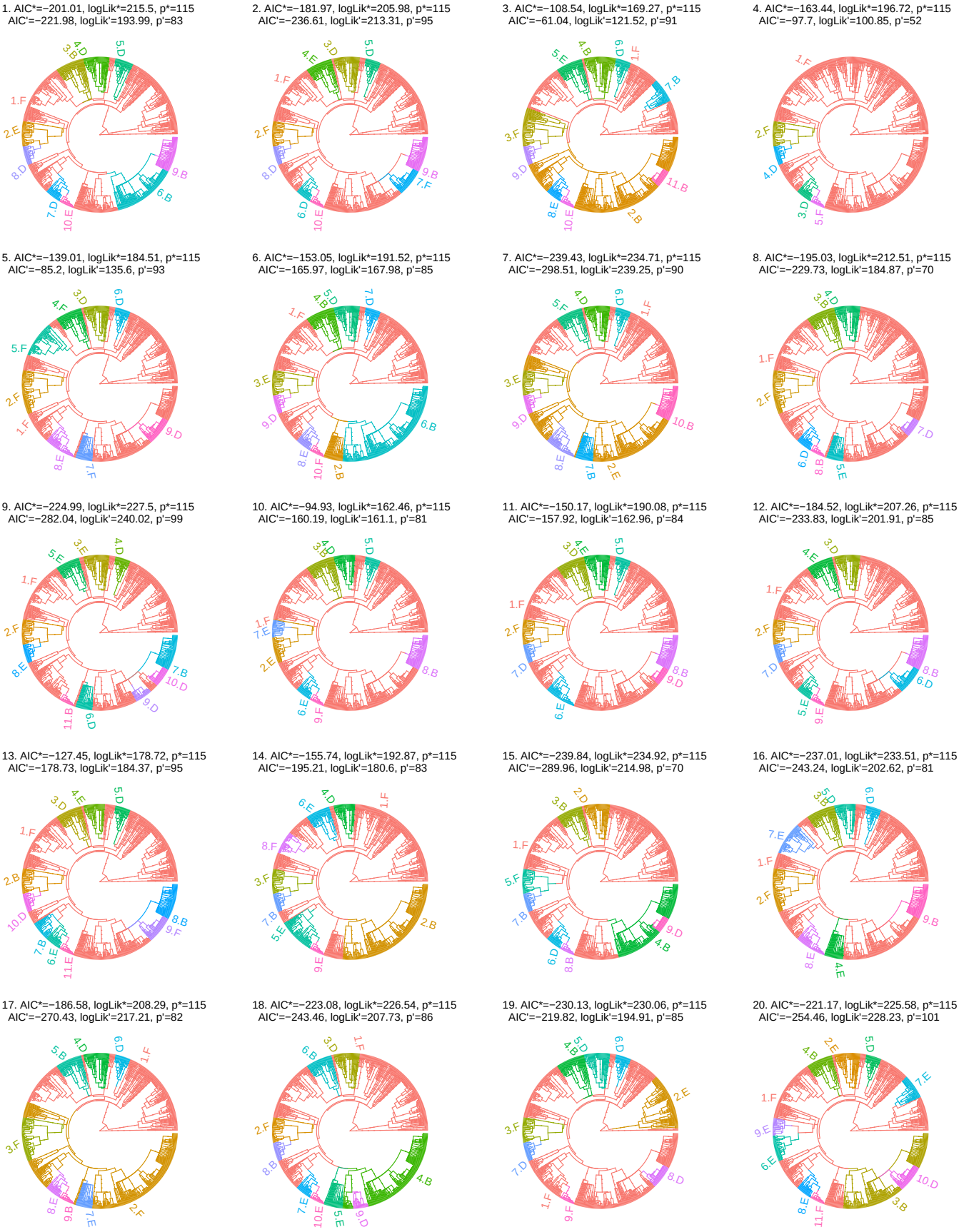


Fig. S5. Parametric bootstrap MGPM models of the mammal tree. The MGPM models were inferred from parametric bootstrap datasets generated by simulating MGPM* on the tree in Fig. 1A. There is no correspondance in color nor in regime number with Fig. 1A. For each tree, the AIC and log-likelihood value from the true model used in the simulation (MGPM*) are compared to the AIC and log-likelihood from the inferred MGPM model. A value of AIC' bigger than AIC* indicates a failure of the search algorithm to find the best fit. This figure depicts the datasets for the trees 1 to 20. For the other trees, see SI Appendix, Figs. S6-S7.

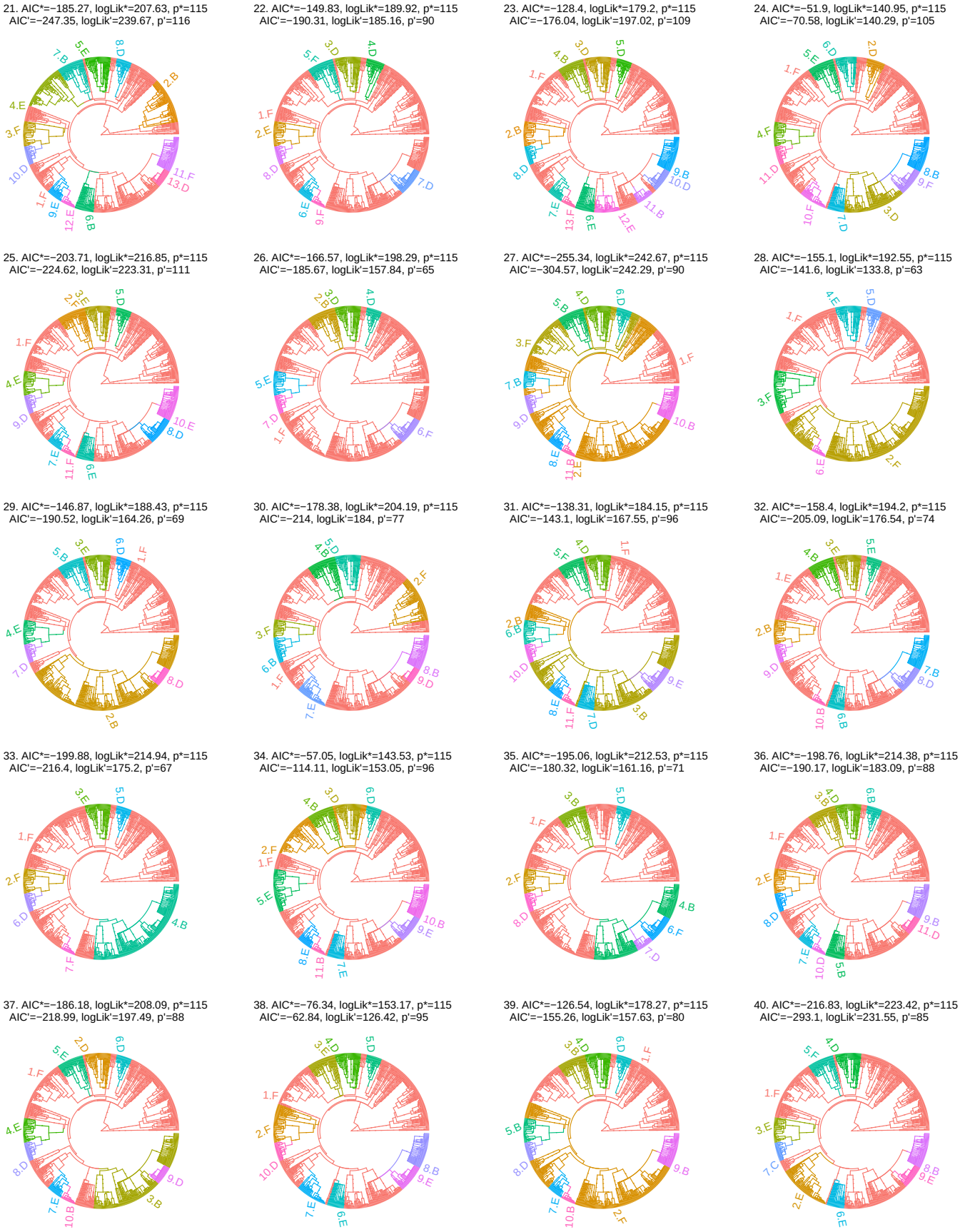
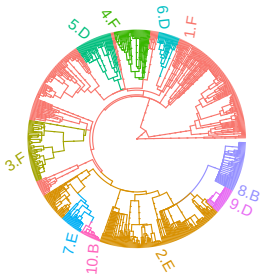
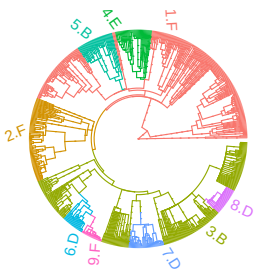


Fig. S6. Parametric bootstrap MGPM models of the mammal tree. The MGPM models were inferred from parametric bootstrap datasets generated by simulating MGPM* on the tree in Fig. 1A. There is no correspondance in color nor in regime number with Fig. 1A. A value of AIC' bigger than AIC* indicates a failure of the search algorithm to find the best fit. This figure depicts the datasets for the trees 21 to 40. For the other trees, see SI Appendix, Figs. S5 and S7.

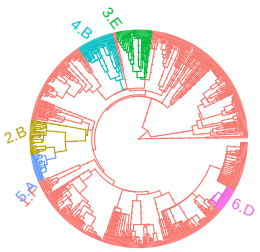
41. AIC*=−201.02, logLik*=215.51, p*=115
AIC'=−253.85, logLik'=217.93, p'=91



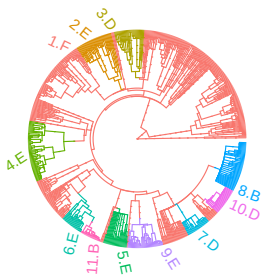
42. AIC*=−154.82, logLik*=192.41, p*=115
AIC'=−200.9, logLik'=181.45, p'=81



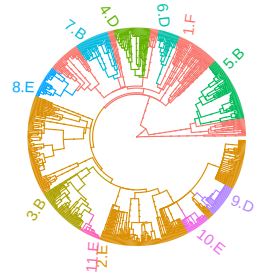
43. AIC*=−262.21, logLik*=246.1, p*=115
AIC'=−283.98, logLik'=186.99, p'=45



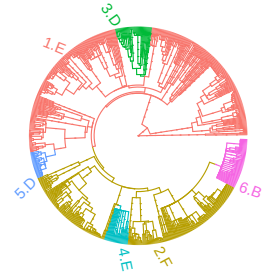
45. AIC*=−266.06, logLik*=248.03, p*=115
AIC'=−305.19, logLik'=251.6, p'=99



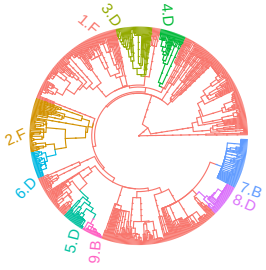
46. AIC*=−243.06, logLik*=236.53, p*=115
AIC'=−234.15, logLik'=211.07, p'=94



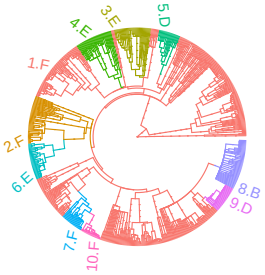
47. AIC*=−161.92, logLik*=195.96, p*=115
AIC'=−199.01, logLik'=154.51, p'=55



48. AIC*=−129.47, logLik*=179.73, p*=115
AIC'=−143.95, logLik'=149.97, p'=78



49. AIC*=−186.39, logLik*=208.2, p*=115
AIC'=−190.13, logLik'=193.07, p'=98



50. AIC*=−239.26, logLik*=234.63, p*=115
AIC'=−233.66, logLik'=191.83, p'=75

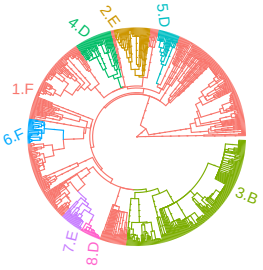


Fig. S7. Parametric bootstrap MGPM models of the mammal tree. The MGPM models were inferred from parametric bootstrap datasets generated by simulating MGPM* on the tree in Fig. 1A. There is no correspondance in color nor in regime number with Fig. 1A. A value of AIC' bigger than AIC* indicates a failure of the search algorithm to find the best fit. This figure depicts the datasets for the trees 41 to 50, excluding one dataset for which the RCP algorithm did not finish in due time. For the other trees, see SI Appendix, Figs. S5 and S6.

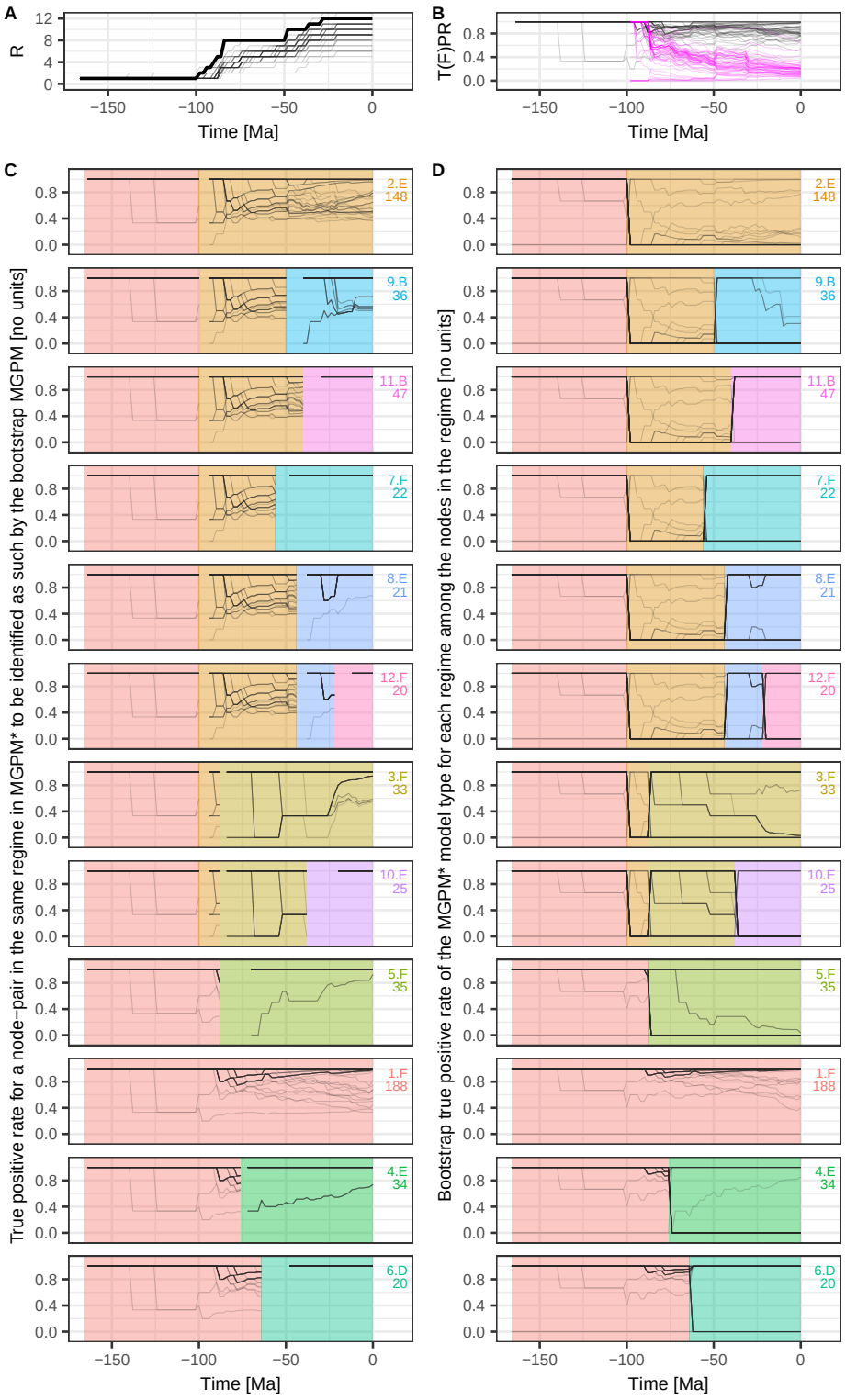


Fig. S8. Summary of the bootstrap support for the shift-point configuration and model type assignment in MGPM*. This figure attempts to provide a visual summary of the bootstrap trees on Figs. S5-S7. To calculate the quantities in the different panels, first, we discretized the time interval from the root of the tree to the present time into epochs at each 2 Ma. Then, for each epoch, we inserted singleton nodes on all branches of the tree intersecting with this epoch. A: Number of regimes (denoted R) in MGPM* (thick black line) and each bootstrap MGPM (think black grey line) through. B: True positive rate (black) vs. False positive rate (magenta) for pairs of nodes being correctly identified by the bootstrap MGPMs as belonging to the same regime in MGPM*. For each pair of nodes at a given epoch, we checked if these two nodes belong to the same regime in MGPM* and the bootstrap MGPM respectively. The true positive rate (TPR) corresponds to the fraction of pairs belonging to the same regime in MGPM* and correctly identified as such by the bootstrap MGPM. The false positive rate (FPR) corresponds to the fraction of the pairs belonging to two different regimes in MGPM*, which were falsely identified as belonging to the same regime by the bootstrap MGPM. In the ideal case, the TPR should equal 1 and the FPR should be equal to 0. C: Evaluation of the true positive rate of a pair of nodes at a given epoch belonging to the same regime in MGPM* to be correctly identified as such in a bootstrap MGPM. This plot differs from the TPR value in panel B in that the TPR is calculated separately for each of the 12 regimes in MGPM*. Note that the lines are interrupted at the beginning of each regime because each regime starts with a single branch, so that there is at most one node (and not a pair) belonging to the regime at that epoch. Note also that a TPR value of 1 is not sufficient for perfect match between the regime in MGPM* and a bootstrap MGPM – a TPR value of 1 guarantees that all nodes in the regime in MGPM* at that epoch belong to a single regime in the bootstrap MGPM, but it is possible that the bootstrap MGPM contains additional nodes at the same epoch (other branches in the tree intersecting with this epoch). D: True positive rate of the model type in each regime in MGPM* calculated over the nodes at a given epoch. A TPR value of 1 means that a bootstrap MGPM has assigned the correct model type to all nodes within the regime in MGPM* at the given epoch. A TPR value smaller than 1 indicates that some of the nodes have been assigned a different model types by the bootstrap MGPM.

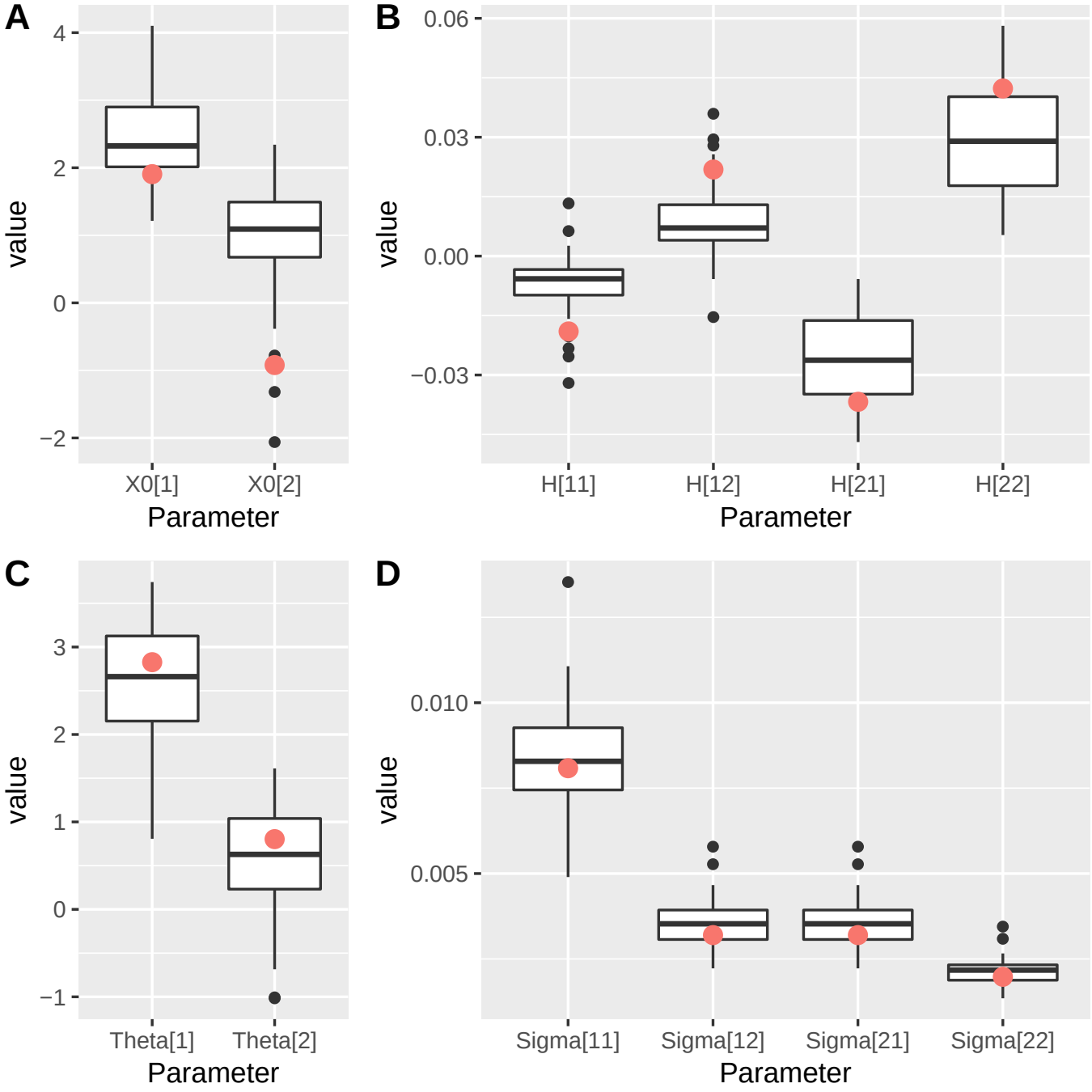


Fig. S9. Parametric bootstrap estimates for $X0$ and the parameters of regime 1. Each box-plot represents the estimates for the corresponding parameter from 49 MGPM models fit to parametric bootstrap datasets generated by simulating MGPM* on the mammal tree with regimes as depicted on Fig. 1A. The red dot in each plot shows the corresponding parameter value in MGPM*. Note: this type of analysis was possible for regime 1 only, since for this regime most of the bootstrap MGPM fits generally agreed on the set of species and type of evolutionary model (OU_F). The coordinate 1 corresponds to body-mass, while the coordinate 2 corresponds to brain-mass. A: X_0, B: paramter $\mathbf{H}$ for the OU_F model, C: paramter $\vec{\theta}$ for the OU_F model, D: paramter $\mathbf{\Sigma}$ for the OU_F model.

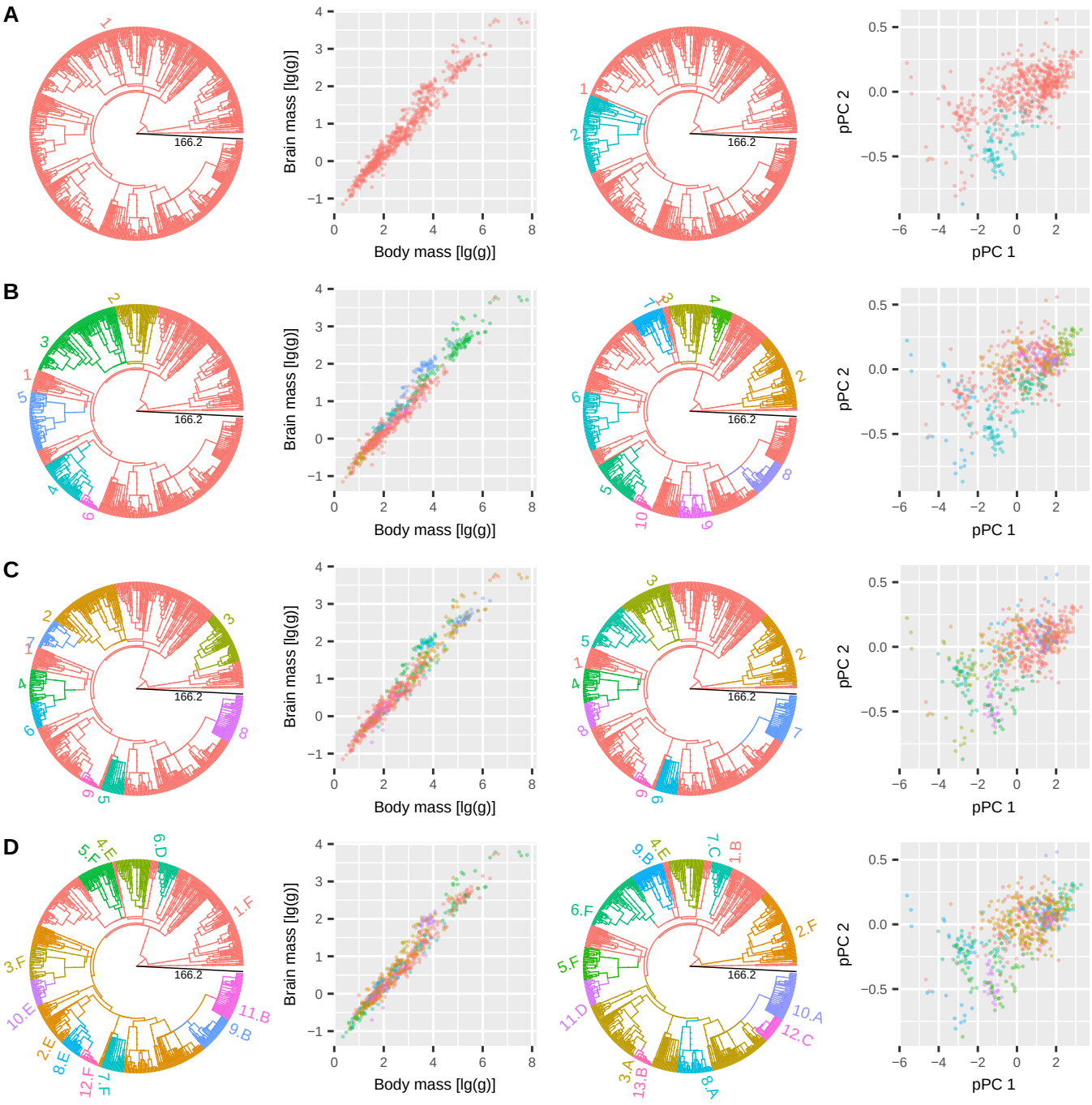


Fig. S10. Comparison between fits of models with shifts to the original mammal data and its corresponding phylogenetic principal component (pPC) scores. Each horizontal panel contains four plots for one model with shifts: left – coloured tree and scatter plot according to the model fit to the original data; right – coloured tree and scatter plot according to the model fit to the pPC scores. The colours distinguish the identified regimes for each model, but attempt has been done to match the colours corresponding to identical shifts on the left and the right plots. **A**: SURFACE OU; **B**: SCALAR OU; **C**: RATEMATRIX BM (BM_B with shifts); **D**: MGPM (A-F) with the note that for this model the two plots on the left are duplicates of Fig. 1A and B in the main text.

L.2. Supplementary figures comparing the performance of different models and inference methods on the simulated data described in SI Appendix, Section I.

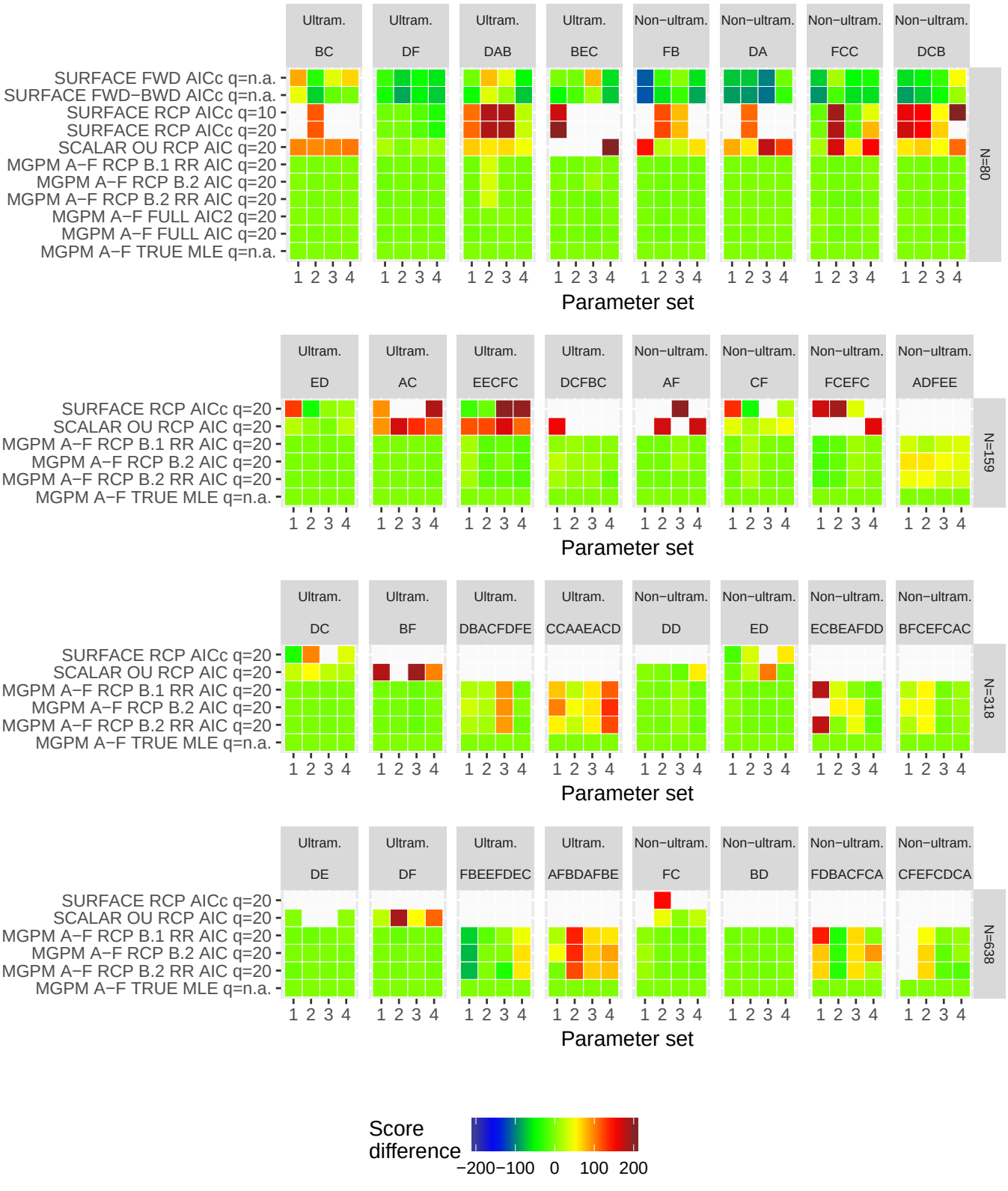


Fig. S11. Difference in the optimal score between a model fit to simulated data and the corresponding score of MGPM A-F TRUE MLE q=n.a. fit on the same data. Each square represents an average value from up to 4 simulated datasets, corresponding to a row of panels in Figs. S30-S61. Lower values are better. White squares denote values higher (worse) than the the colour scale limit. This figure shows the results for parameters sets 1, ..., 4; see Fig. S12 for parameter sets 5, ..., 8.

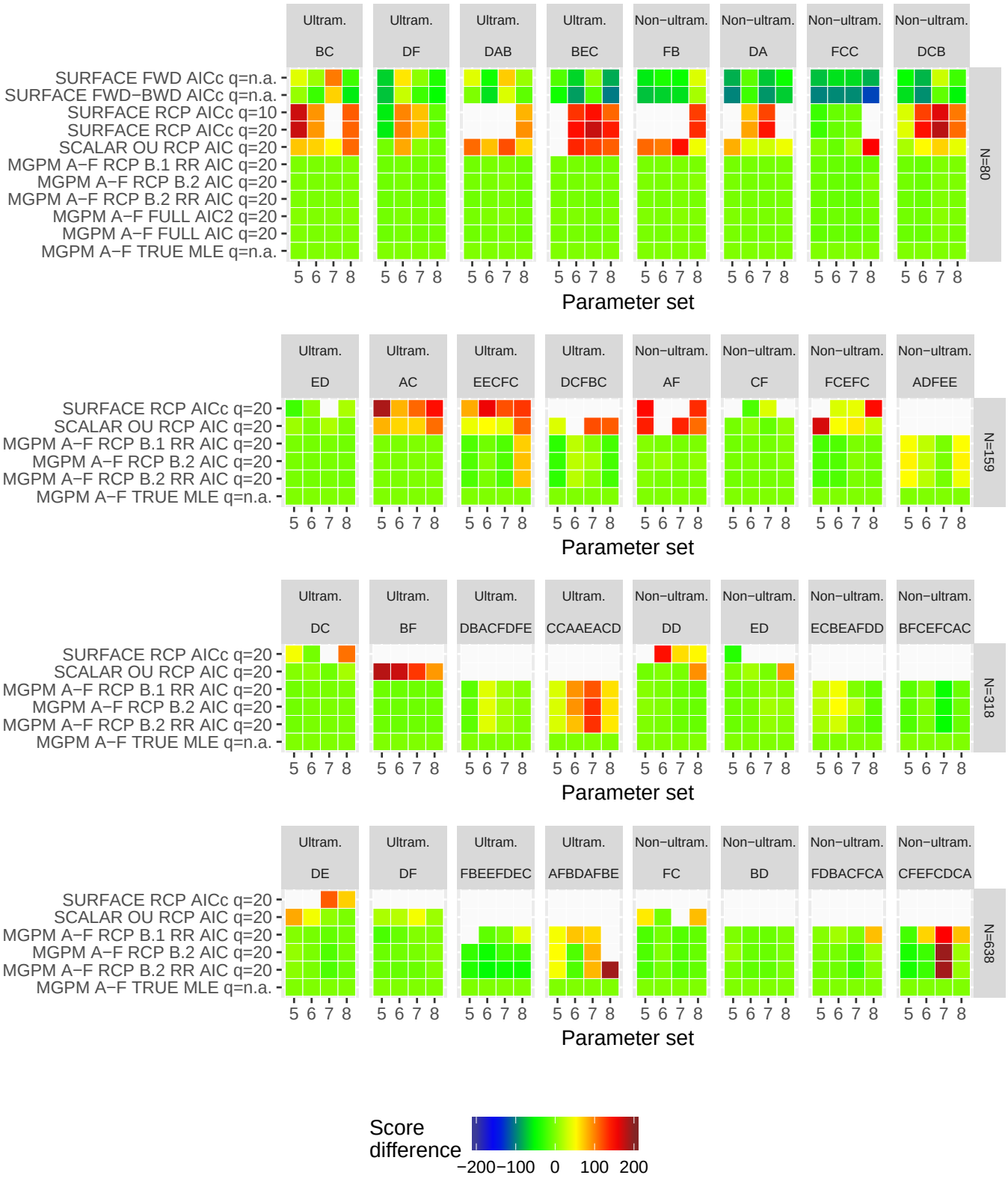


Fig. S12. Difference in the optimal score between a model fit to simulated data and the corresponding score of MGPM A-F TRUE MLE q=n.a. fit on the same data. Each square represents an average value from up to 4 simulated datasets, corresponding to a row of panels in Figs. S30-S61. Lower values are better. White squares denote values higher (worse) than the the colour scale limit. This figure shows the results for parameters sets 5, ..., 8; see Fig. S11 for parameter sets 1, ..., 4.

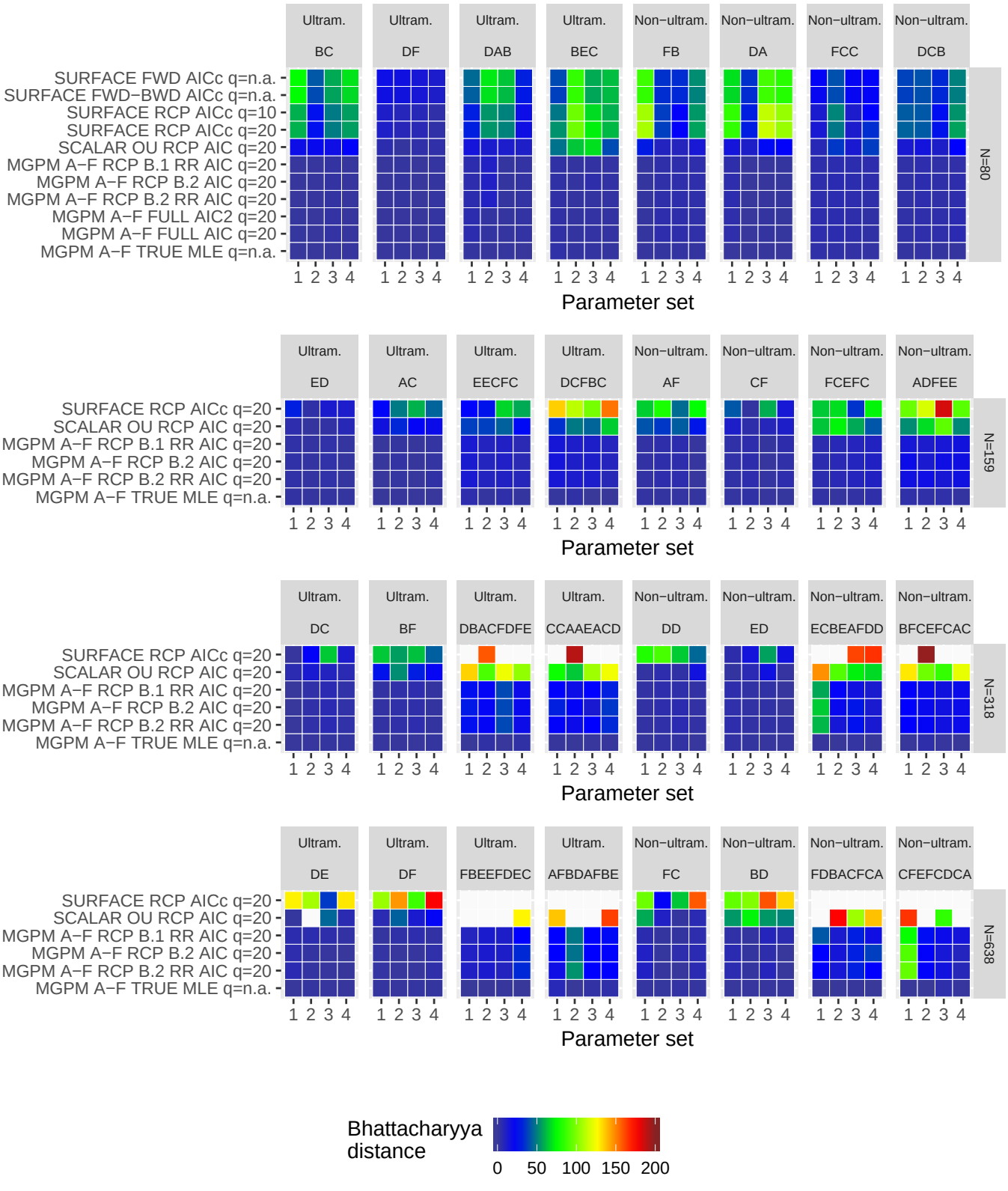


Fig. S13. Bhattacharyya distance between the expected normal distribution from a model fit to simulated data and the expected normal distribution for the true model used to simulate the data. Each square represents an average value from up to 4 simulated datasets, corresponding to a row of panels in Figs. S30-S61. Lower values are better. White squares denote values higher than the the colour scale limit. This figure shows the results for parameters sets 1, ..., 4; see Fig. S14 for parameter sets 5, ..., 8.

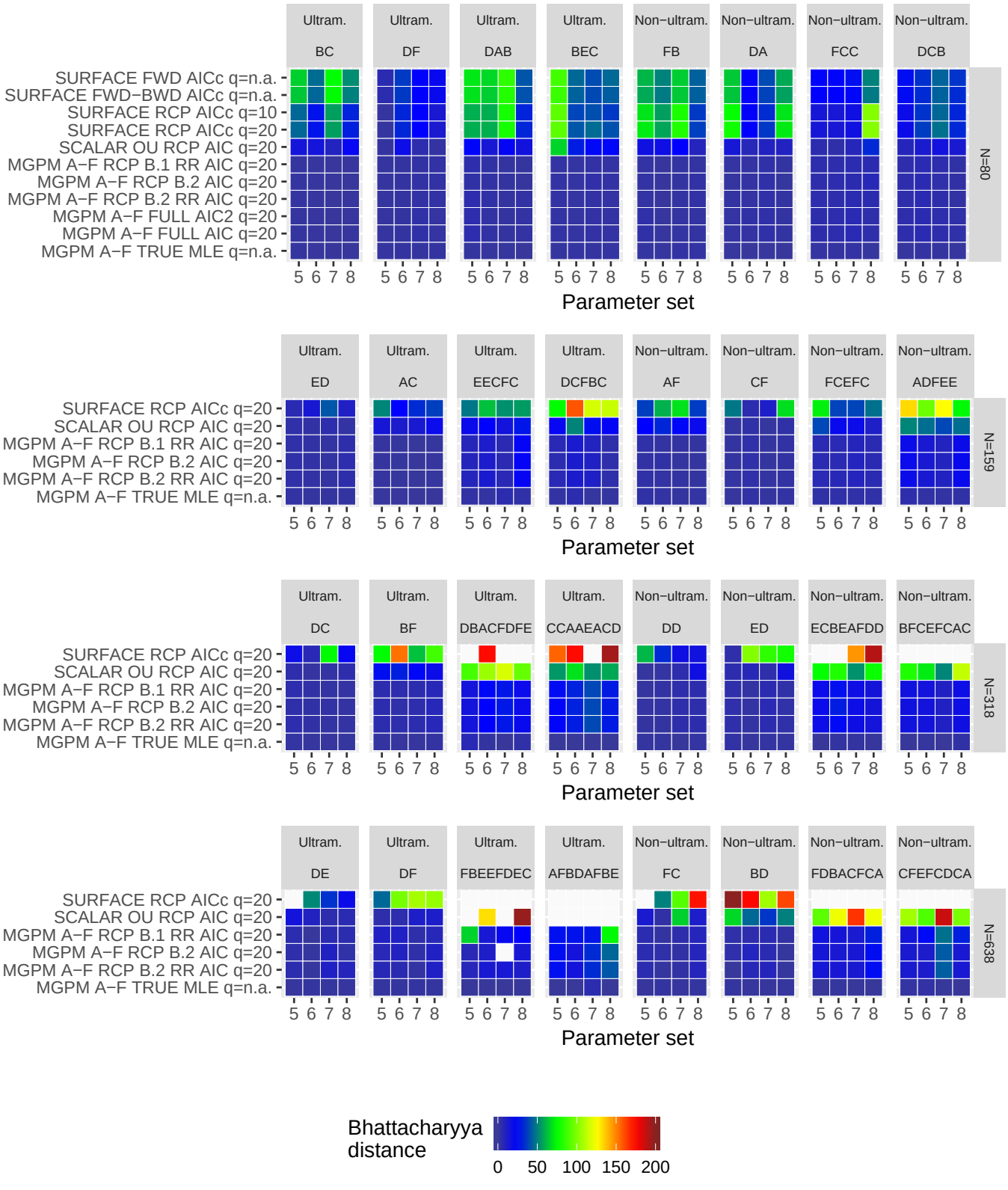


Fig. S14. Bhattacharyya distance between the expected normal distribution from a model fit to simulated data and the expected normal distribution for the true model used to simulate the data. Lower values are better. White squares denote values higher than the the colour scale limit. This figure shows the results for parameters sets 5, ..., 8; see Fig. S13 for parameter sets 1, ..., 4.

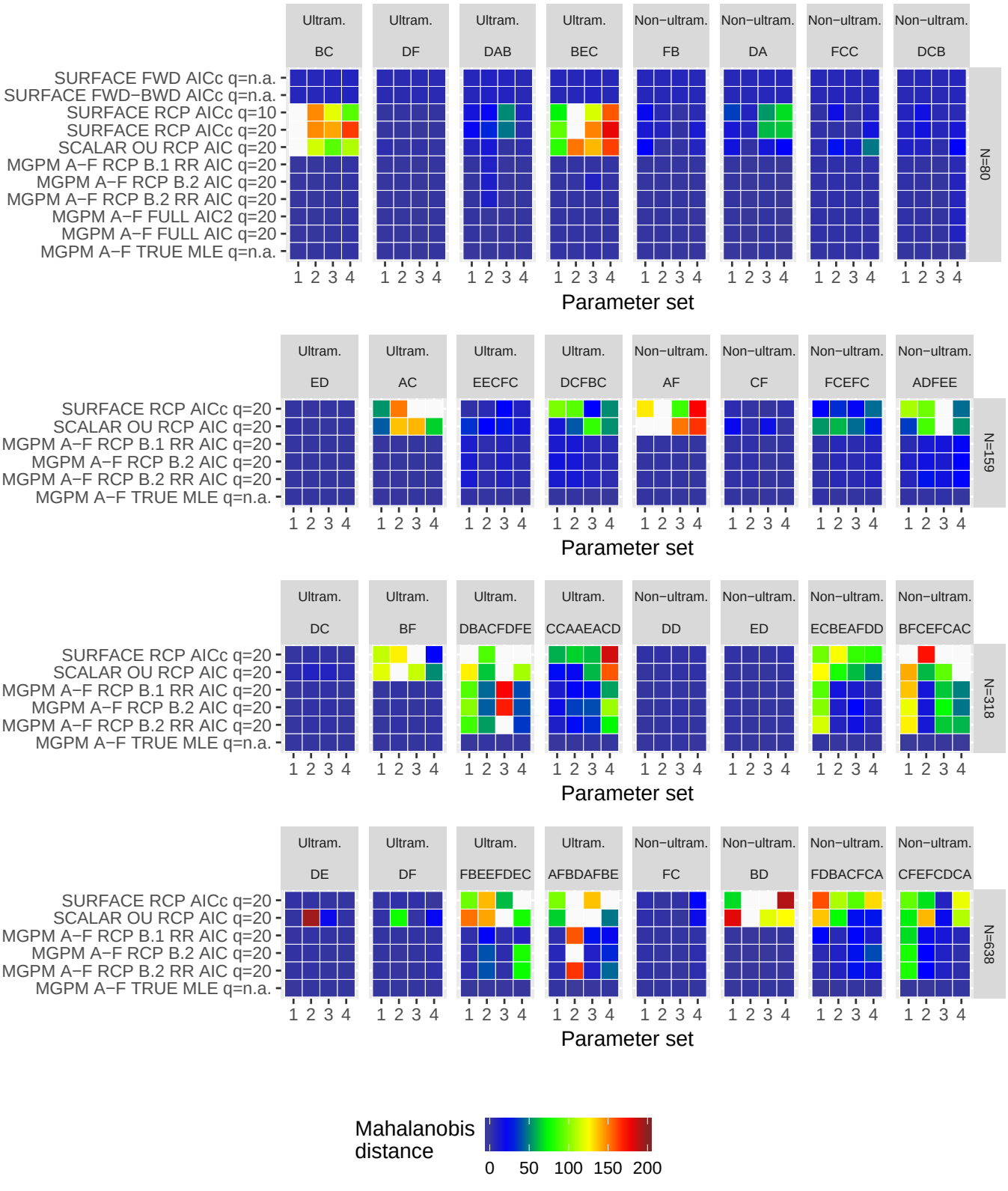


Fig. S15. Mahalanobis distance between the expected normal distribution from a model fit to simulated data and the expected normal distribution for the true model used to simulate the data. White squares denote values higher than the the colour scale limit. This figure shows the results for parameters sets 1, ..., 4; see Fig. S16 for parameter sets 5, ..., 8.

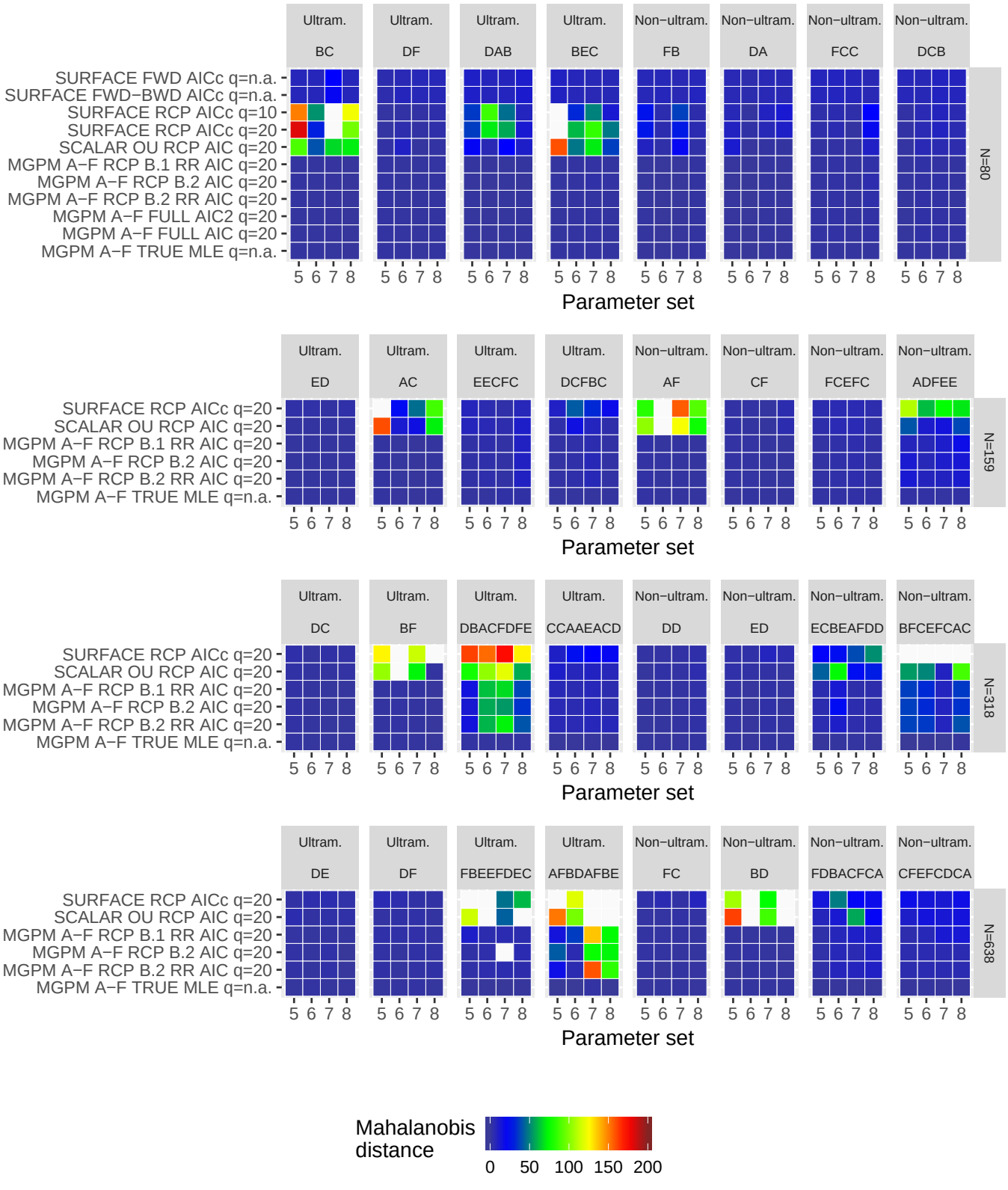


Fig. S16. Mahalanobis distance between the expected normal distribution from a model fit to simulated data and the expected normal distribution for the true model used to simulate the data. White squares denote values higher than the the colour scale limit. This figure shows the results for parameters sets 5, ..., 8; see Fig. S15 for parameter sets 1, ..., 4.

Fig. S17. Difference in the number of regimes between a model fit to simulated data and the number of regimes in the true model used to simulate the data. Each square represents an average value from up to 4 simulated datasets, corresponding to a row of panels in Figs. S30-S61. Values closer to 0 are better. Negative values indicate negative bias, i.e. the model infers fewer regimes than the true number. White squares denote values higher than the the colour scale limit. This figure shows the results for parameters sets 1, ..., 4; see Fig. S18 for parameter sets 5, ..., 8.

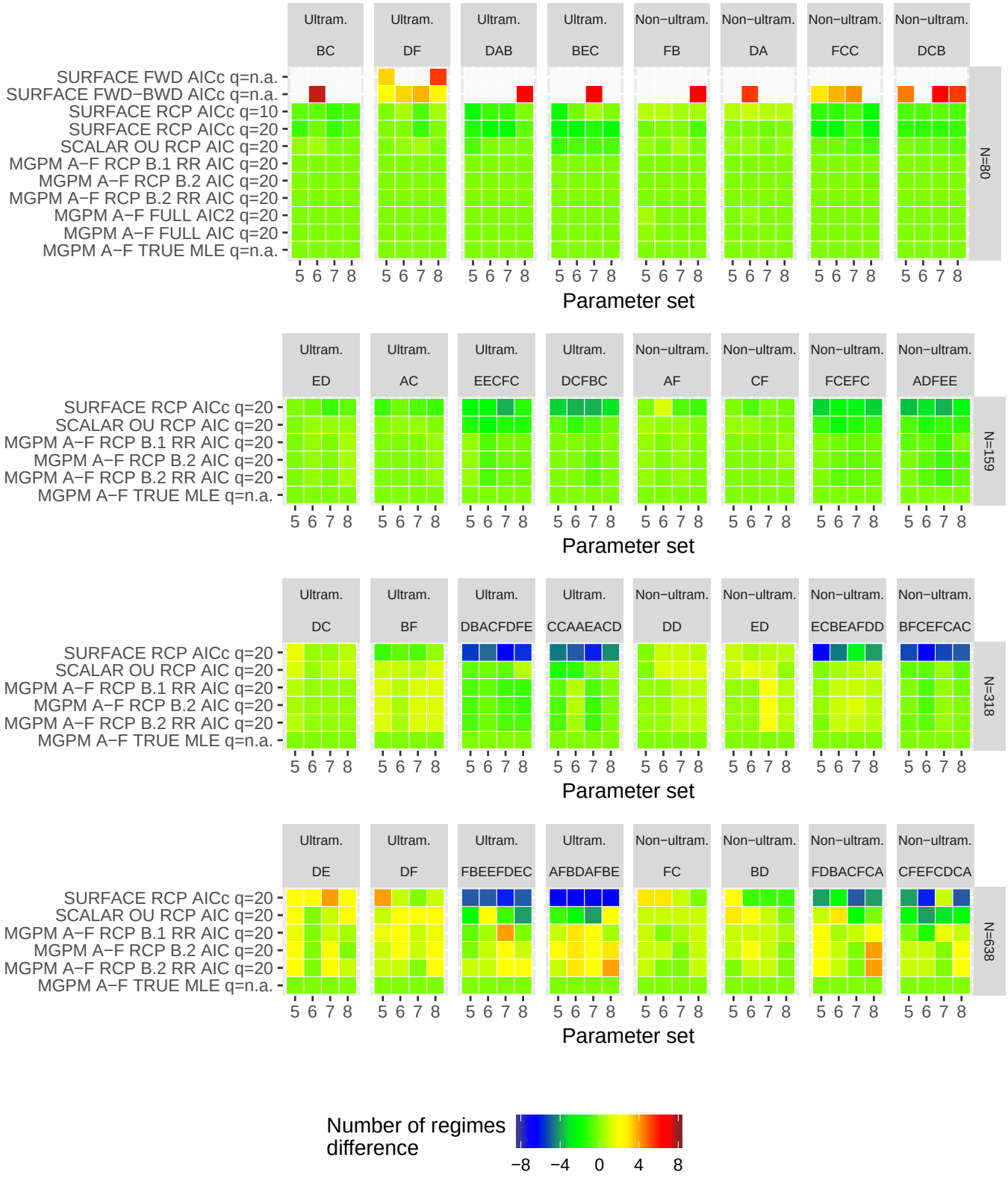


Fig. S18. Difference in the number of regimes between a model fit to simulated data and the number of regimes in the true model used to simulate the data. Each square represents an average value from up to 4 simulated datasets, corresponding to a row of panels in Figs. S30-S61. Values closer to 0 are better. Negative values indicate negative bias, i.e. the model infers fewer regimes than the true number. White squares denote values higher than the the colour scale limit. This figure shows the results for parameters sets 5, ..., 8; see Fig. S17 for parameter sets 1, ..., 4.

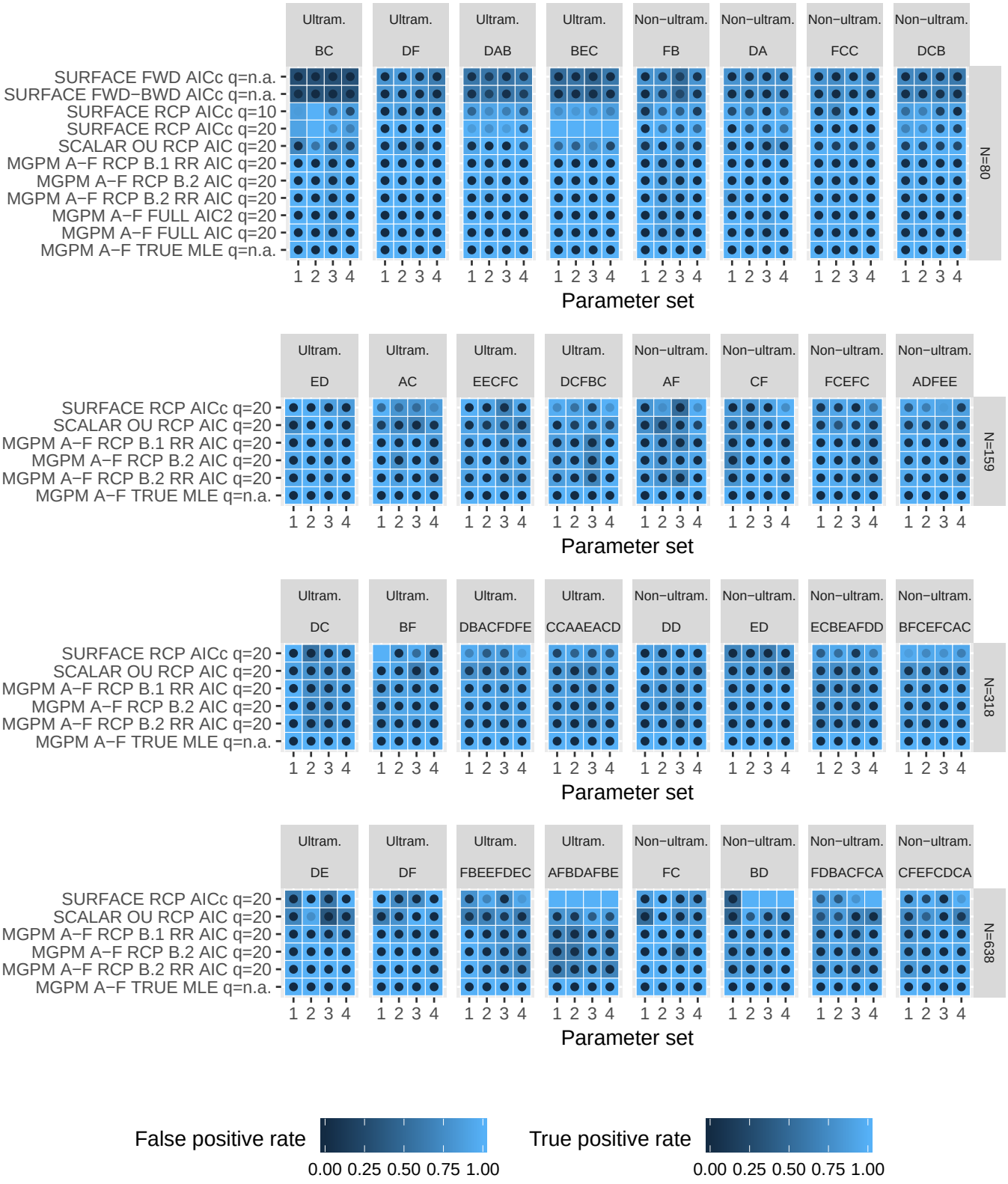


Fig. S19. Performance of the inferred models with respect to the "Cluster" criterion. Each square with a circle represents the average true positive rate (square background) versus the average false positive rate (point inside the square) from up to four simulations for a given parameter set. We say that a method is performing well if the corresponding square is bright (high true positive rate), while the inner circle is dark (low false positive rate). This legend applies to all the figures reporting binary criteria. If for a given case the square or the inner circle is painted in grey, this indicates that the true positive or the false positive rate cannot be evaluated for this case. This figure shows the results for parameters sets 1, ..., 4; see Fig. S20 for parameter sets 5, ..., 8.

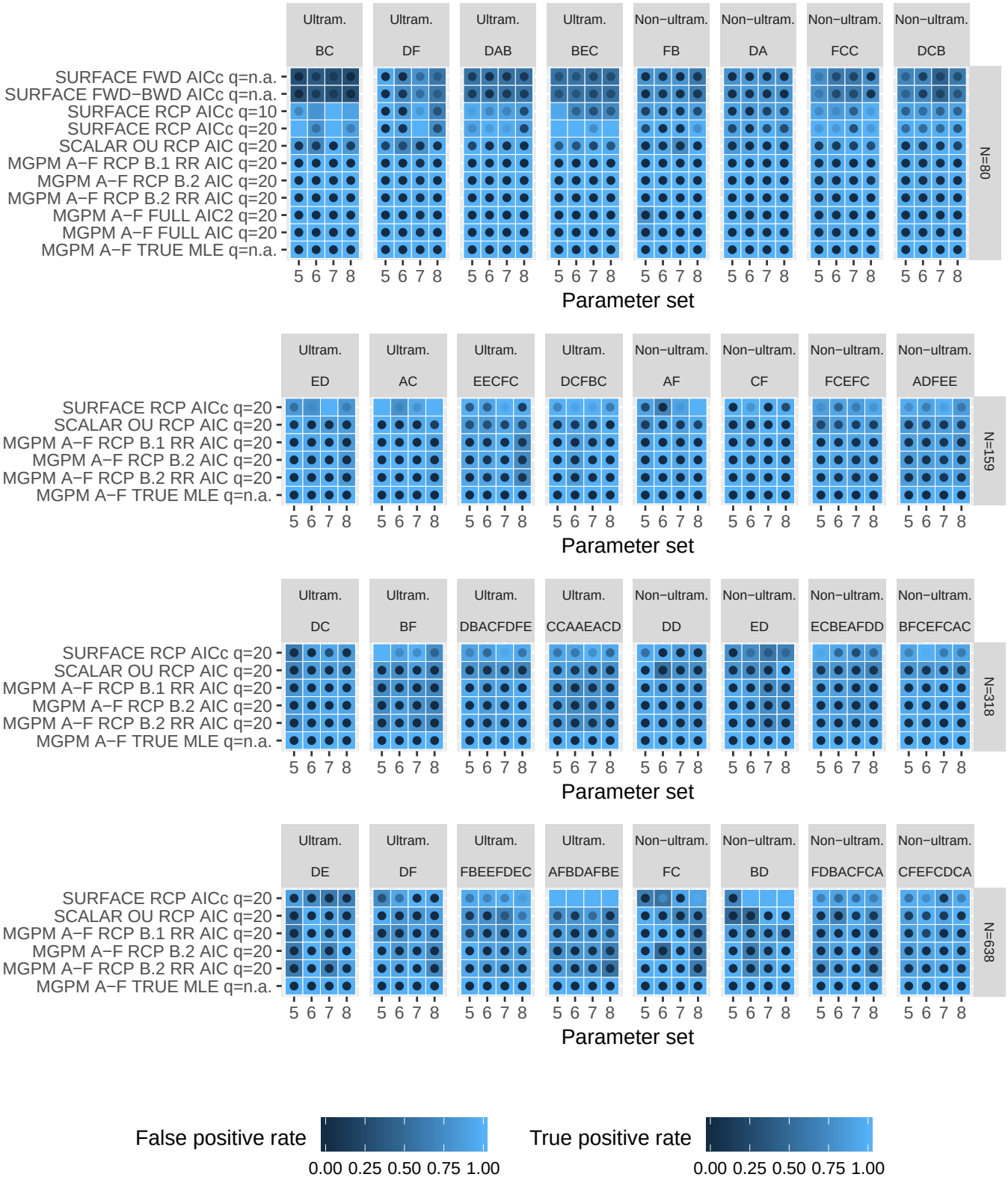


Fig. S20. Performance of the inferred models with respect to the "Cluster" criterion. Each square with a circle represents the average true positive rate (square background) versus the average false positive rate (point inside the square) from up to four simulations for a given parameter set. We say that a method is performing well if the corresponding square is bright (high true positive rate), while the inner circle is dark (low false positive rate). This legend applies to all the figures reporting binary criteria. If for a given case the square or the inner circle is painted in grey, this indicates that the true positive or the false positive rate cannot be evaluated for this case. This figure shows the results for parameters sets 5, ..., 8; see Fig. S19 for parameter sets 1, ..., 4.

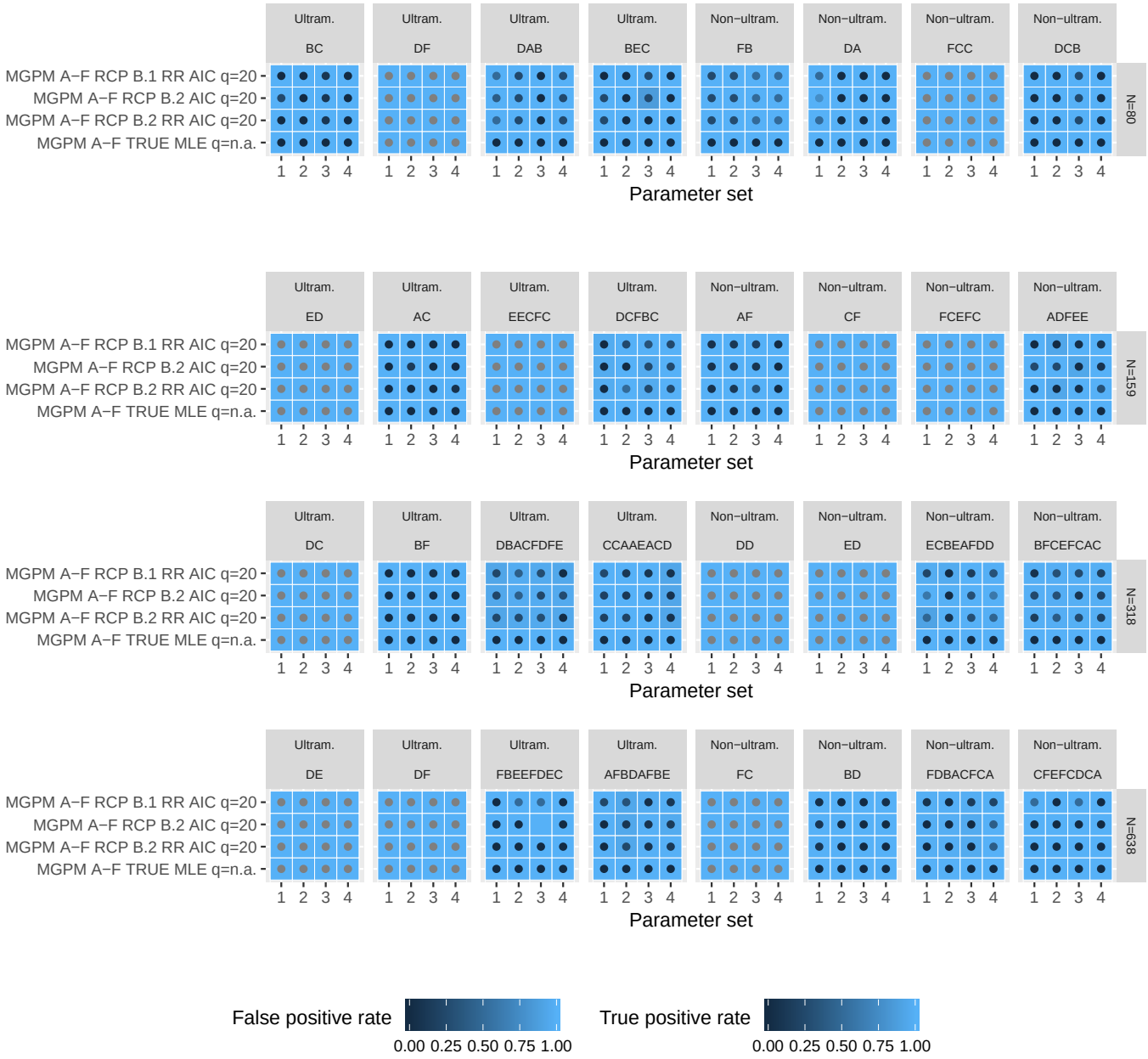


Fig. S21. Performance evaluation for the "OU process" criterion. For a description, see the legend for Figs. S19-S20. This figure shows the results for parameters sets 1, ..., 4; see Fig. S22 for parameter sets 5, ..., 8.

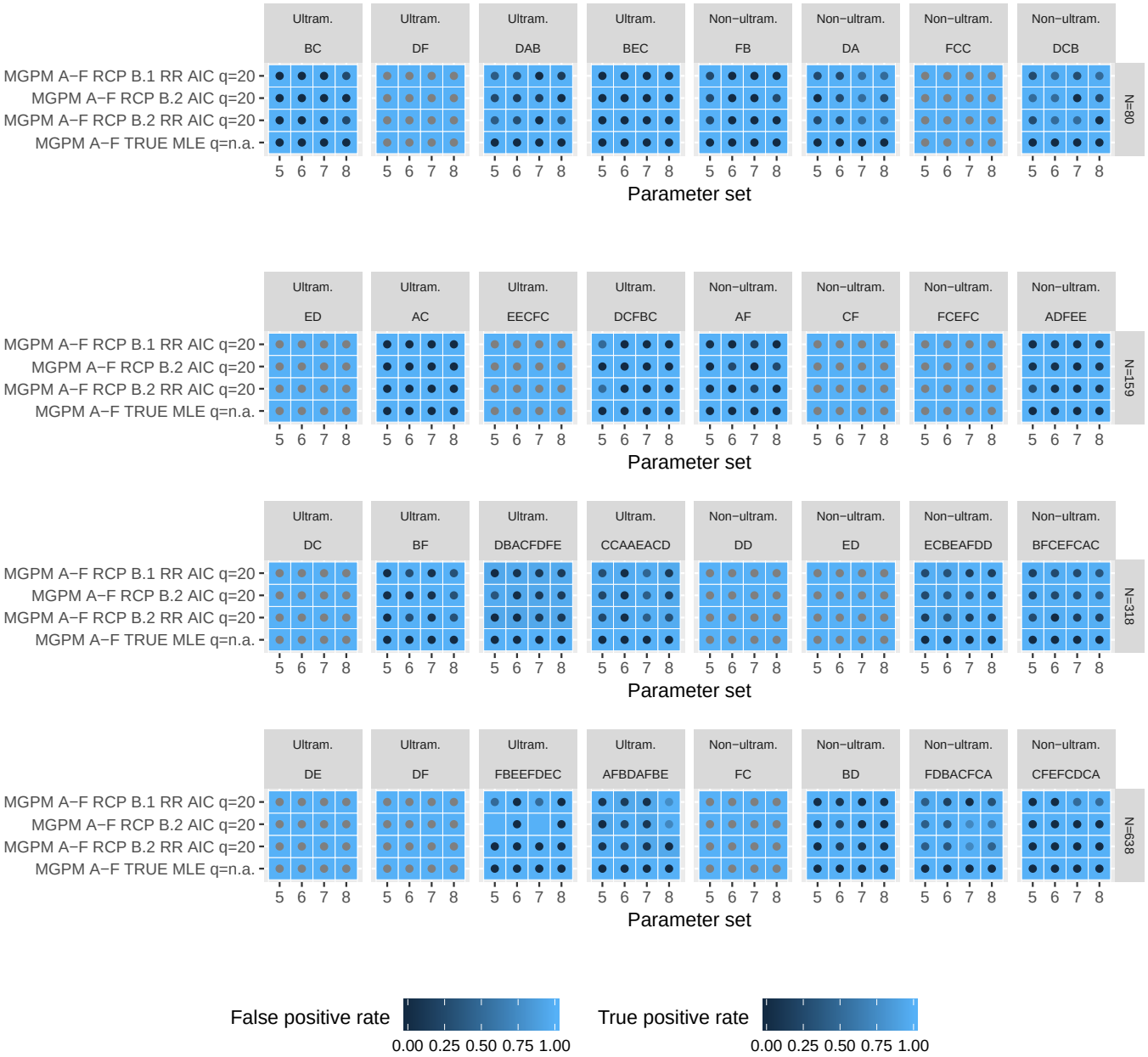


Fig. S22. Performance evaluation for the "OU process" criterion. For a description, see the legend for Figs. S19-S20. This figure shows the results for parameters sets 5, ..., 8; see Fig. S21 for parameter sets 1, ..., 4.

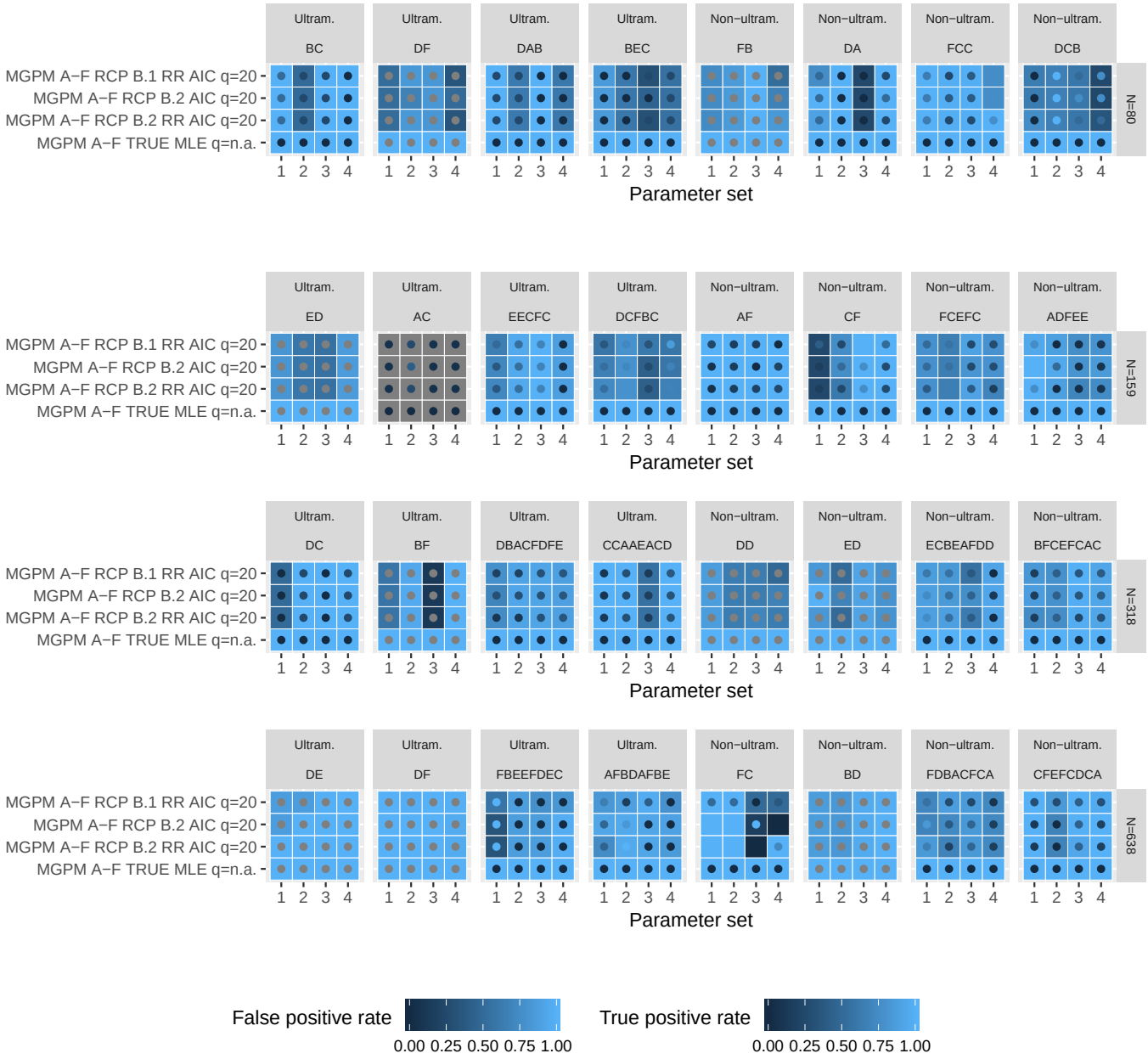


Fig. S23. Performance evaluation for the "Correlated traits" criterion. For a description, see the legend for Figs. S19-S20. This figure shows the results for parameters sets 1, ..., 4; see Fig. S24 for parameter sets 5, ..., 8.

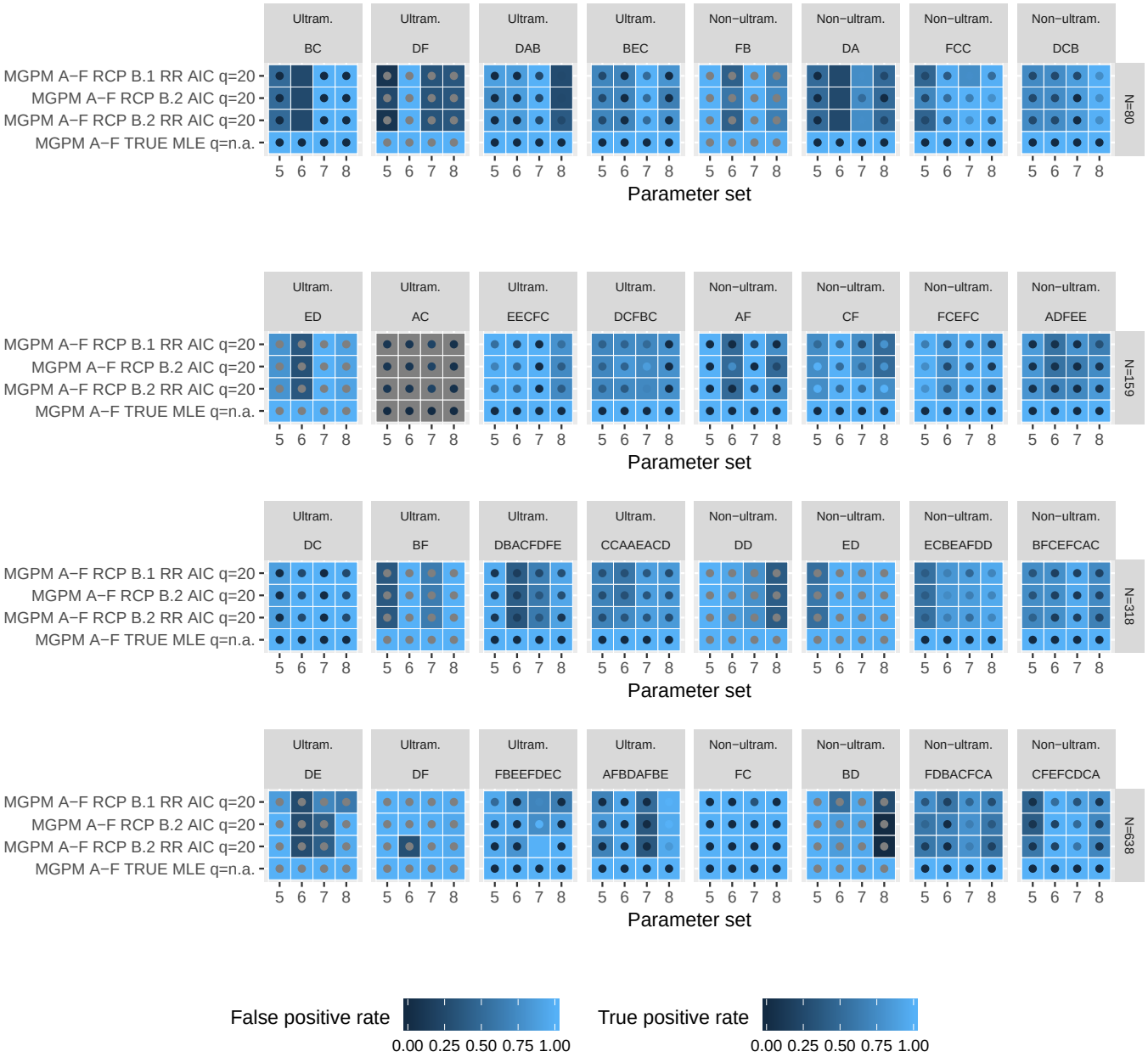


Fig. S24. Performance evaluation for the "Correlated traits" criterion. For a description, see the legend for Figs. S19-S20. This figure shows the results for parameters sets 5, ..., 8; see Fig. S23 for parameter sets 1, ..., 4.

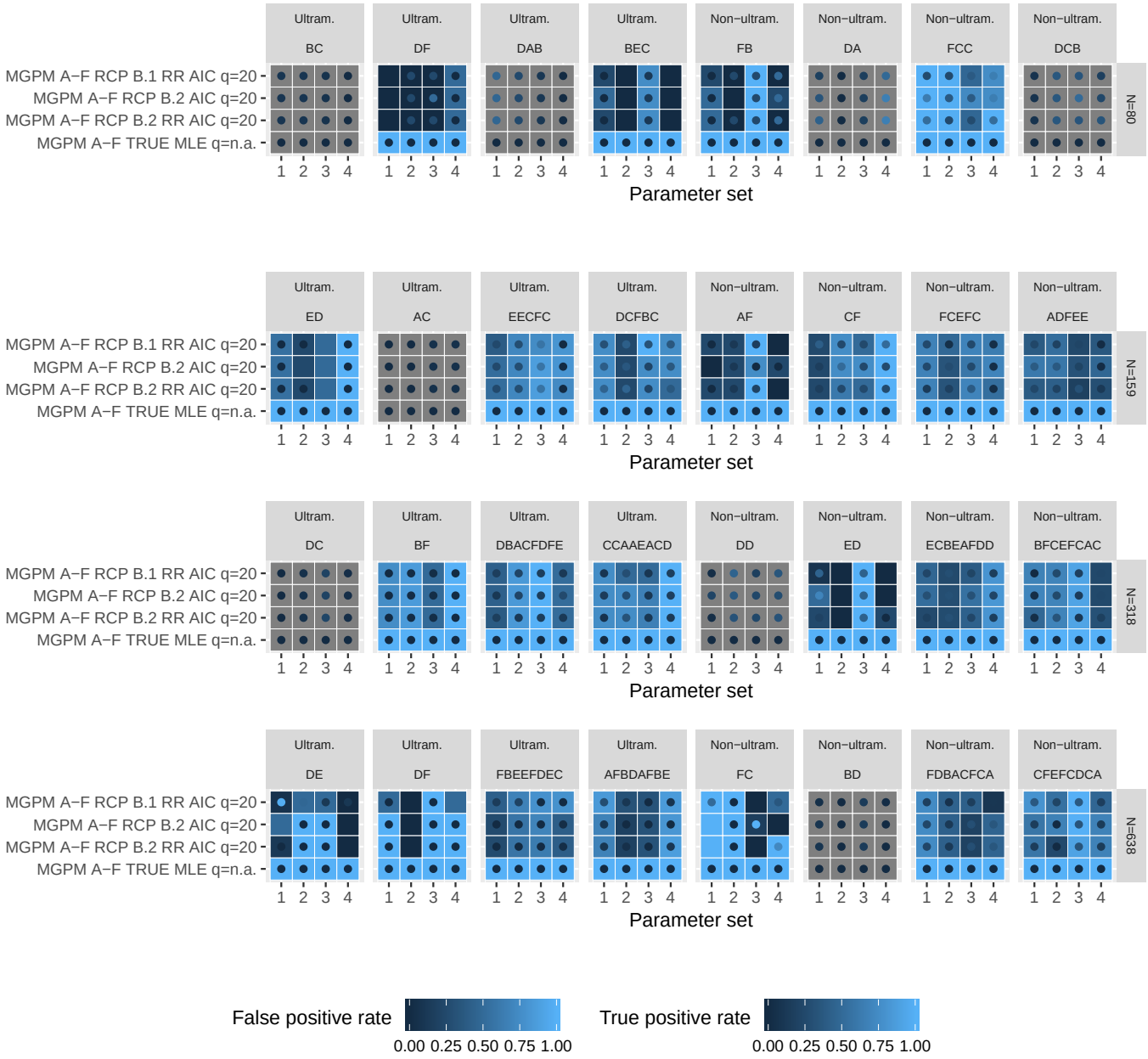


Fig. S25. Performance evaluation for the "NonDiagonal H" criterion. For a description, see the legend for Figs. S19-S20. This figure shows the results for parameters sets 1, ..., 4; see Fig. S26 for parameter sets 5, ..., 8.

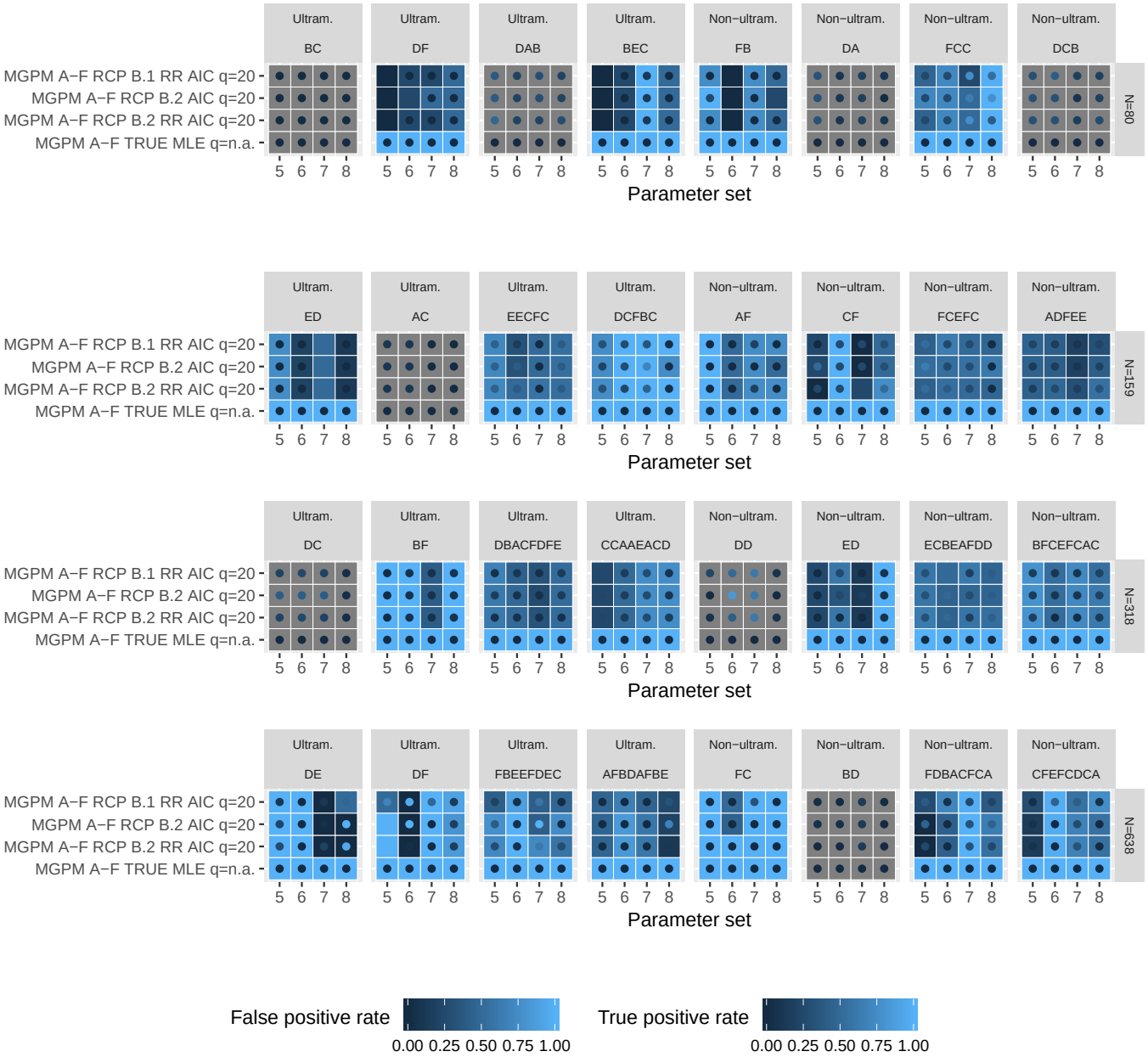


Fig. S26. Performance evaluation for the "NonDiagonal H" criterion. For a description, see the legend for Figs. S19-S20. This figure shows the results for parameters sets 5, ..., 8; see Fig. S25 for parameter sets 1, ..., 4.

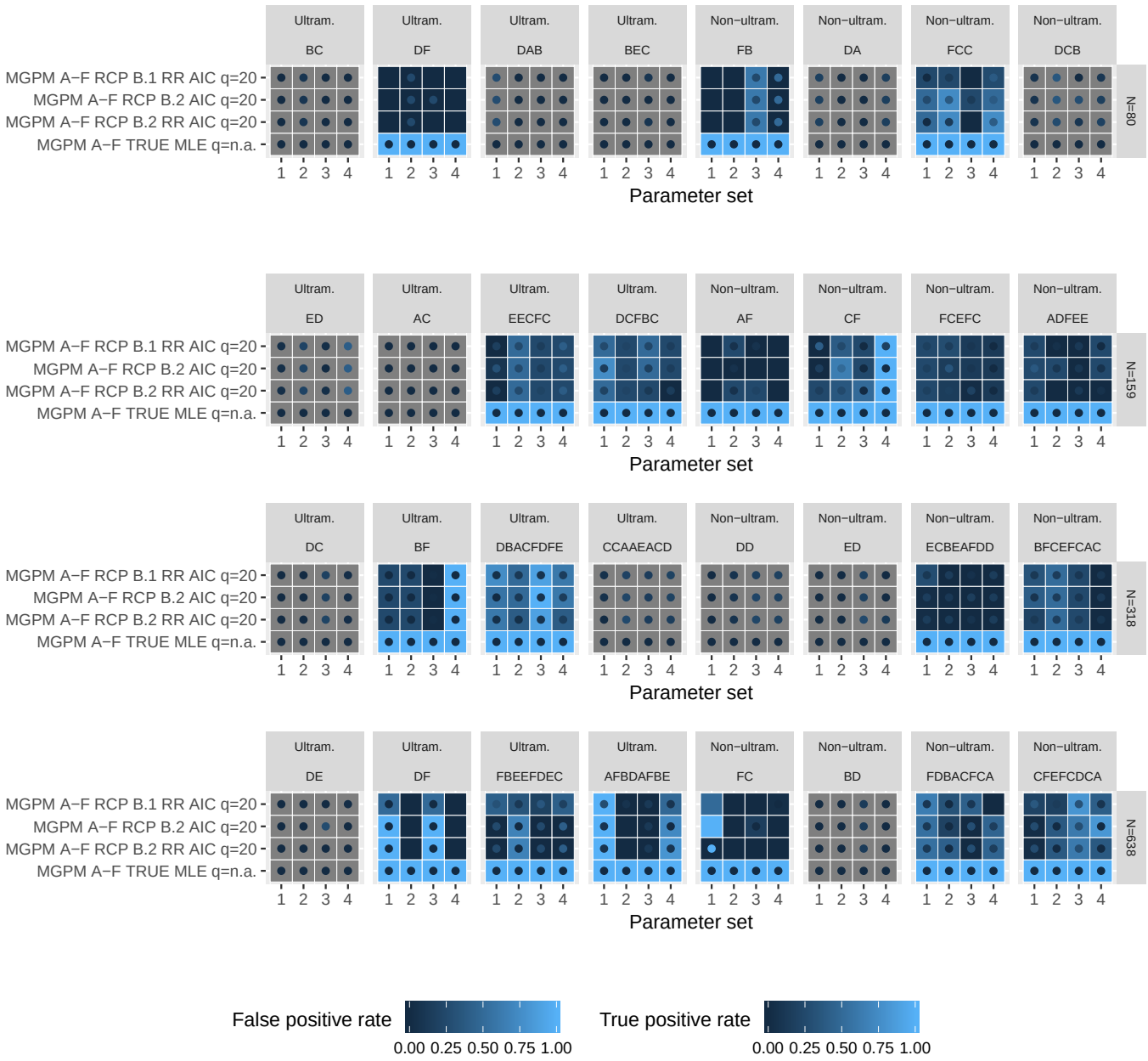


Fig. S27. Performance evaluation for the "Asymmetric H" criterion. For a description, see the legend for Figs. S19-S20. This figure shows the results for parameters sets 1, ..., 4; see Fig. S28 for parameter sets 5, ..., 8.

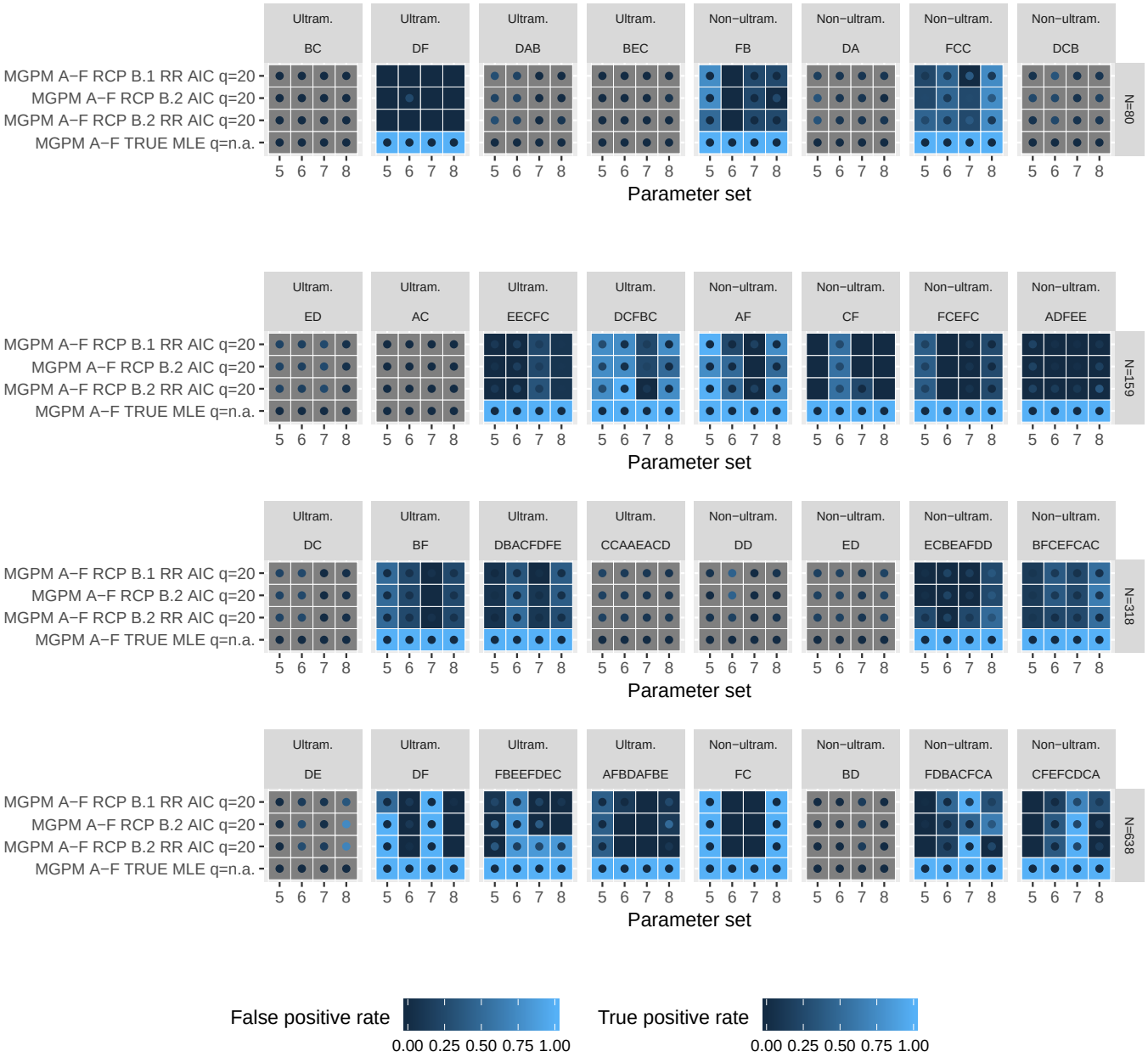


Fig. S28. Performance evaluation for the "Asymmetric H" criterion. For a description, see the legend for Figs. S19-S20. This figure shows the results for parameters sets 5, ..., 8; see Fig. S27 for parameter sets 1, ..., 4.

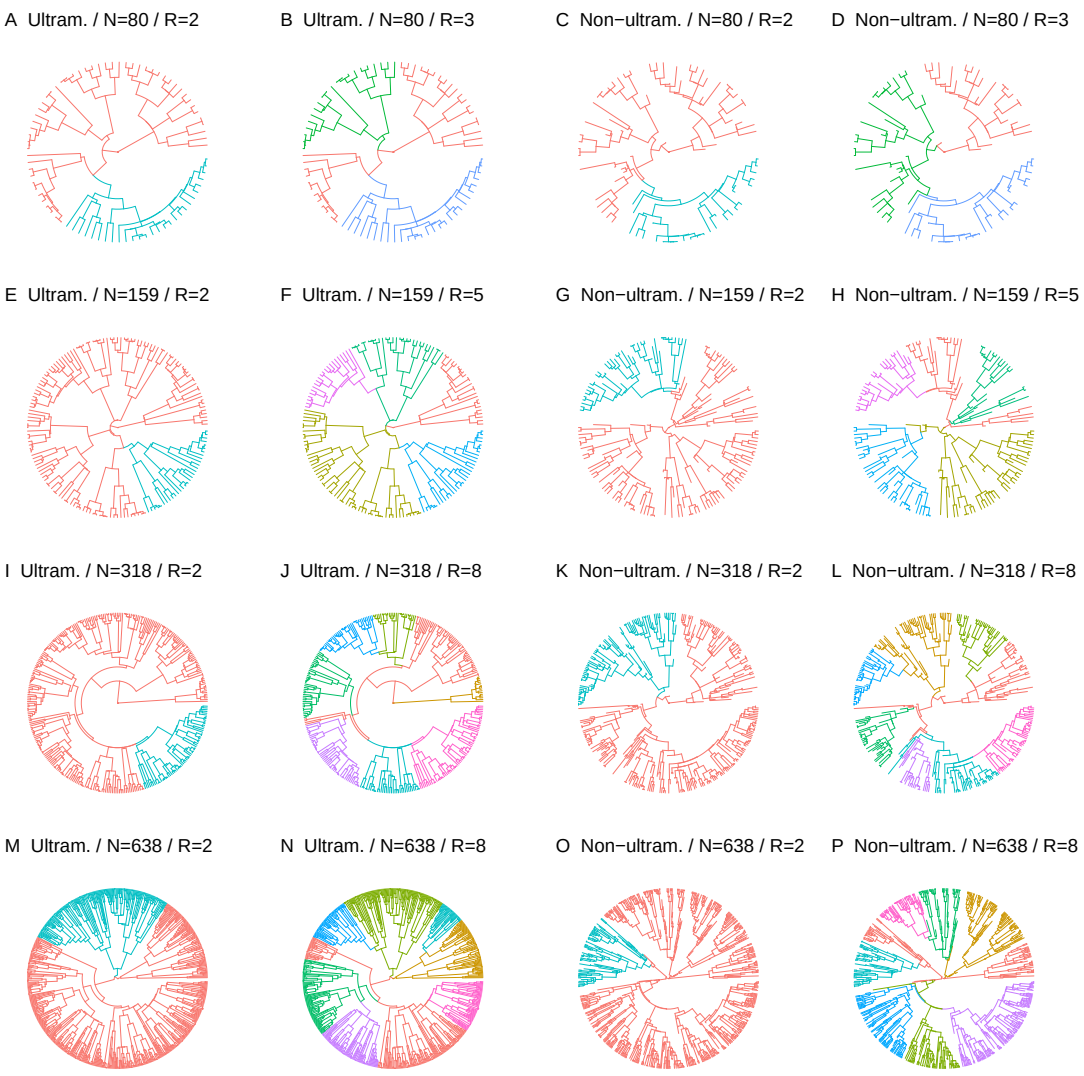


Fig. S29. Simulated birth-death trees.

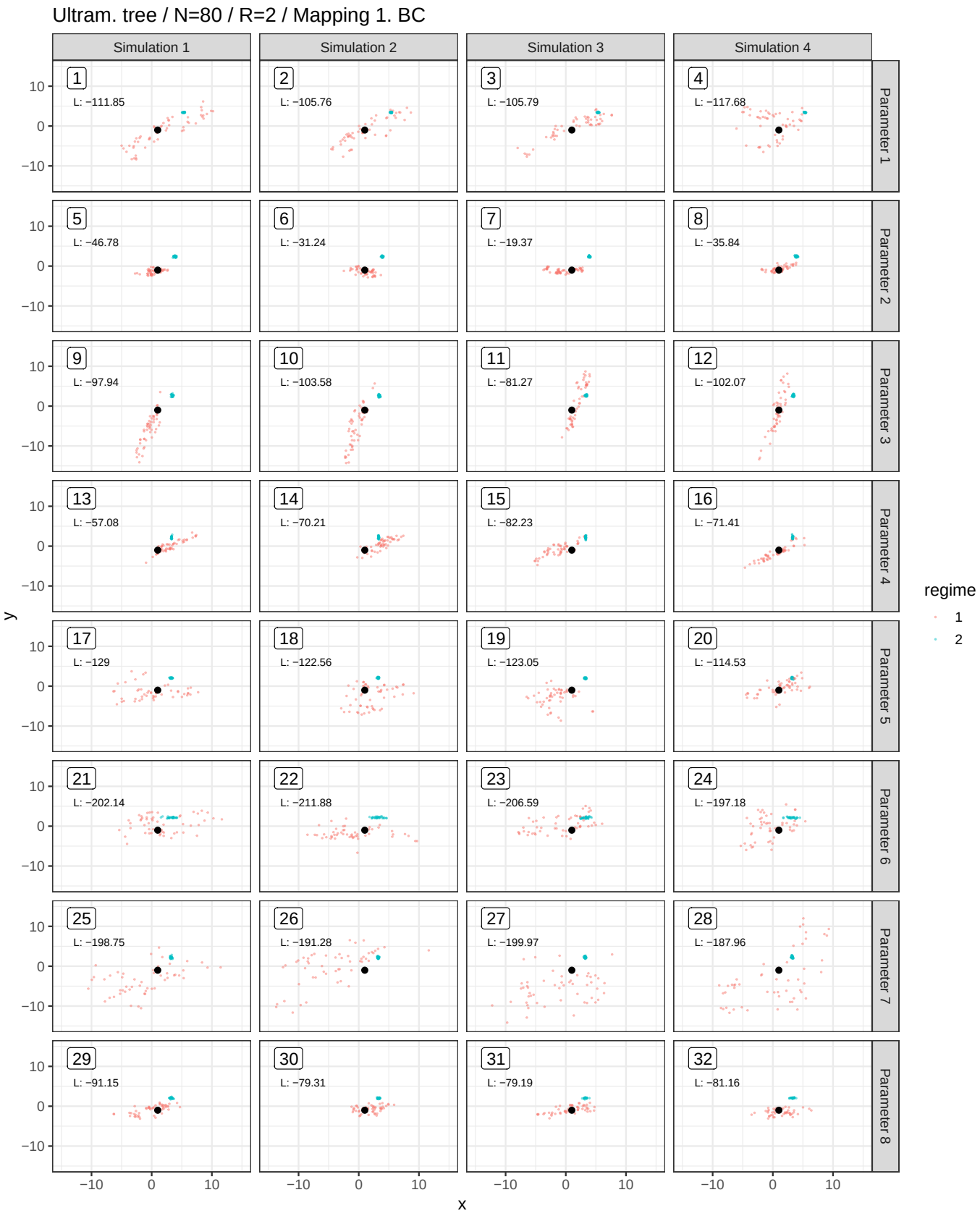


Fig. S30. Scatter plots of the simulated trait data. In each panel, each coloured point represents the values of the two simulated traits (x and y) for one tip in the tree. The number label in the top-left corner of each panel denotes the identifier of the simulation, which can be used to look-up the simulated data in the testData_t5 data.table of the accompanying package MGPMSimulations. The label denoted by capital letter L denotes the log-likelihood of the data evaluated under the true model. A black point denotes the starting trait value for each simulation.

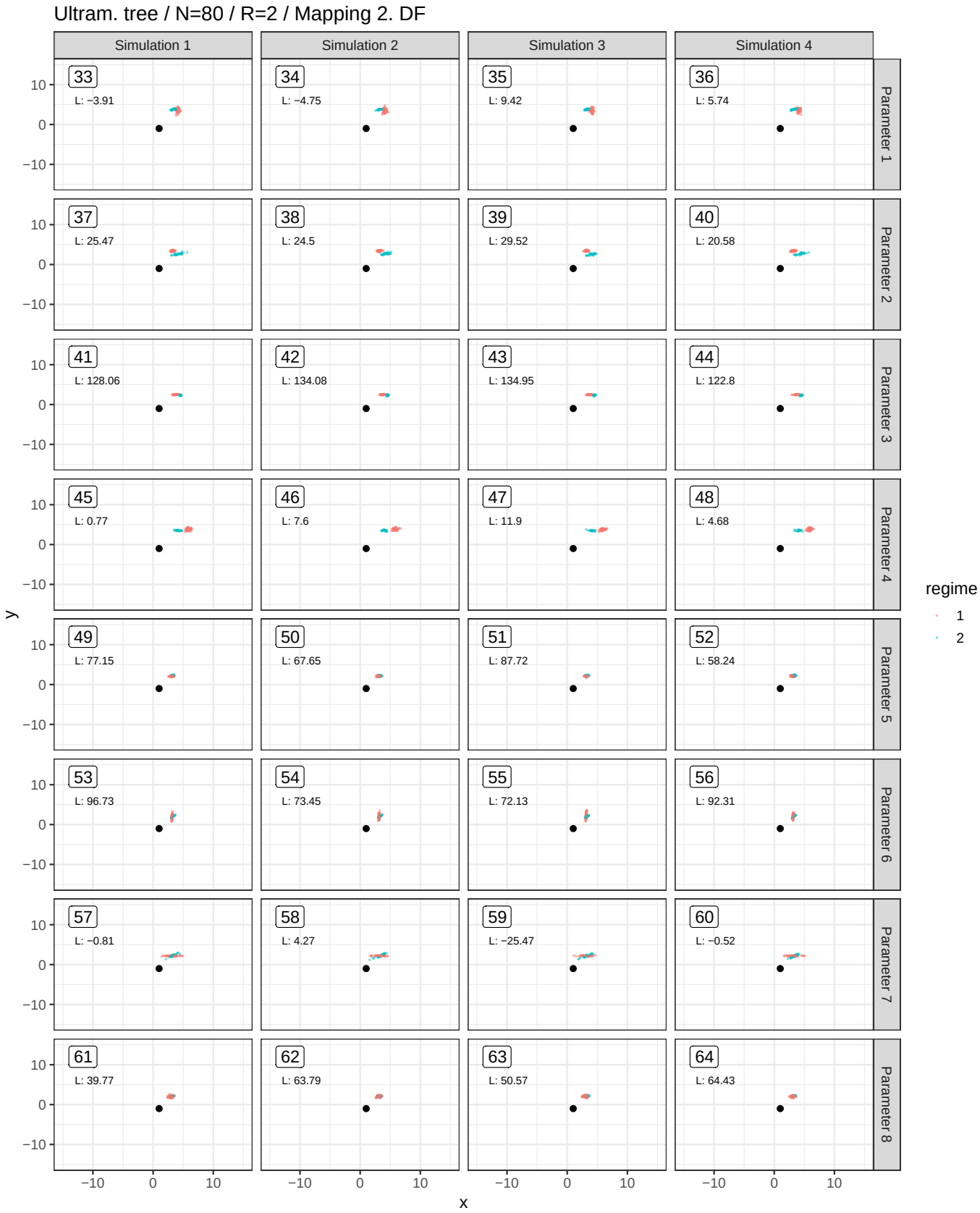


Fig. S31. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

Fig. S32. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

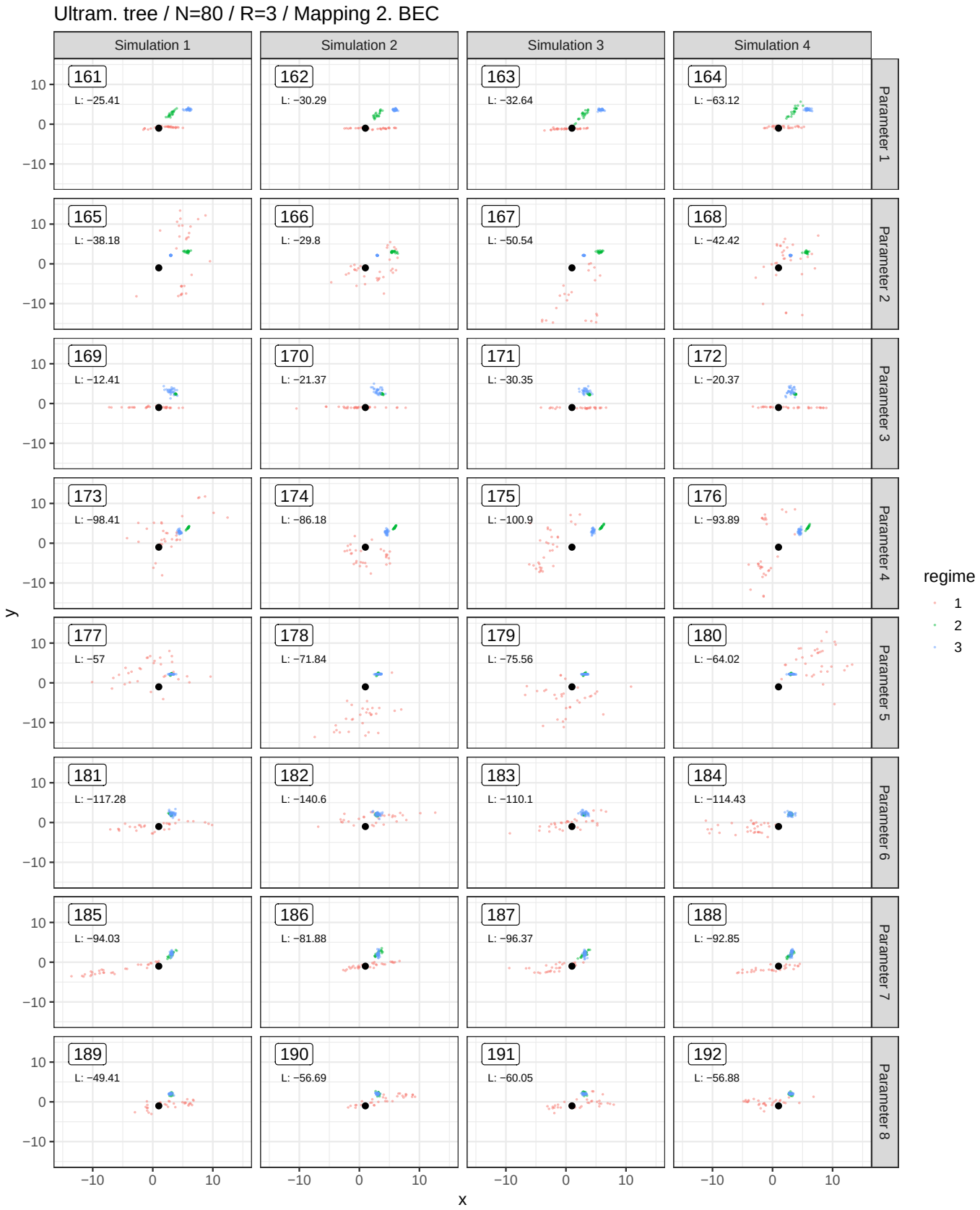


Fig. S33. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

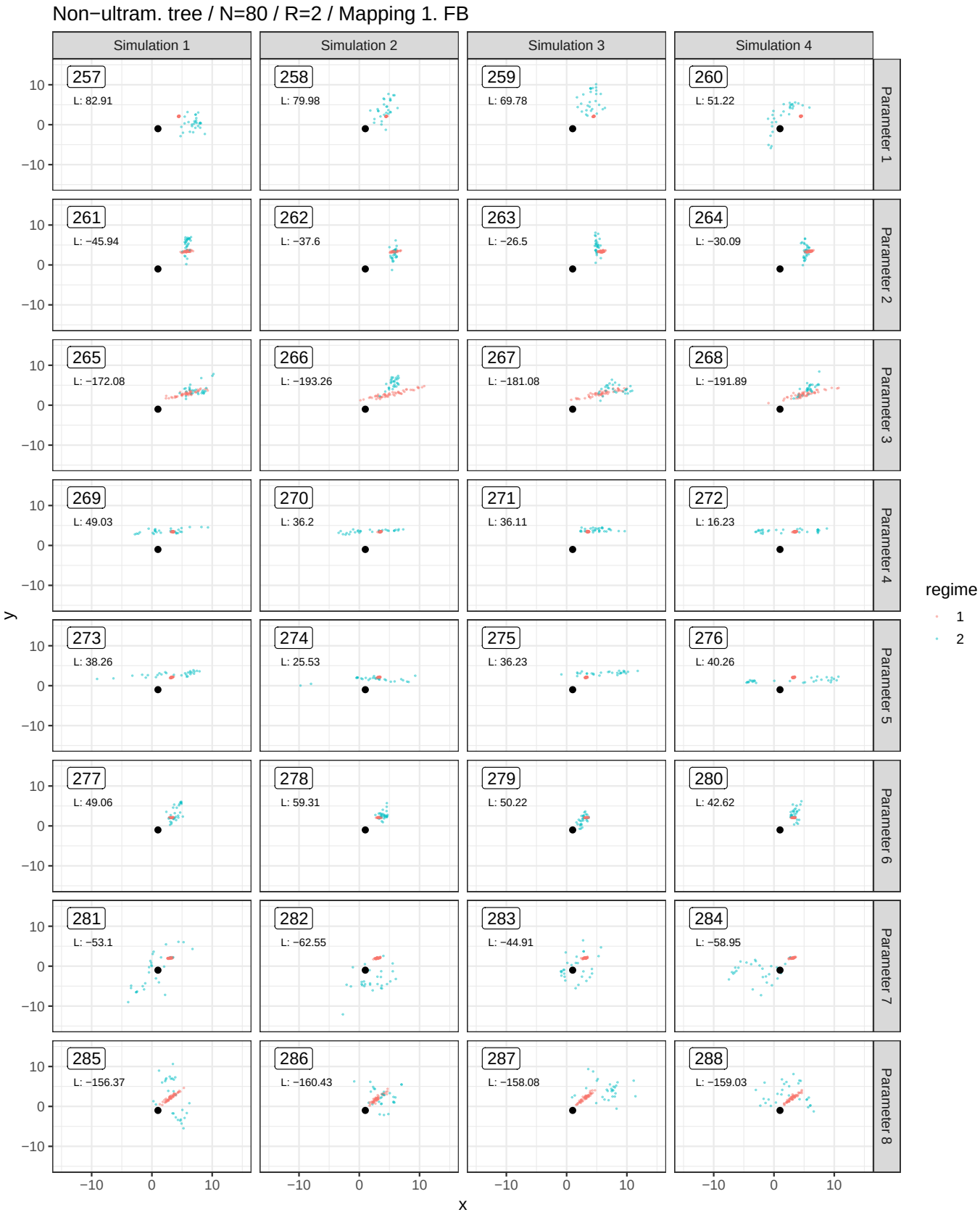


Fig. S34. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

Fig. S35. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

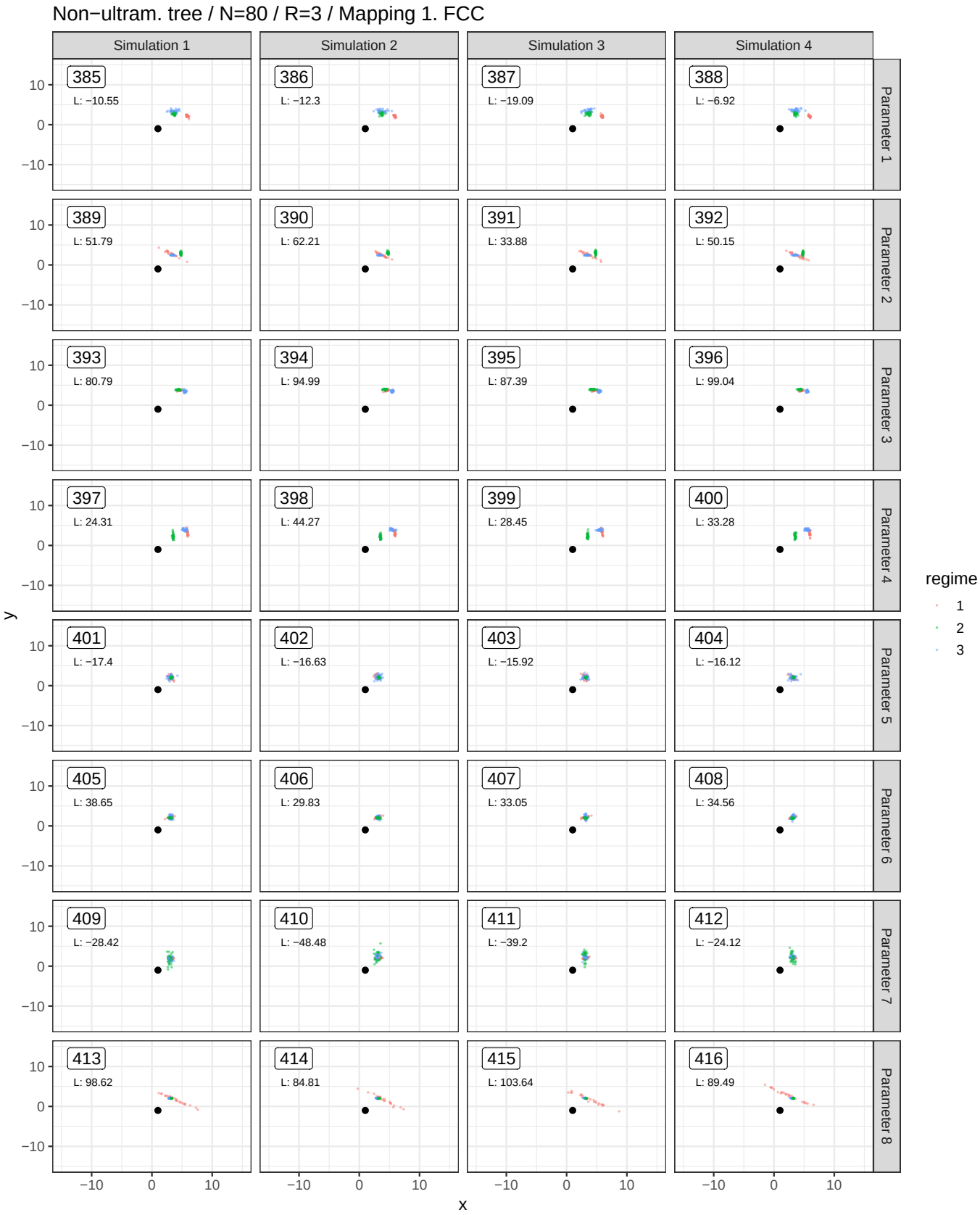


Fig. S36. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

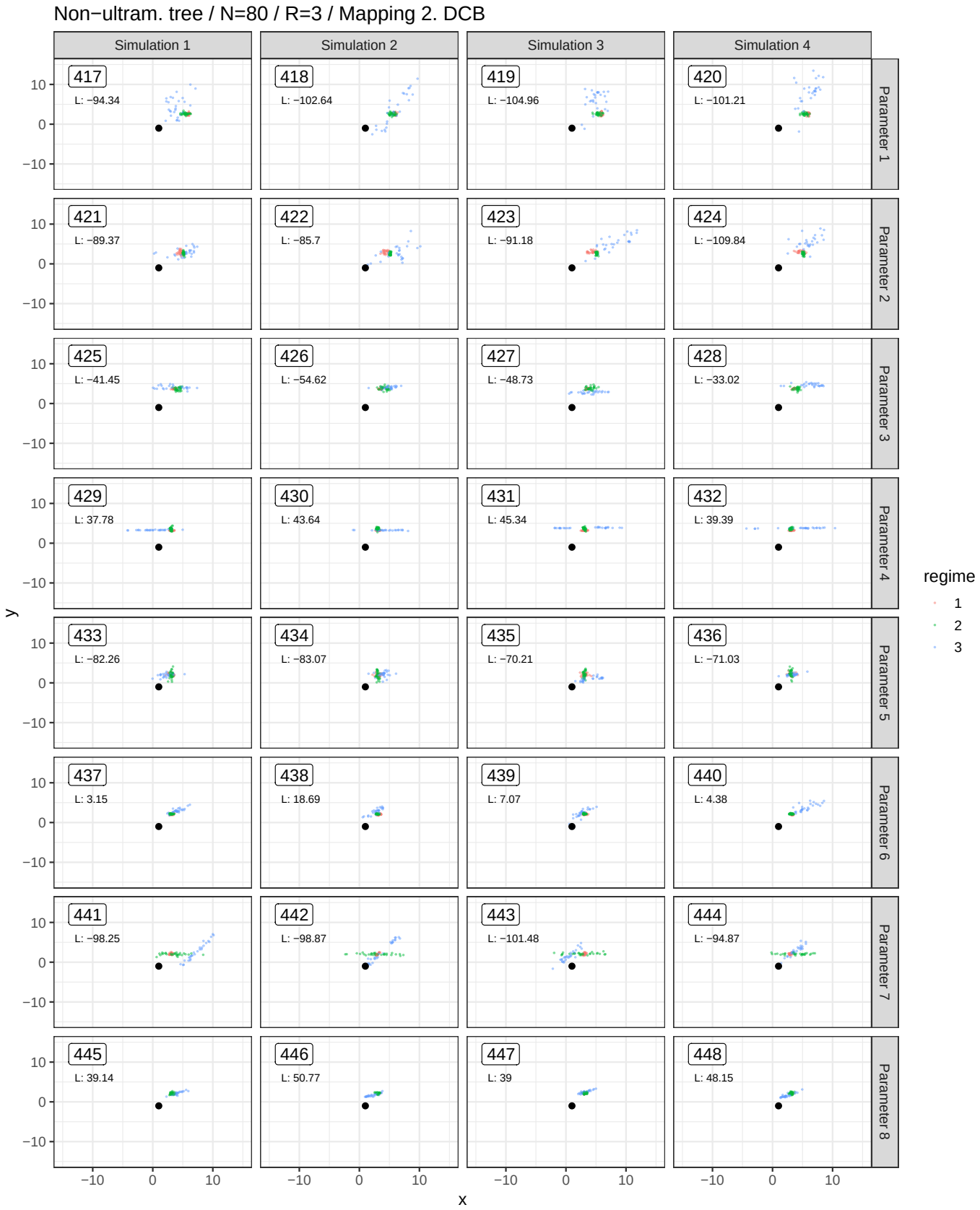


Fig. S37. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

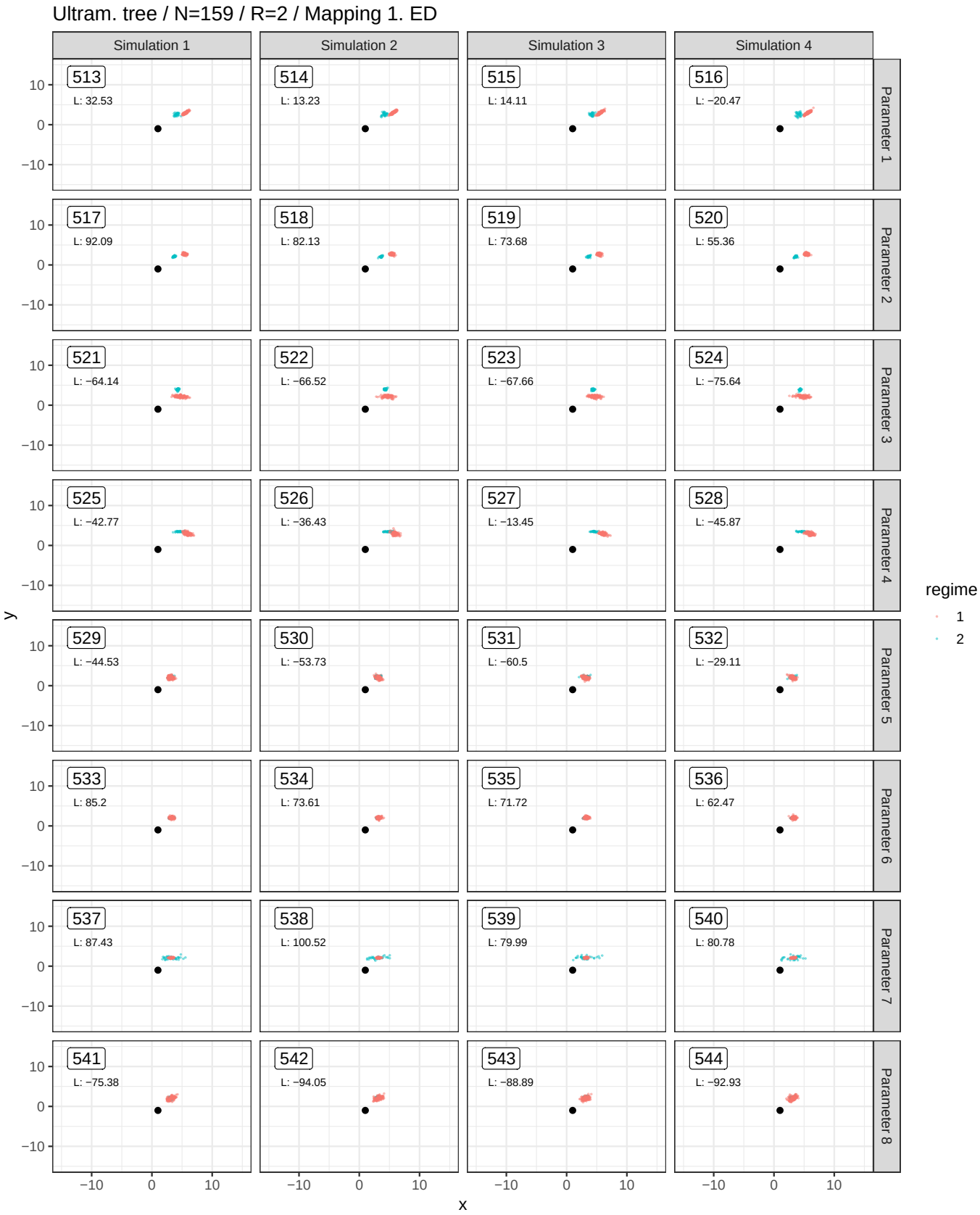


Fig. S38. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

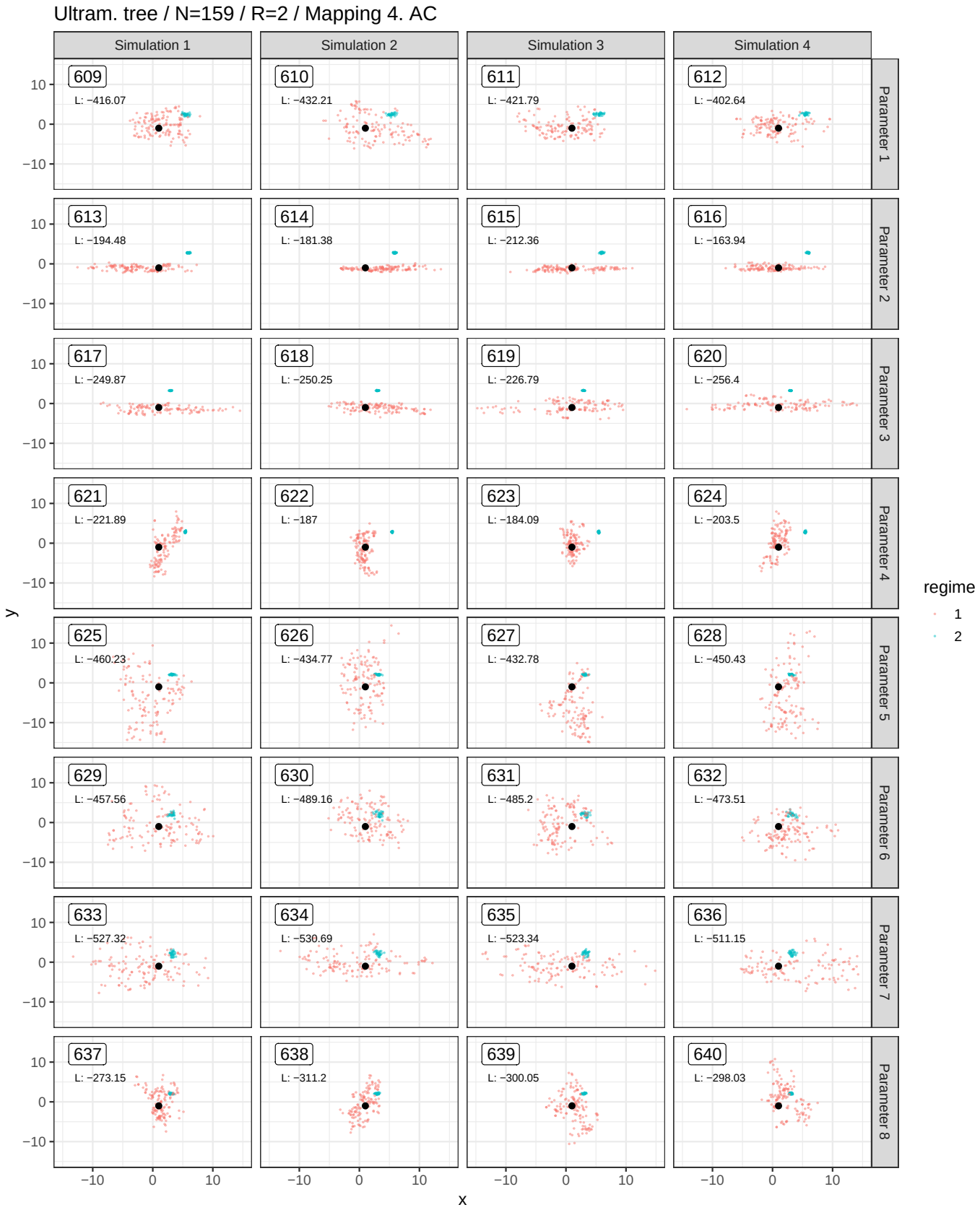


Fig. S39. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

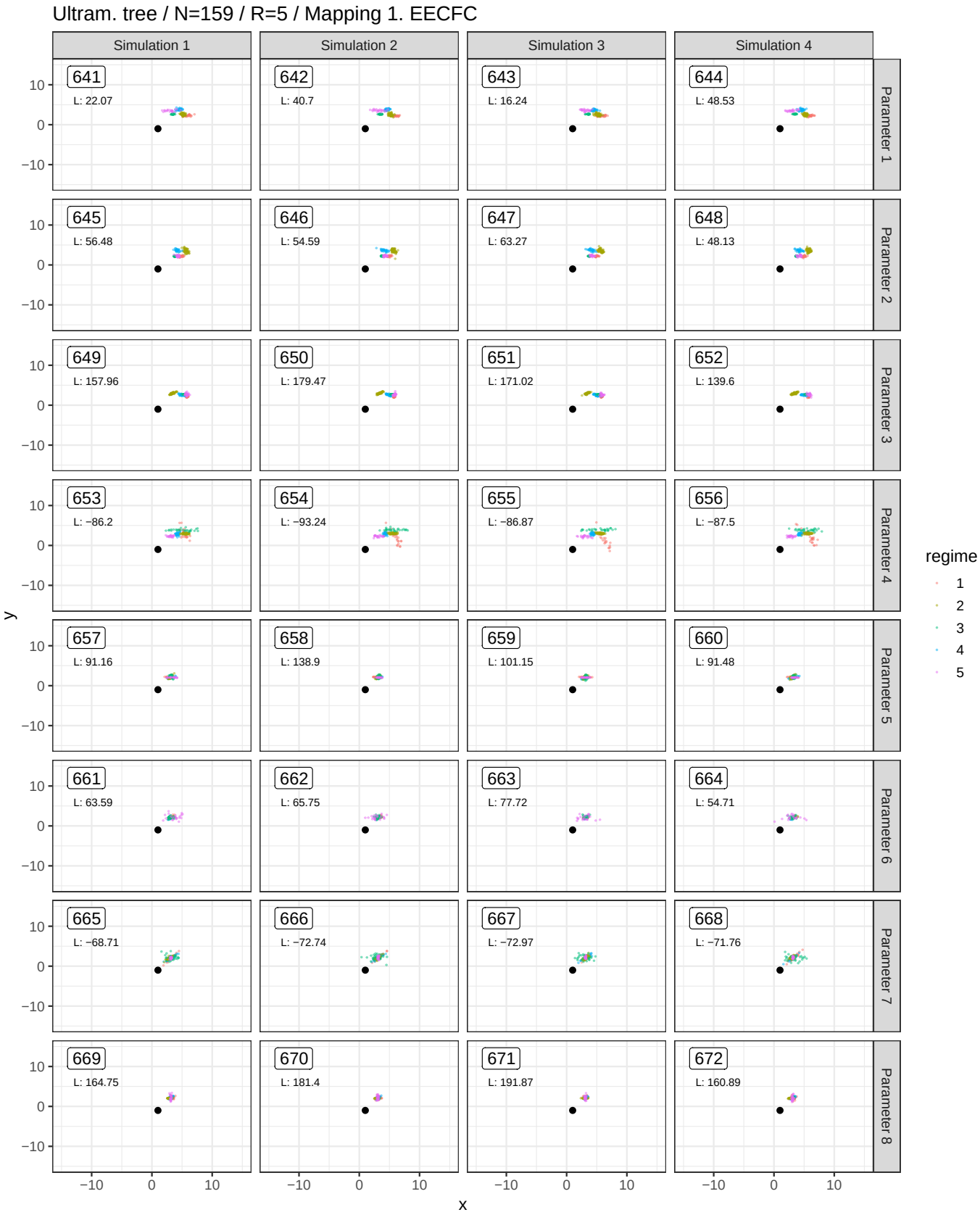


Fig. S40. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

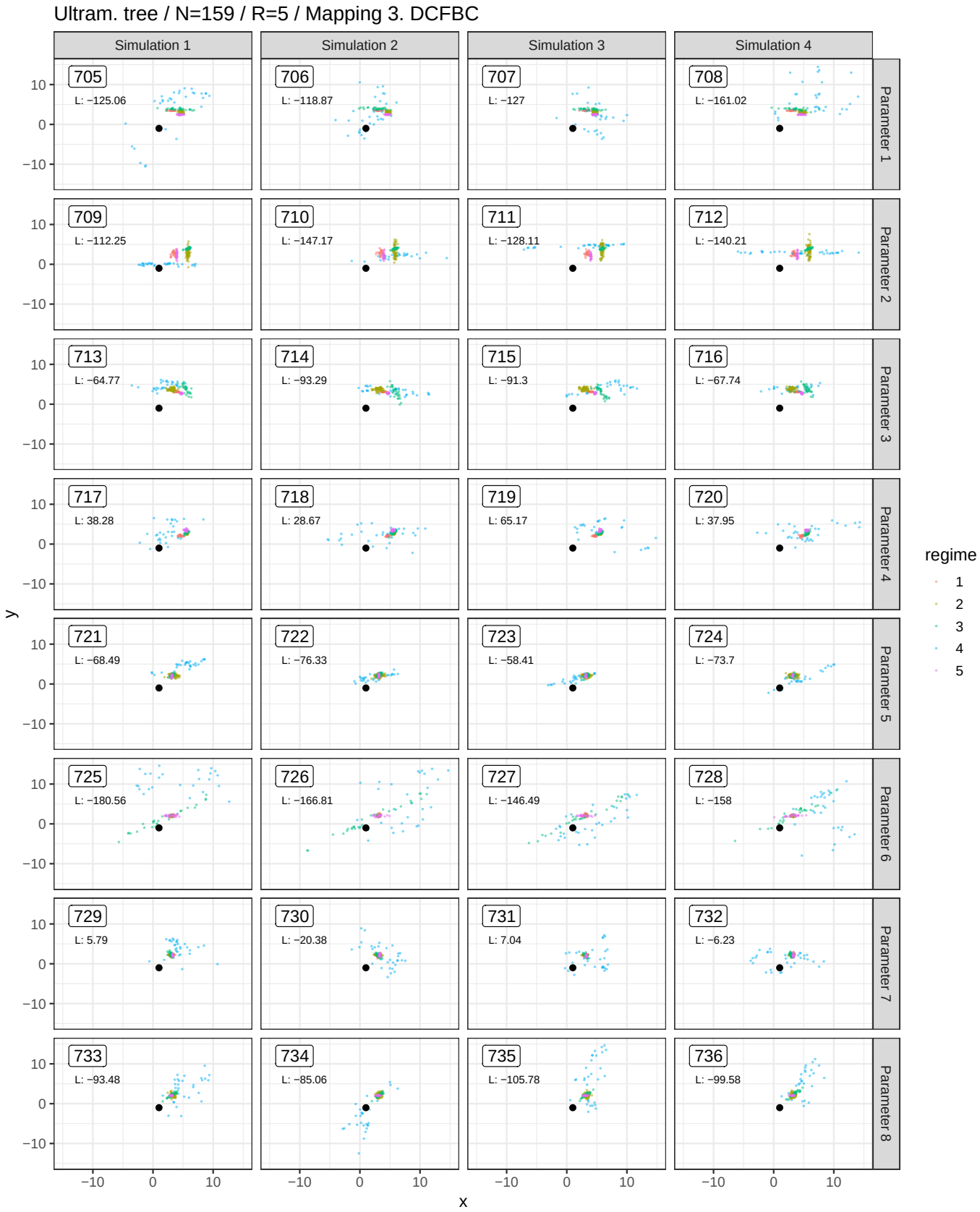


Fig. S41. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

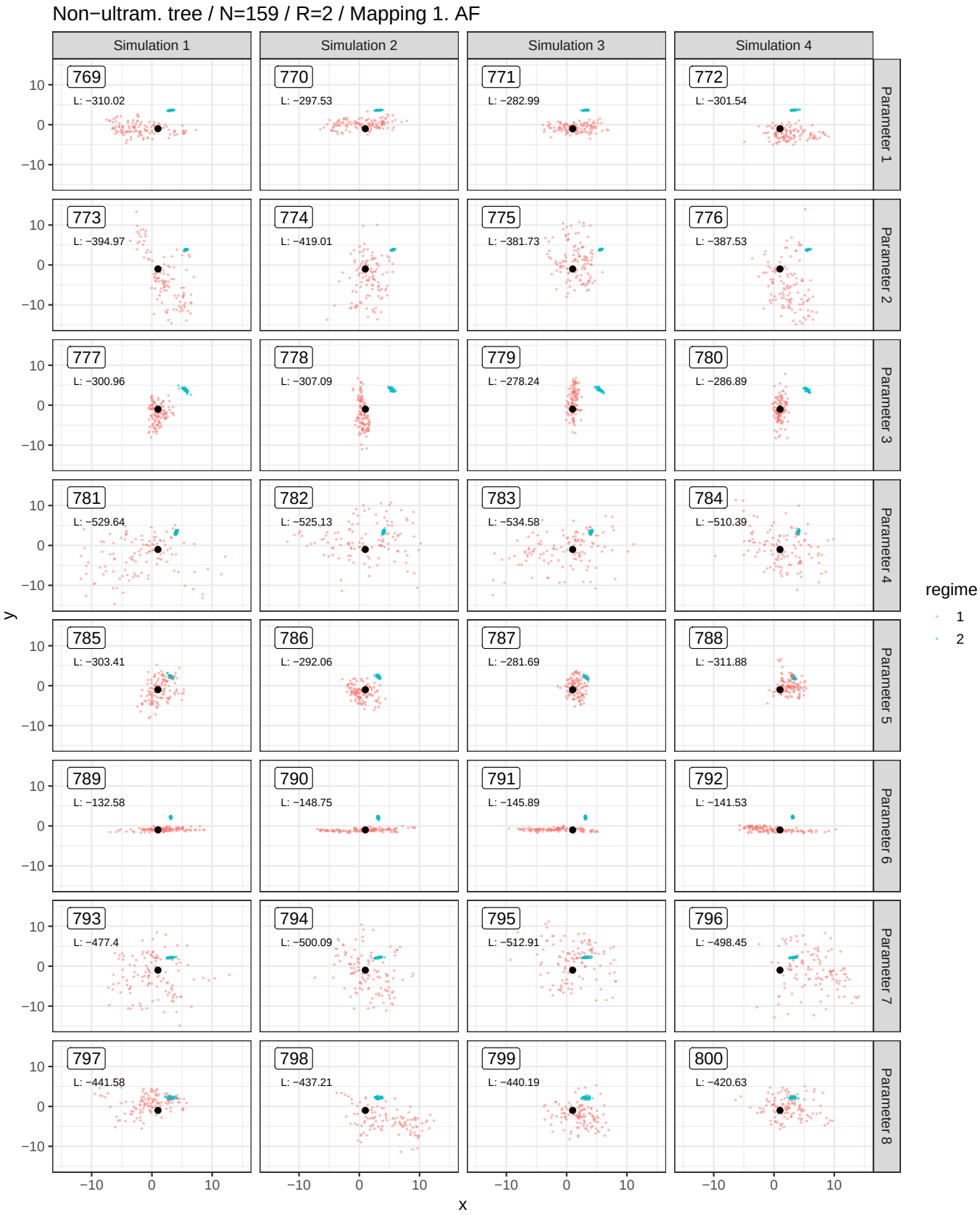


Fig. S42. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

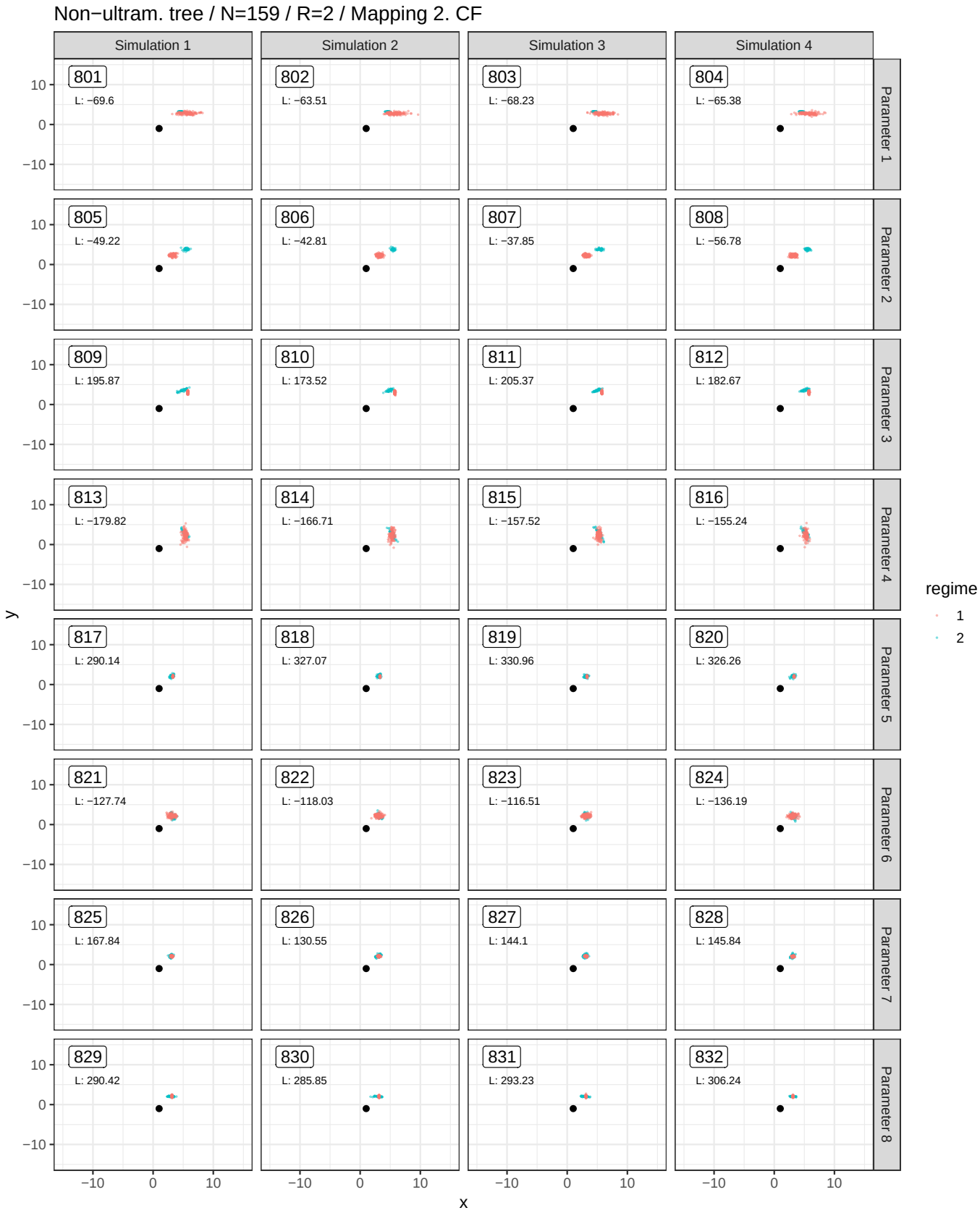


Fig. S43. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

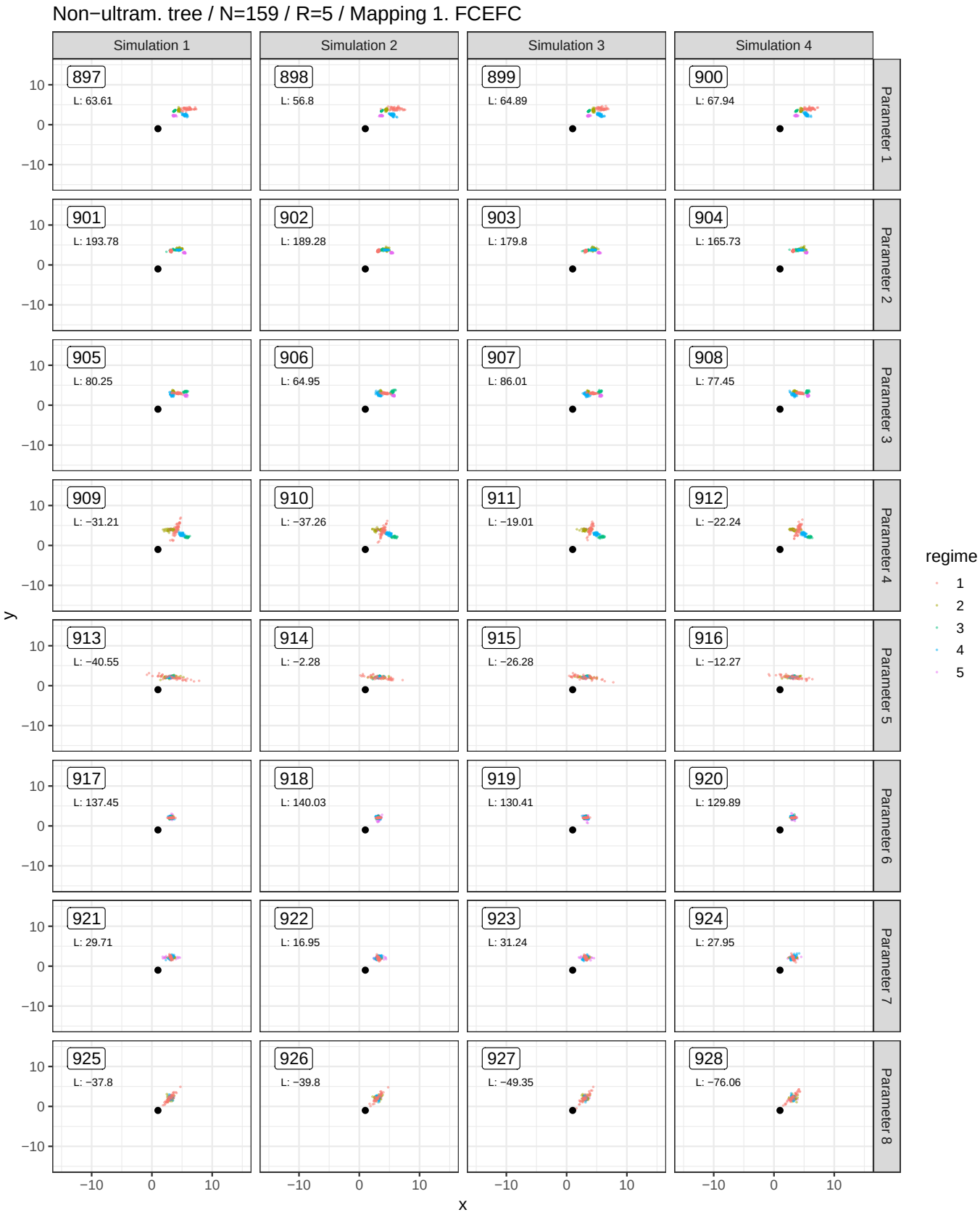


Fig. S44. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

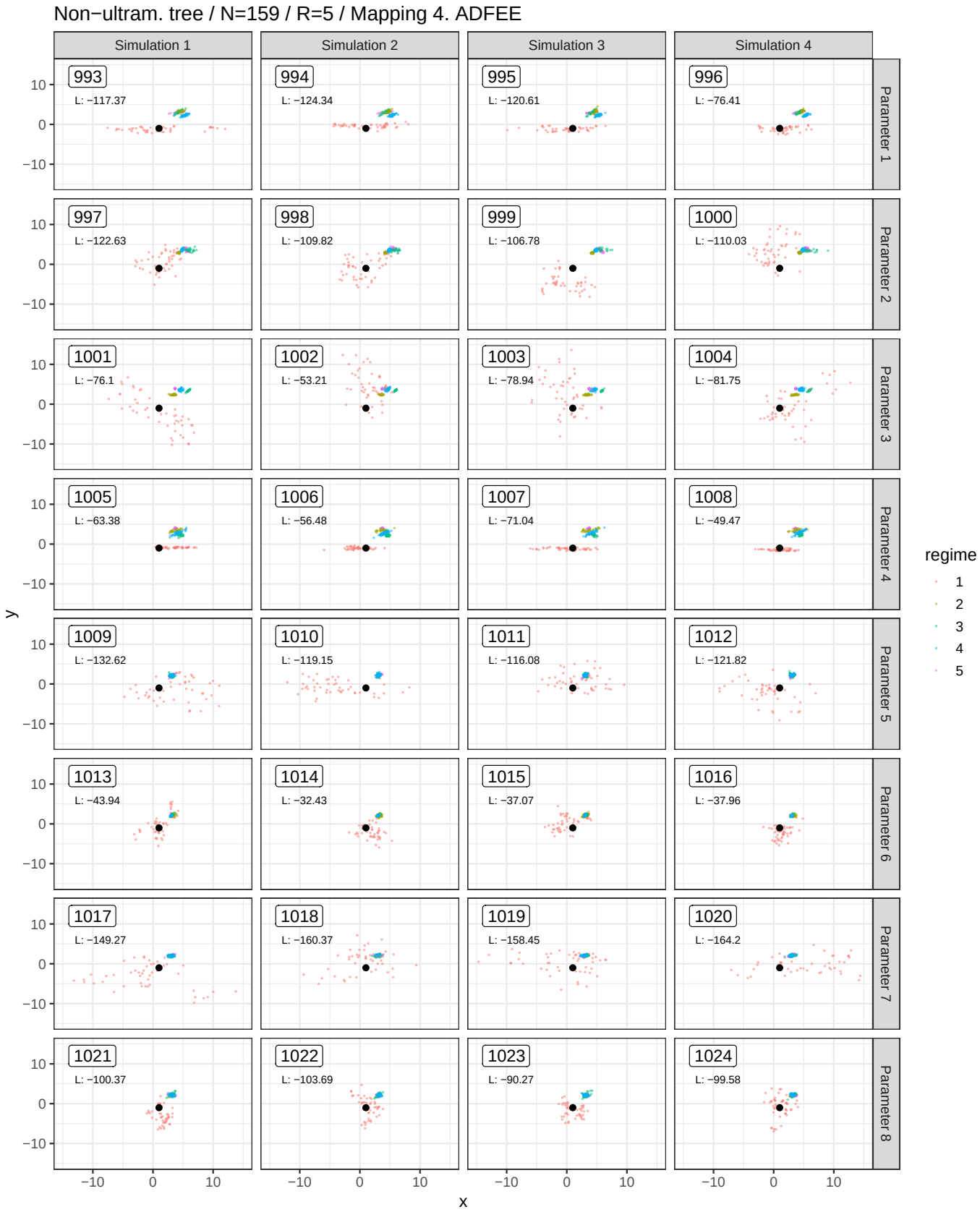


Fig. S45. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

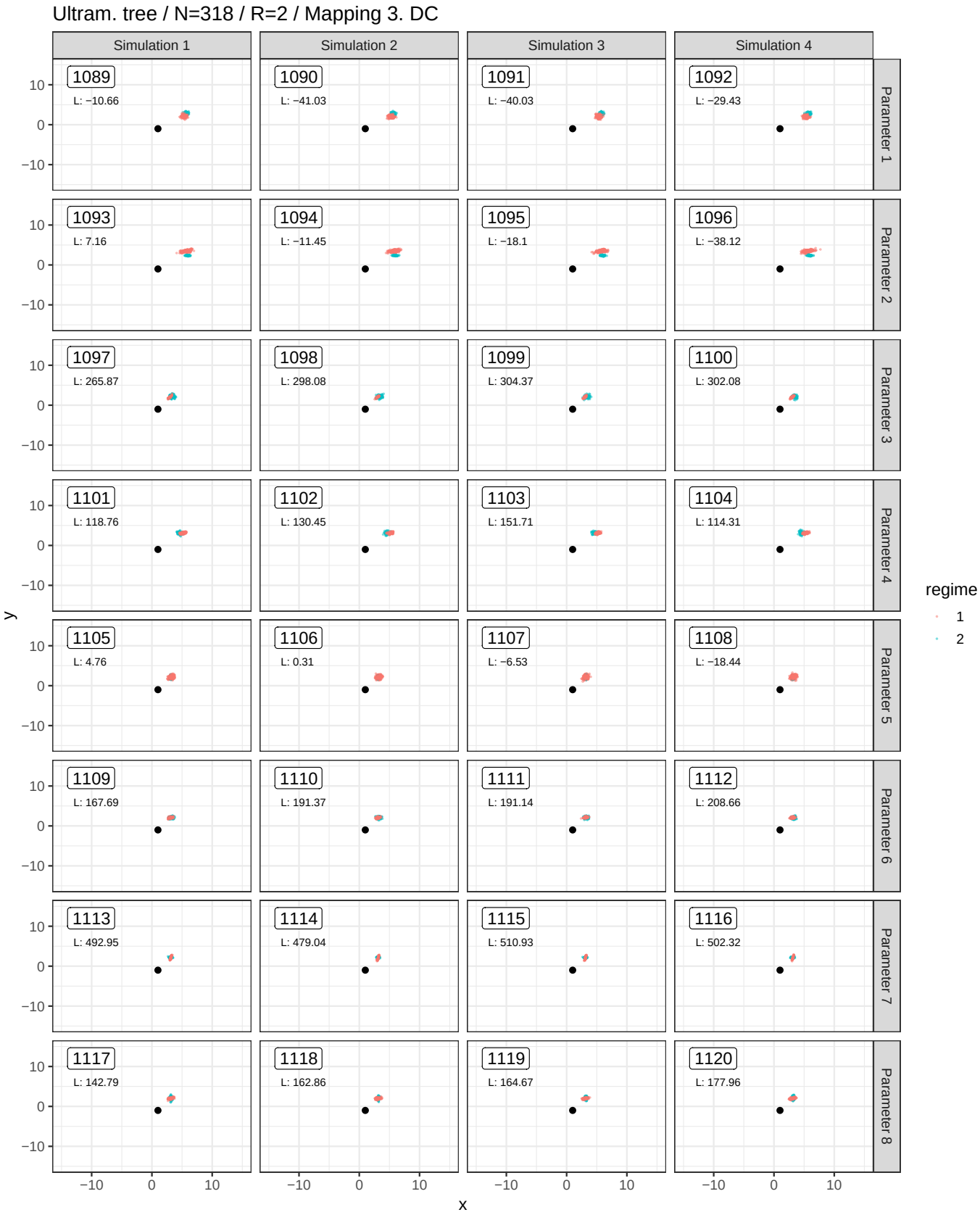


Fig. S46. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

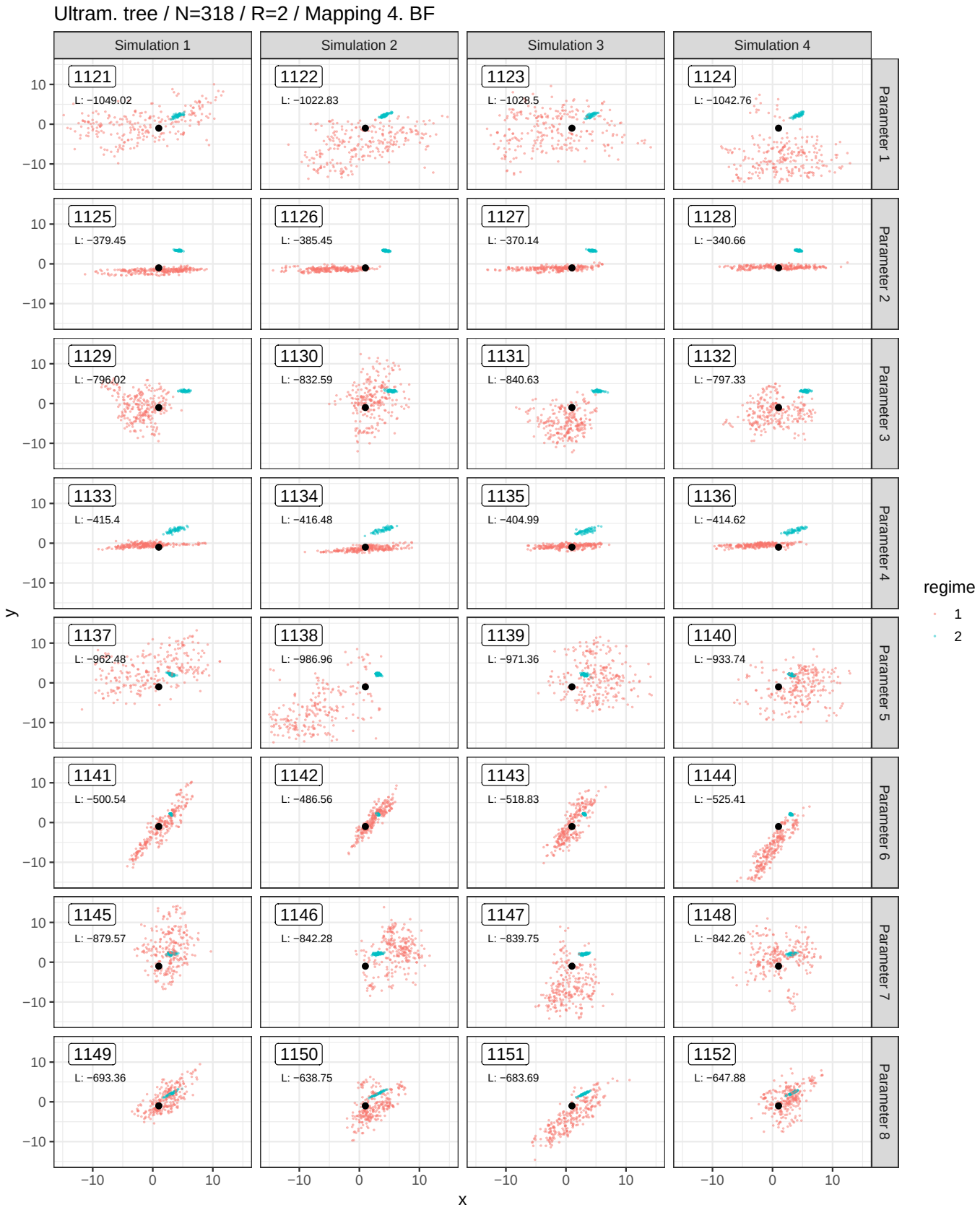


Fig. S47. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

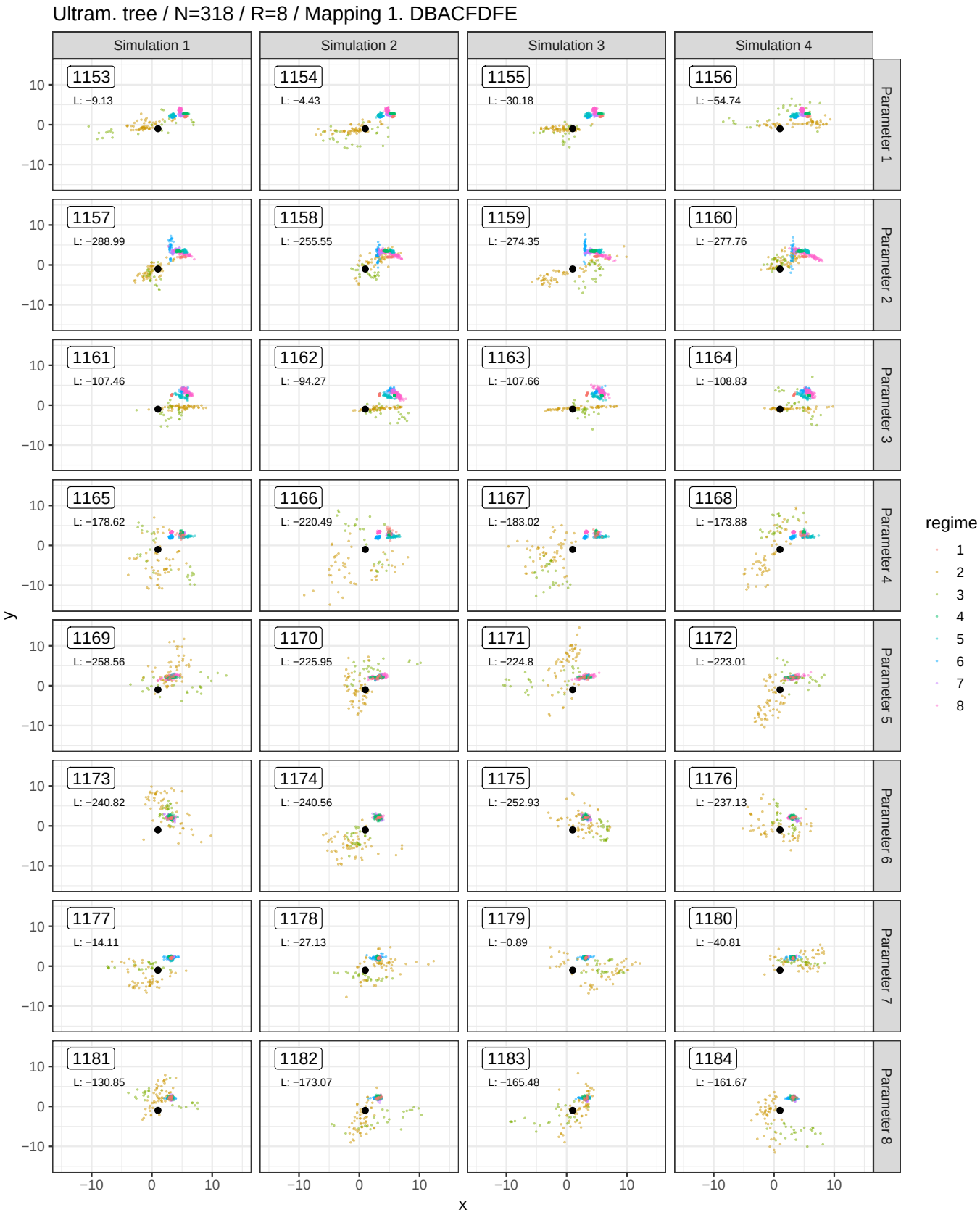


Fig. S48. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

Fig. S49. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

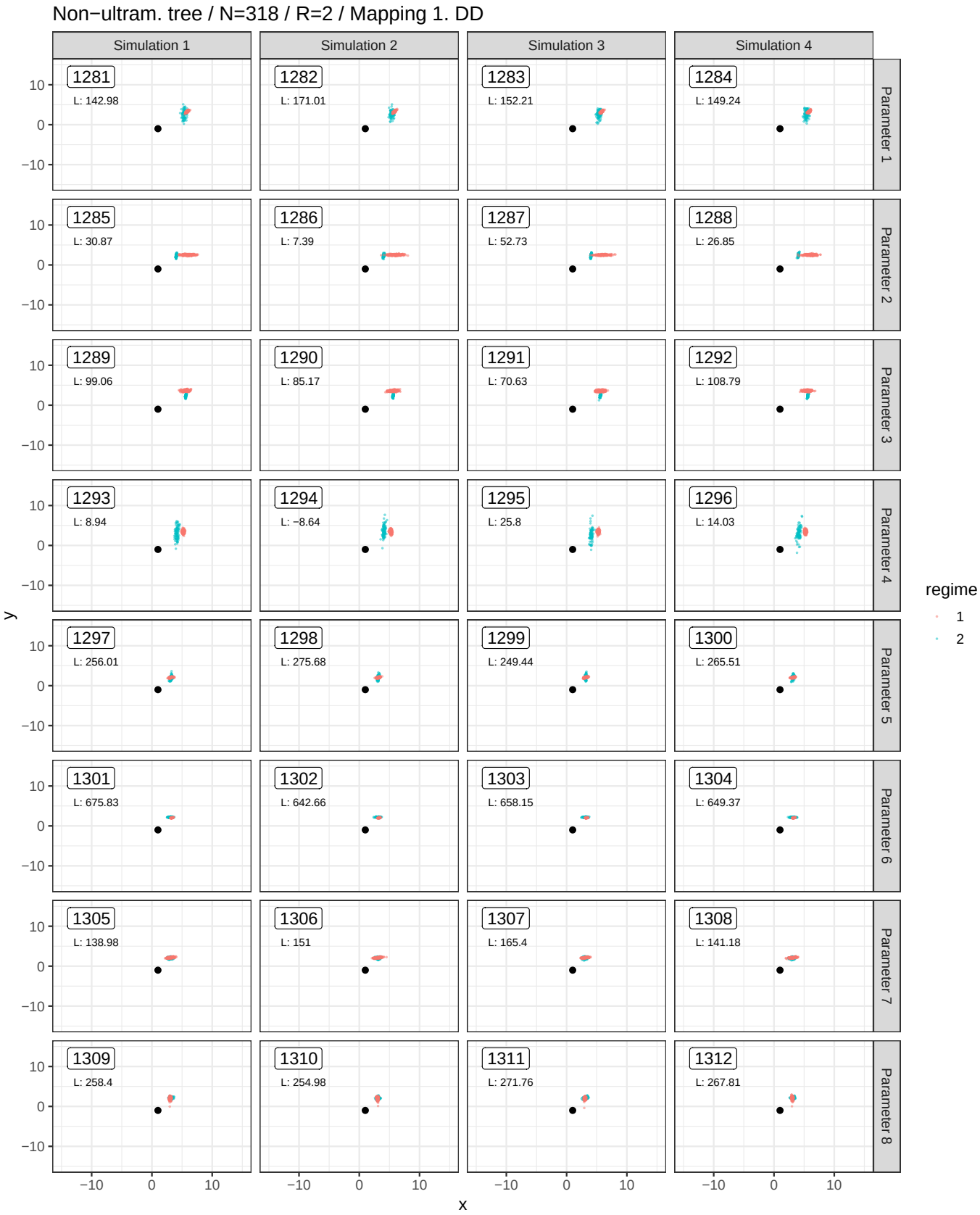


Fig. S50. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

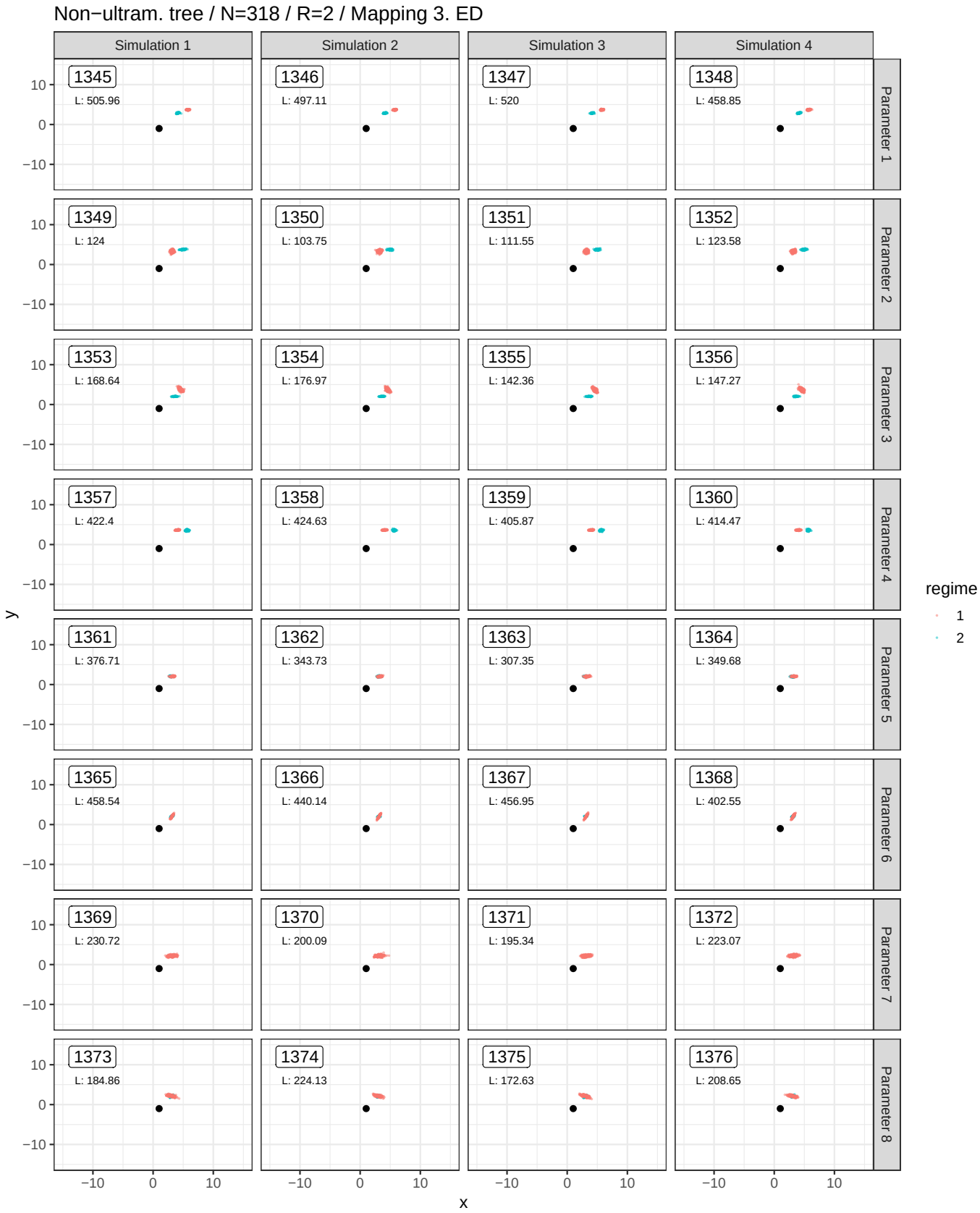


Fig. S51. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

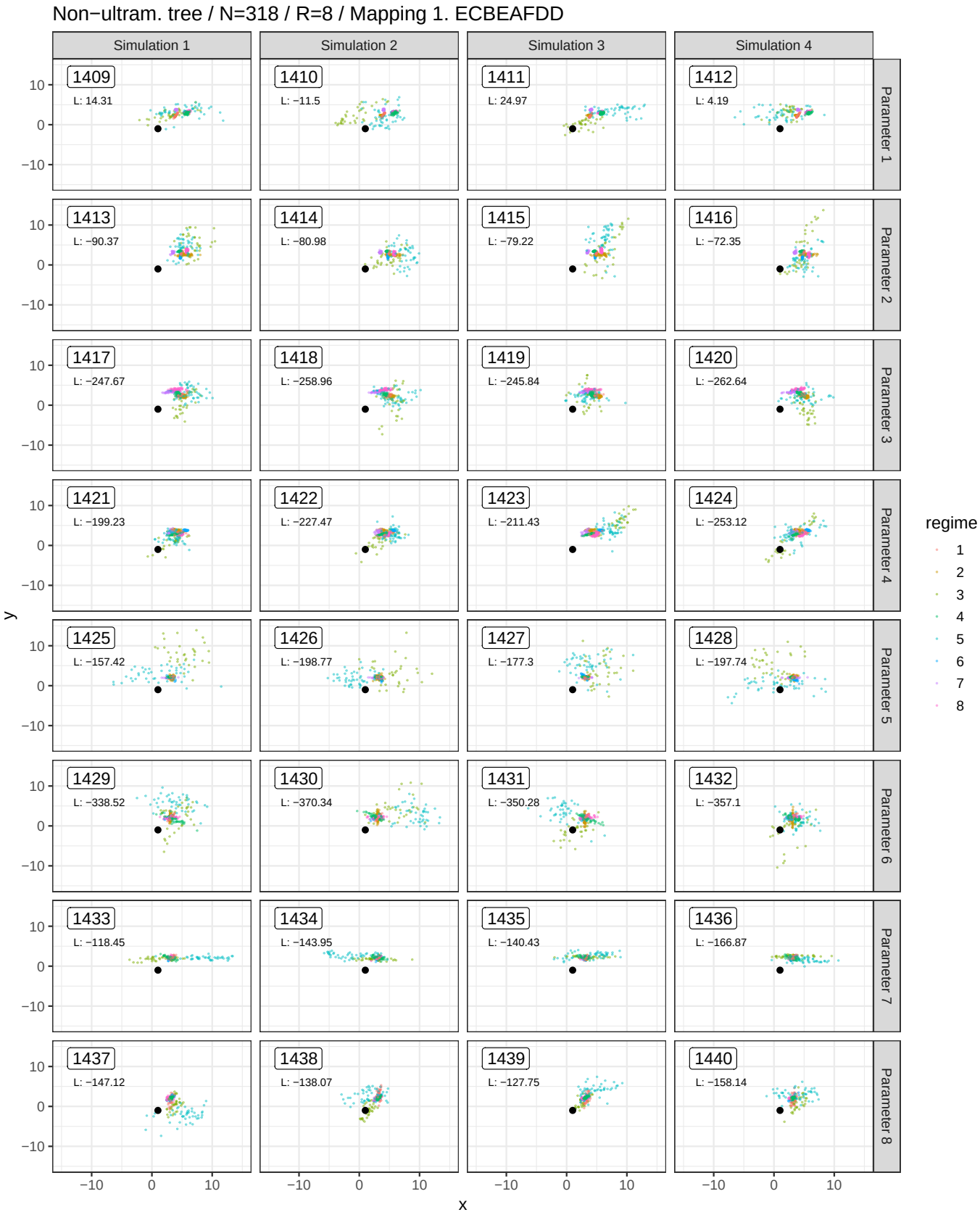


Fig. S52. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

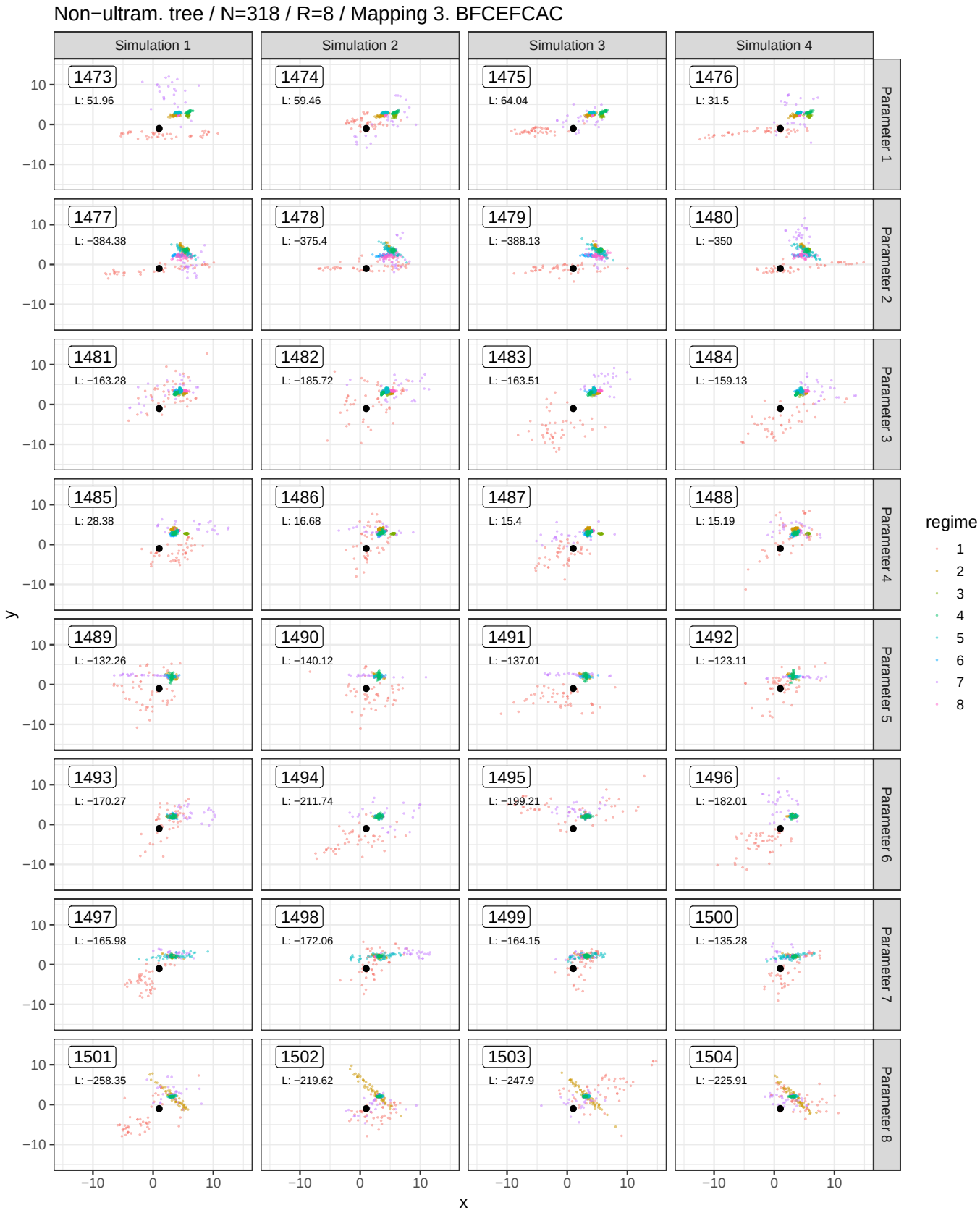


Fig. S53. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

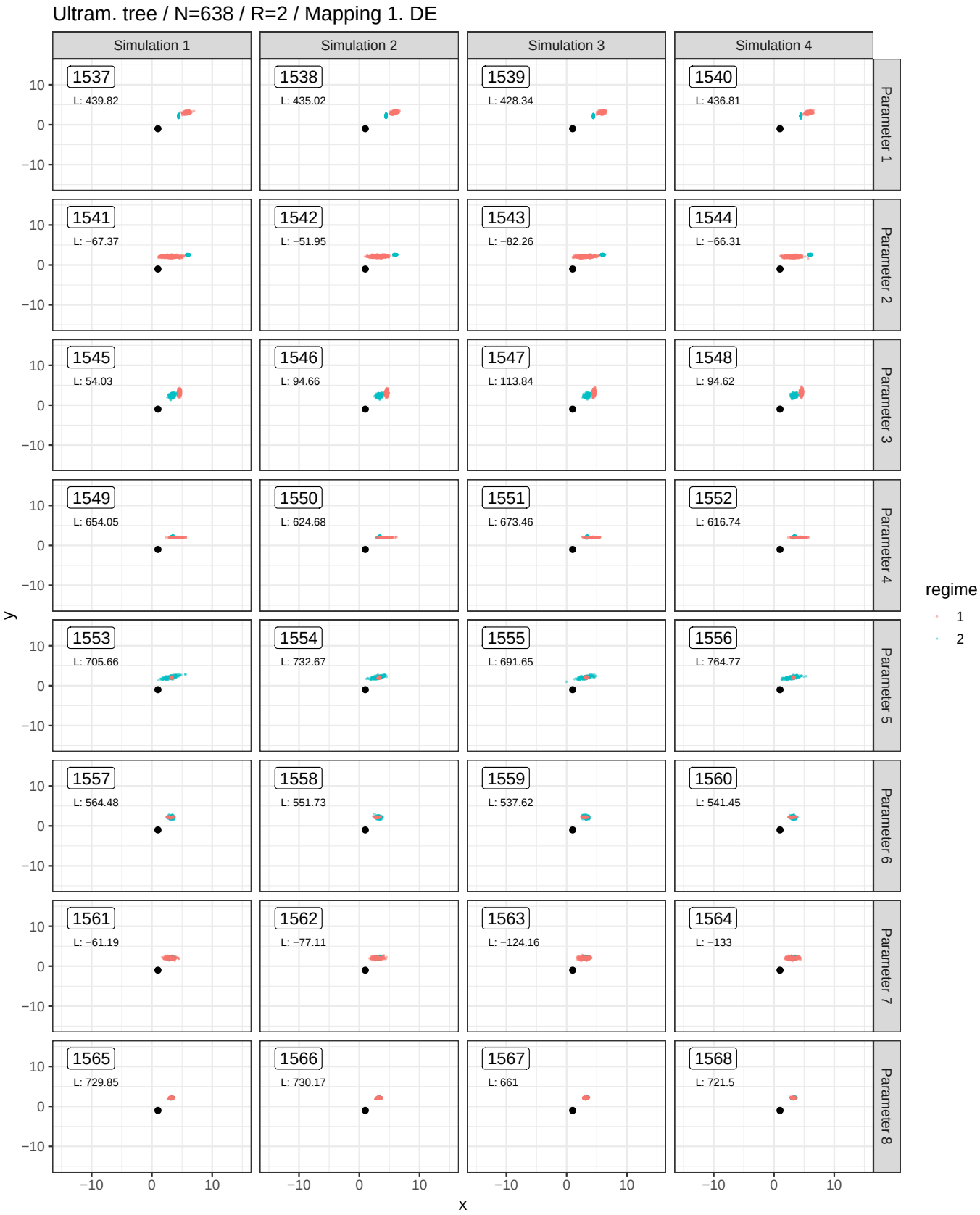


Fig. S54. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

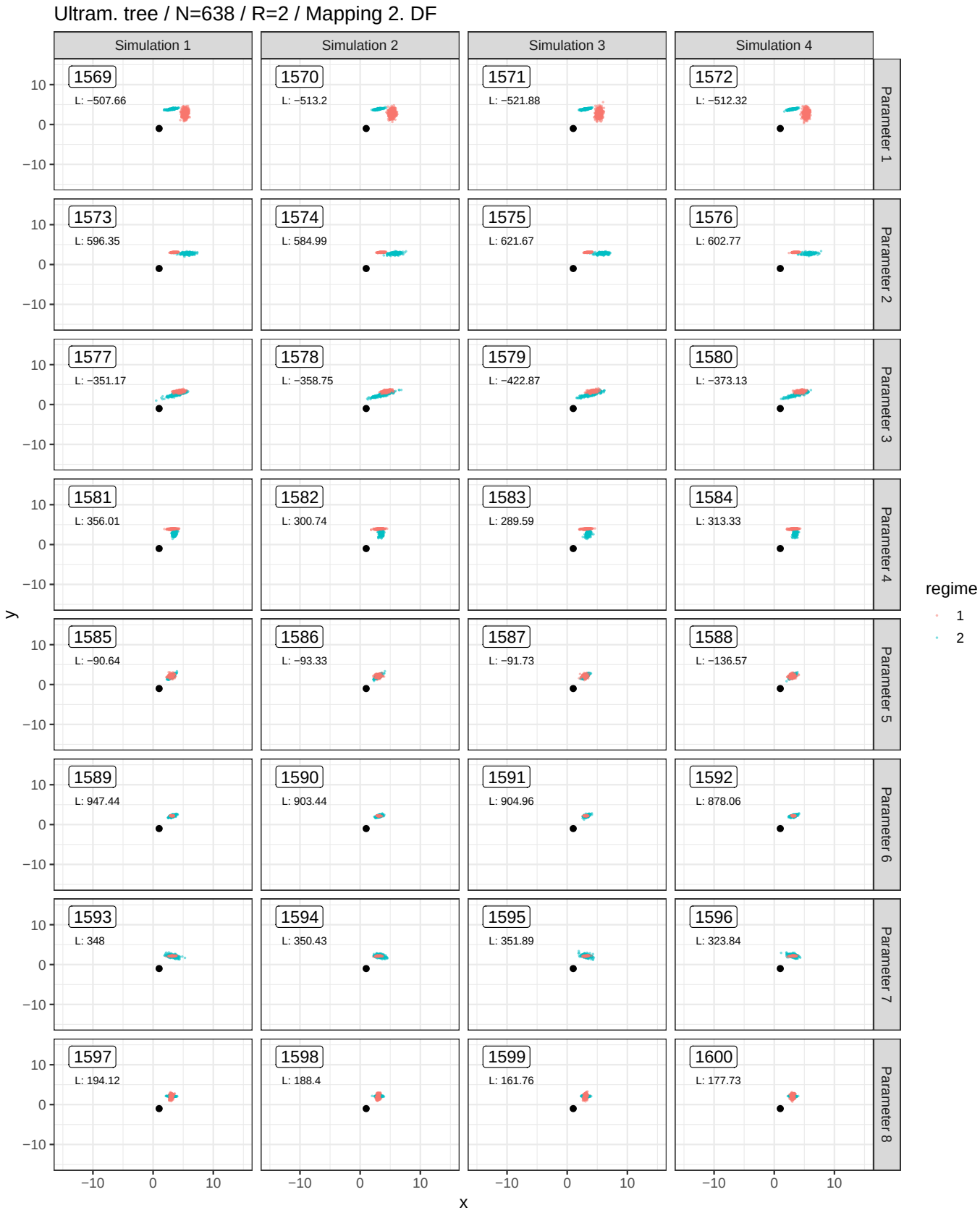


Fig. S55. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

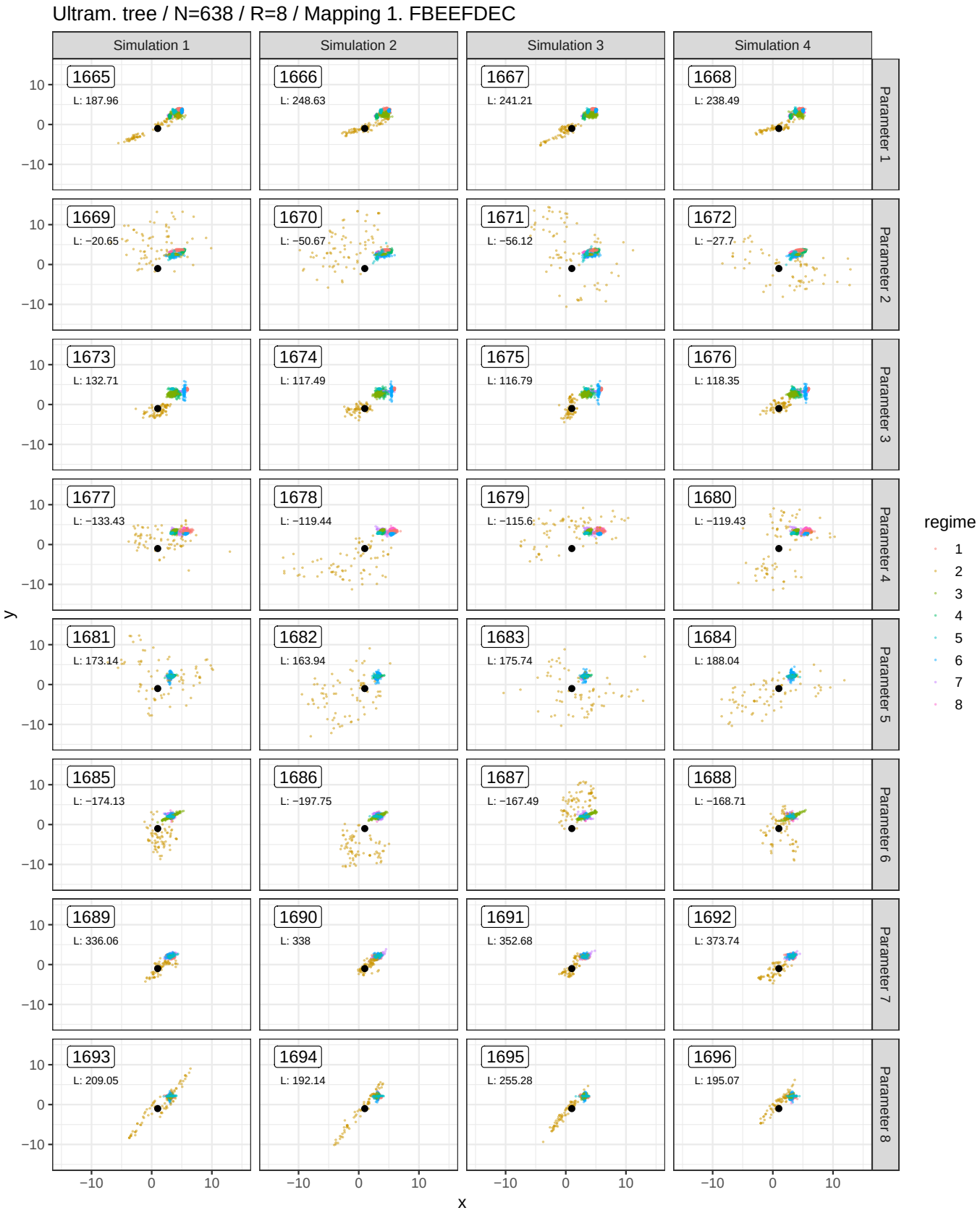


Fig. S56. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

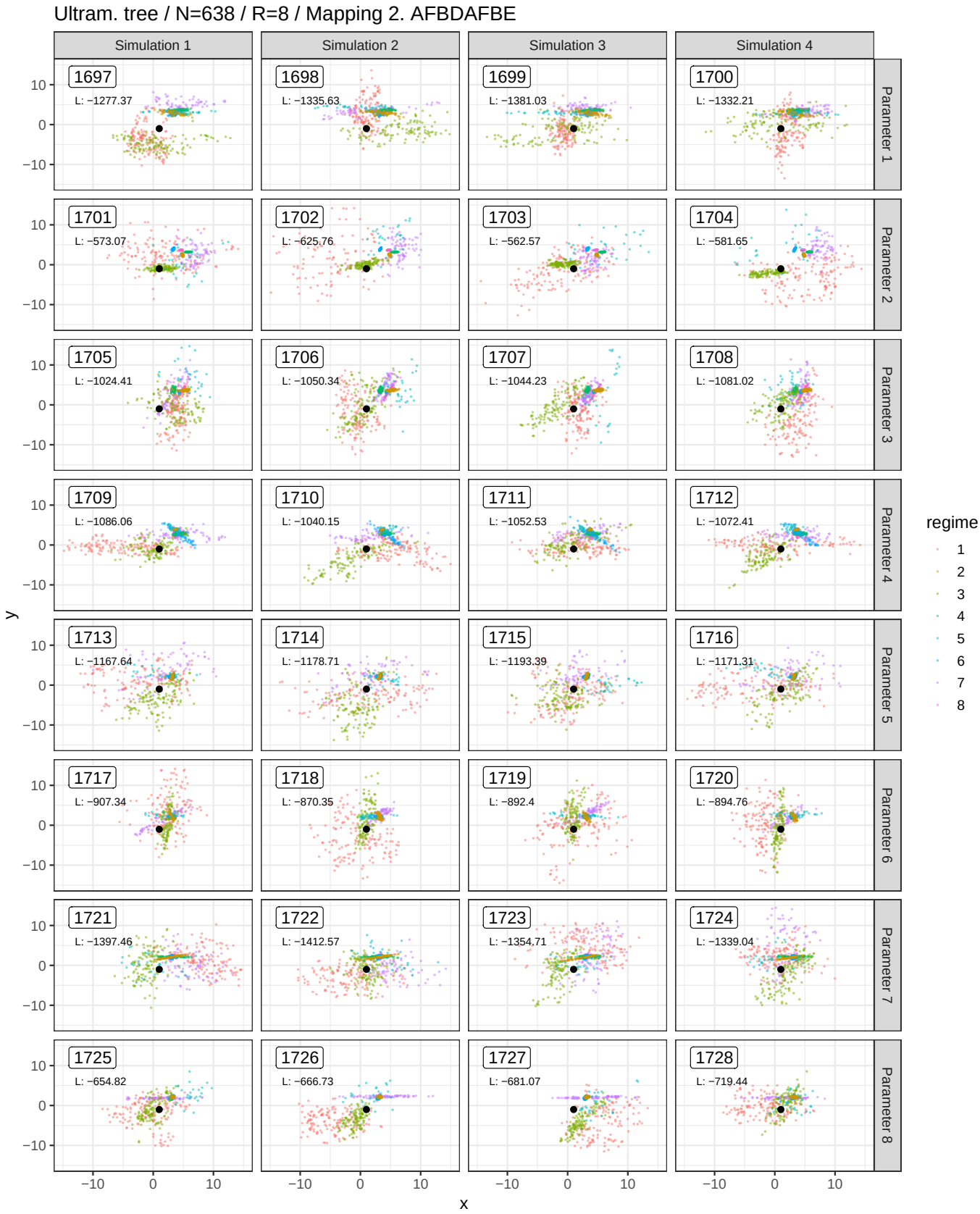


Fig. S57. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

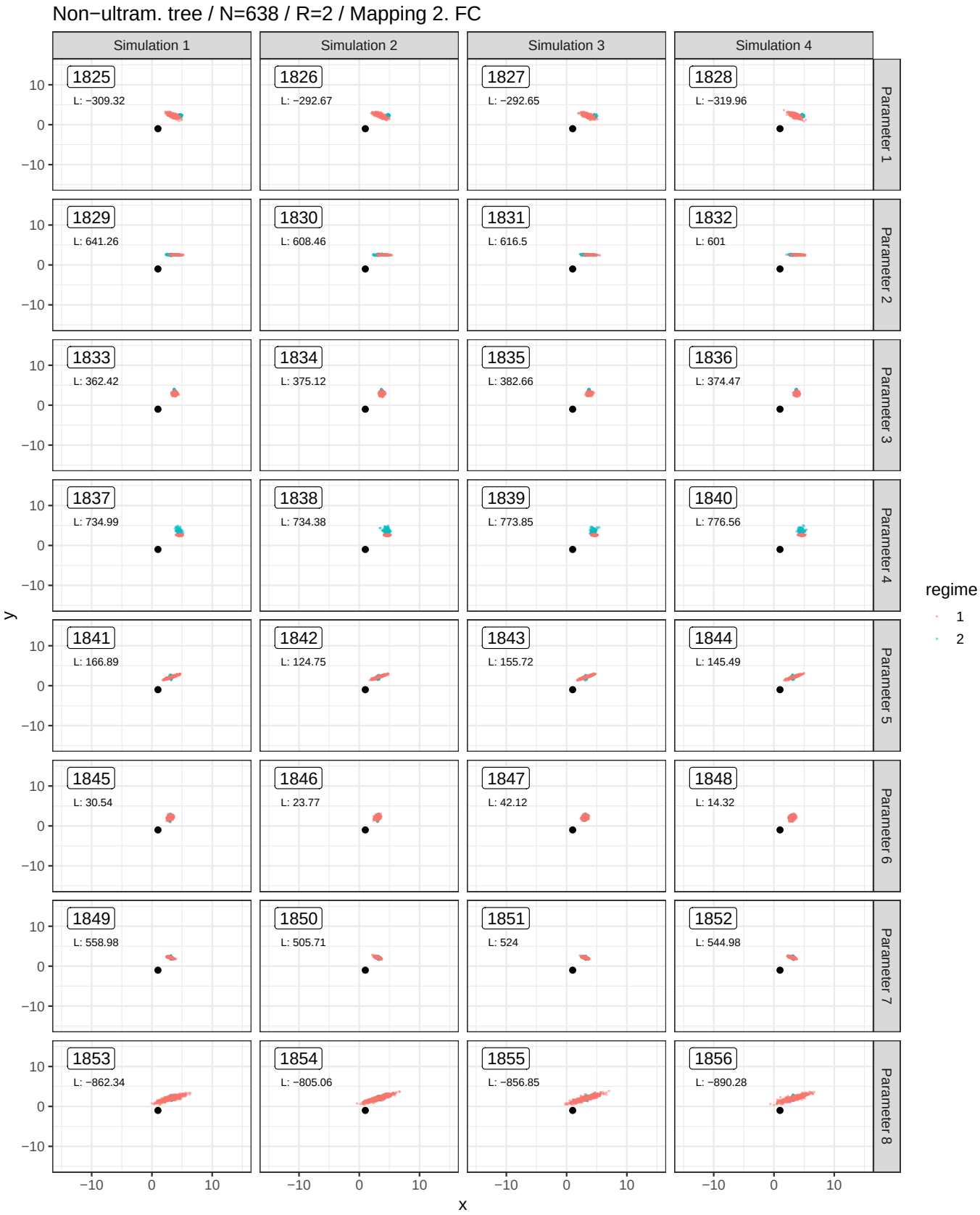


Fig. S58. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

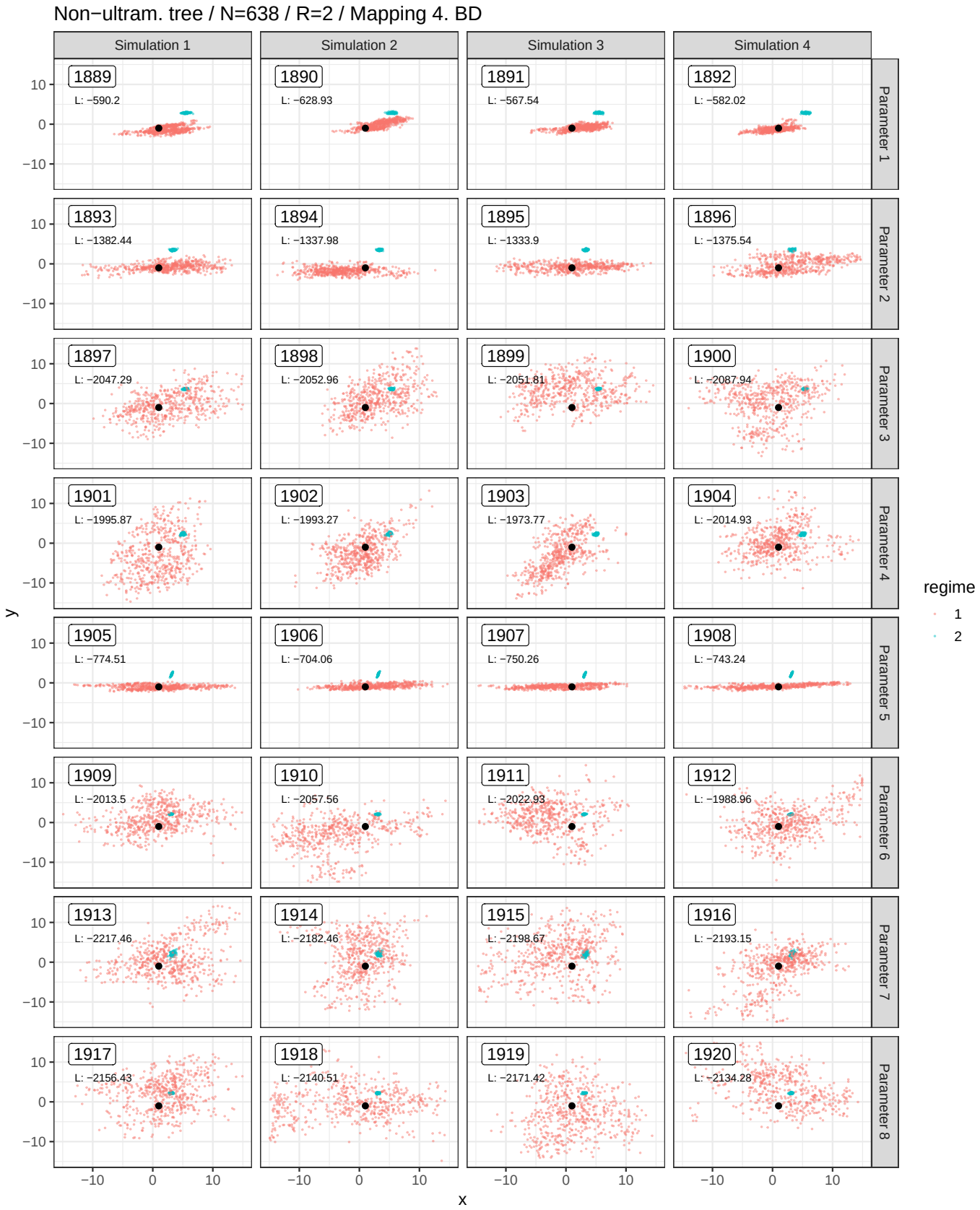


Fig. S59. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

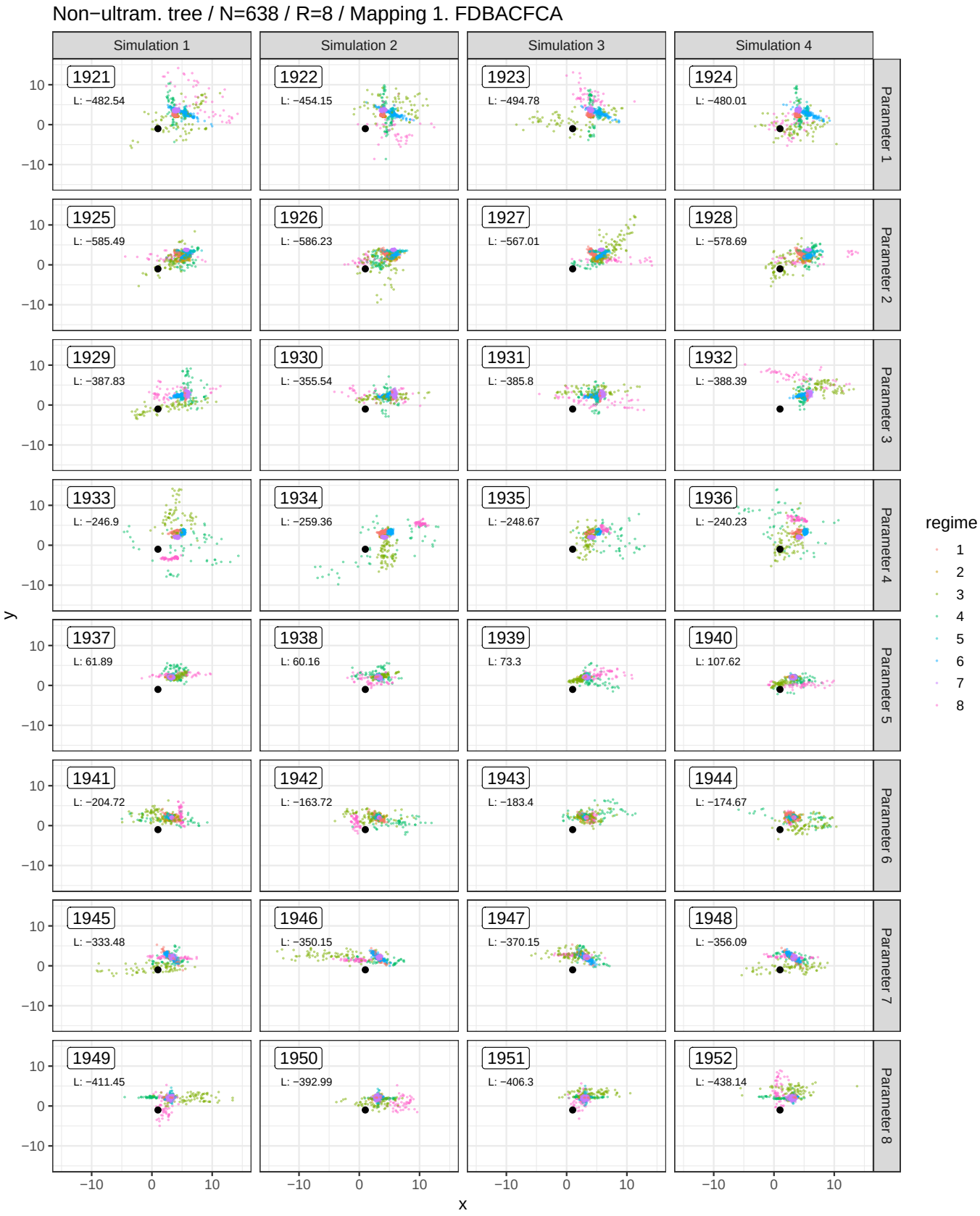


Fig. S60. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

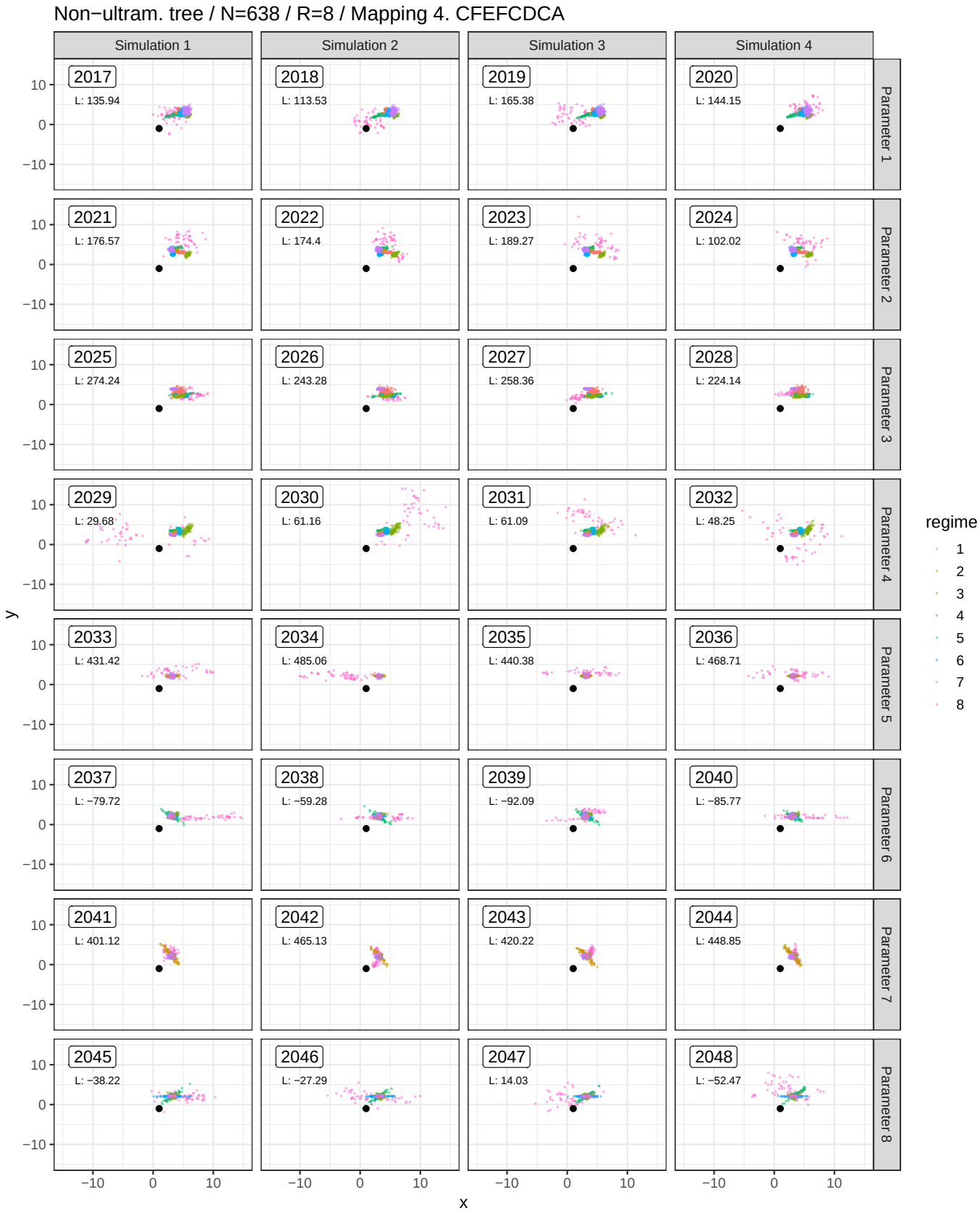


Fig. S61. Simulated datasets See figure title for the type, size, number of regimes and the model mapping. See also the legend for Fig. S30

L.3. Supplementary figure for evaluating the type I and II errors in single-regime simulations of BM_A and BM_B models described in SI Appendix, Section J.

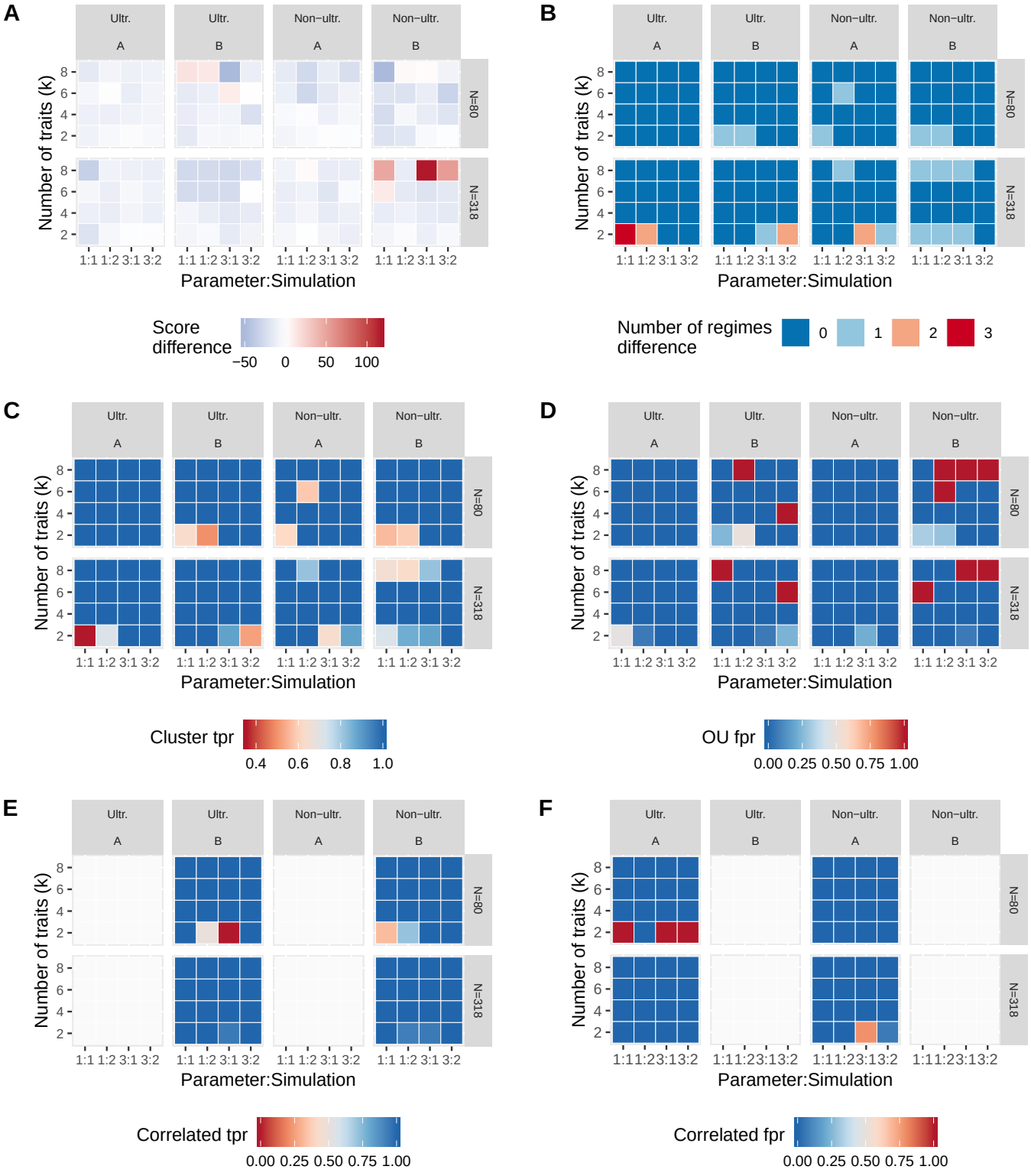


Fig. S62. Performance evaluation for simulations using a single-regime BM_A or BM_B model. The horizontal strip-labels denote the type of tree (Ultrametric vs, Non-ultrametric) and the type of simulated model (A stays for BM_A, B stays for BM_B). The vertical strip-labels denote the tree sizes. Each coloured square within each panel shows the value for one inferred model for one of the criteria defined in Appendix, Section J. The color coding has been set so that blue indicates "good" performance while red indicates "poor" performance. A: Criterion 1. ΔS; B: Criterion 4. ΔR; C: Criterion 5. Cluster; D: Criterion 6. OU process; E: Criterion 7. Correlated traits - true positive rate (models B only); F: Criterion 7. Correlated traits - false positive rate (models A only).

M. Supplementary Tables.

M.1. Inferred parameters of the model fits to the mammal data.

Table S1. Inferred parameters of model Global BM_A to body- and brain-mass data from 629 mammal species. Schur and upper triangular forms of the parameters $\mathbf{H}$ (OU models only), $\mathbf{\Sigma}$ are denoted by H_S and $\mathbf{\Sigma}_u$, respectively. The regime column denotes the scope of the parameters with :global: denoting the global scope, i.e. parameter values inherited by all regimes, and integer numbers denoting each of the model regimes. The type column denotes the model type associated with each regime. See also Table 1 in the main text for the log-likelihood and the AIC scores of the model.

regime	type	$X0$	Σ_u	Σ
:global:	NA	$\begin{bmatrix} 2.9367 \\ 0.8669 \end{bmatrix}$		
1	A		$\begin{bmatrix} 0.0928 & 0.0000 \\ 0.0000 & 0.0599 \end{bmatrix}$	$\begin{bmatrix} 0.0086 & 0.0000 \\ 0.0000 & 0.0036 \end{bmatrix}$

Table S2. Inferred parameters of model Global BM_B to body- and brain-mass data from 629 mammal species. Schur and upper triangular forms of the parameters $\mathbf{H}$ (OU models only), $\mathbf{\Sigma}$ are denoted by H_S and $\mathbf{\Sigma}_u$, respectively. The regime column denotes the scope of the parameters with :global: denoting the global scope, i.e. parameter values inherited by all regimes, and integer numbers denoting each of the model regimes. The type column denotes the model type associated with each regime. See also Table 1 in the main text for the log-likelihood and the AIC scores of the model.

regime	type	$X0$	Σ_u	Σ
:global:	NA	$\begin{bmatrix} 2.9440 \\ 0.8722 \end{bmatrix}$		
1	B		$\begin{bmatrix} 0.0318 & 0.0859 \\ 0.0000 & 0.0601 \end{bmatrix}$	$\begin{bmatrix} 0.0084 & 0.0052 \\ 0.0052 & 0.0036 \end{bmatrix}$

Table S3. Inferred parameters of model Global OU_C to body- and brain-mass data from 629 mammal species. Schur and upper triangular forms of the parameters $\mathbf{H}$ (OU models only), $\mathbf{\Sigma}$ are denoted by H_S and $\mathbf{\Sigma}_u$, respectively. The regime column denotes the scope of the parameters with :global: denoting the global scope, i.e. parameter values inherited by all regimes, and integer numbers denoting each of the model regimes. The type column denotes the model type associated with each regime. See also Table 1 in the main text for the log-likelihood and the AIC scores of the model.

regime	type	$X0$	H_S	H	Θ	Σ_u	Σ
:global:	NA	$\begin{bmatrix} 2.9367 \\ 0.8690 \end{bmatrix}$					
1	C		$\begin{bmatrix} 0.0000 & 0.0000 \\ 0.0000 & 0.0000 \end{bmatrix}$	$\begin{bmatrix} 0.0000 & 0.0000 \\ 0.0000 & 0.0000 \end{bmatrix}$	$\begin{bmatrix} 0.7019 \\ 1.1802 \end{bmatrix}$	$\begin{bmatrix} 0.0928 & 0.0000 \\ 0.0000 & 0.0599 \end{bmatrix}$	$\begin{bmatrix} 0.0086 & 0.0000 \\ 0.0000 & 0.0036 \end{bmatrix}$

Table S4. Inferred parameters of model Global OU_D to body- and brain-mass data from 629 mammal species. Schur and upper triangular forms of the parameters $\mathbf{H}$ (OU models only), $\mathbf{\Sigma}$ are denoted by H_S and $\mathbf{\Sigma}_u$, respectively. The regime column denotes the scope of the parameters with :global: denoting the global scope, i.e. parameter values inherited by all regimes, and integer numbers denoting each of the model regimes. The type column denotes the model type associated with each regime. See also Table 1 in the main text for the log-likelihood and the AIC scores of the model.

regime	type	$X0$	H_S	H	Θ	Σ_u	Σ
:global:	NA	$\begin{bmatrix} 2.9440 \\ 0.8722 \end{bmatrix}$					
1	D		$\begin{bmatrix} 0.0000 & 0.0000 \\ 0.0000 & 0.0000 \end{bmatrix}$	$\begin{bmatrix} 0.0000 & 0.0000 \\ 0.0000 & 0.0000 \end{bmatrix}$	$\begin{bmatrix} 0.0000 \\ 0.0000 \end{bmatrix}$	$\begin{bmatrix} 0.0318 & 0.0859 \\ 0.0000 & 0.0601 \end{bmatrix}$	$\begin{bmatrix} 0.0084 & 0.0052 \\ 0.0052 & 0.0036 \end{bmatrix}$

Table S5. Inferred parameters of model Global OU_E to body- and brain-mass data from 629 mammal species. Schur and upper triangular forms of the parameters $\mathbf{H}$ (OU models only), $\mathbf{\Sigma}$ are denoted by H_S and $\mathbf{\Sigma}_u$, respectively. The regime column denotes the scope of the parameters with :global: denoting the global scope, i.e. parameter values inherited by all regimes, and integer numbers denoting each of the model regimes. The type column denotes the model type associated with each regime. See also Table 1 in the main text for the log-likelihood and the AIC scores of the model.

regime	type	$X0$	H_S	H	Θ	Σ_u	Σ
:global:	NA	$\begin{bmatrix} 3.0844 \\ 0.9559 \end{bmatrix}$					
1	E		$\begin{bmatrix} 0.0000 & -1.0886 \\ 0.0000 & 0.0085 \end{bmatrix}$	$\begin{bmatrix} 0.0045 & -0.0042 \\ -0.0042 & 0.0040 \end{bmatrix}$	$\begin{bmatrix} 2.3956 \\ 0.3382 \end{bmatrix}$	$\begin{bmatrix} 0.0378 & 0.0872 \\ 0.0000 & 0.0561 \end{bmatrix}$	$\begin{bmatrix} 0.0090 & 0.0049 \\ 0.0049 & 0.0031 \end{bmatrix}$

Table S6. Inferred parameters of model Global OU_F to body- and brain-mass data from 629 mammal species. Schur and upper triangular forms of the parameters $\mathbf{H}$ (OU models only), $\mathbf{\Sigma}$ are denoted by H_S and $\mathbf{\Sigma}_u$, respectively. The regime column denotes the scope of the parameters with :global: denoting the global scope, i.e. parameter values inherited by all regimes, and integer numbers denoting each of the model regimes. The type column denotes the model type associated with each regime. See also Table 1 in the main text for the log-likelihood and the AIC scores of the model.

regime	type	$X0$	H_S	H	Θ	Σ_u	Σ
:global:	NA	$\begin{bmatrix} 3.1042 \\ 0.8688 \end{bmatrix}$					
1	F		$\begin{bmatrix} 0.0034 & -5.3024 \\ 0.0095 & 0.0000 \end{bmatrix}$	$\begin{bmatrix} -0.0021 & 0.0014 \\ -0.0080 & 0.0055 \end{bmatrix}$	$\begin{bmatrix} 2.4983 \\ 0.3376 \end{bmatrix}$	$\begin{bmatrix} 0.0375 & 0.0821 \\ 0.0000 & 0.0515 \end{bmatrix}$	$\begin{bmatrix} 0.0081 & 0.0042 \\ 0.0042 & 0.0026 \end{bmatrix}$

Table S7. Inferred parameters of model SURFACE OU to body- and brain-mass data from 629 mammal species. Schur and upper triangular forms of the parameters $\mathbf{H}$ (OU models only), $\mathbf{\Sigma}$ are denoted by H_S and $\mathbf{\Sigma}_u$, respectively. The regime column denotes the scope of the parameters with :global: denoting the global scope, i.e. parameter values inherited by all regimes, and integer numbers denoting each of the model regimes. The type column denotes the model type associated with each regime. See also Table 1 in the main text for the log-likelihood and the AIC scores of the model.

regime	type	$X0$	H_S	H	Σ_u	Σ	Θ
:global:	NA	$\begin{bmatrix} 2.9035 \\ 0.7674 \end{bmatrix}$	$\begin{bmatrix} 0.0000 & 0.0000 \\ 0.0000 & 0.0000 \end{bmatrix}$	$\begin{bmatrix} 0.0000 & 0.0000 \\ 0.0000 & 0.0000 \end{bmatrix}$	$\begin{bmatrix} 0.0928 & 0.0000 \\ 0.0000 & 0.0599 \end{bmatrix}$	$\begin{bmatrix} 0.0086 & 0.0000 \\ 0.0000 & 0.0036 \end{bmatrix}$	
1	SURFACE OU						$\begin{bmatrix} 2.4099 \\ 0.7810 \end{bmatrix}$

Table S8. Inferred parameters of model SCALAR OU to body- and brain-mass data from 629 mammal species. Schur and upper triangular forms of the parameters $\mathbf{H}$ (OU models only), $\mathbf{\Sigma}$ are denoted by H_S and $\mathbf{\Sigma}_u$, respectively. The regime column denotes the scope of the parameters with :global: denoting the global scope, i.e. parameter values inherited by all regimes, and integer numbers denoting each of the model regimes. The type column denotes the model type associated with each regime. See also Table 1 in the main text for the log-likelihood and the AIC scores of the model.

regime	type	$X0$	H_S	H	Θ	Σ_u	Σ
:global:	NA	$\begin{bmatrix} 2.9846 \\ 0.9033 \end{bmatrix}$	$\begin{bmatrix} 0.0000 & 0.0000 \\ 0.0000 & 0.0000 \end{bmatrix}$	$\begin{bmatrix} 0.0000 & 0.0000 \\ 0.0000 & 0.0000 \end{bmatrix}$			
1	SCALAR OU				$\begin{bmatrix} 5.2719 \\ 2.2507 \end{bmatrix}$	$\begin{bmatrix} 0.0274 & 0.0819 \\ 0.0000 & 0.0565 \end{bmatrix}$	$\begin{bmatrix} 0.0075 & 0.0046 \\ 0.0046 & 0.0032 \end{bmatrix}$
2	SCALAR OU				$\begin{bmatrix} 0.1737 \\ 0.0630 \end{bmatrix}$	$\begin{bmatrix} 0.0101 & 0.0749 \\ 0.0000 & 0.0626 \end{bmatrix}$	$\begin{bmatrix} 0.0057 & 0.0047 \\ 0.0047 & 0.0039 \end{bmatrix}$
3	SCALAR OU				$\begin{bmatrix} 1.8582 \\ 1.0944 \end{bmatrix}$	$\begin{bmatrix} 0.0437 & 0.1197 \\ 0.0000 & 0.0778 \end{bmatrix}$	$\begin{bmatrix} 0.0162 & 0.0093 \\ 0.0093 & 0.0060 \end{bmatrix}$
4	SCALAR OU				$\begin{bmatrix} 2.6962 \\ 0.7702 \end{bmatrix}$	$\begin{bmatrix} 0.0180 & 0.0601 \\ 0.0000 & 0.0386 \end{bmatrix}$	$\begin{bmatrix} 0.0039 & 0.0023 \\ 0.0023 & 0.0015 \end{bmatrix}$
5	SCALAR OU				$\begin{bmatrix} 5.2720 \\ 2.5832 \end{bmatrix}$	$\begin{bmatrix} 0.0410 & 0.0646 \\ 0.0000 & 0.0576 \end{bmatrix}$	$\begin{bmatrix} 0.0059 & 0.0037 \\ 0.0037 & 0.0033 \end{bmatrix}$
6	SCALAR OU				$\begin{bmatrix} 3.6879 \\ 1.0812 \end{bmatrix}$	$\begin{bmatrix} 0.0720 & 0.1076 \\ 0.0000 & 0.0720 \end{bmatrix}$	$\begin{bmatrix} 0.0168 & 0.0077 \\ 0.0077 & 0.0052 \end{bmatrix}$

Table S9. Inferred parameters of model RATEMATRIX BM (BM_B with shifts) to body- and brain-mass data from 629 mammal species. Schur and upper triangular forms of the parameters $\mathbf{H}$ (OU models only), $\mathbf{\Sigma}$ are denoted by H_S and $\mathbf{\Sigma}_u$, respectively. The regime column denotes the scope of the parameters with :global: denoting the global scope, i.e. parameter values inherited by all regimes, and integer numbers denoting each of the model regimes. The type column denotes the model type associated with each regime. See also Table 1 in the main text for the log-likelihood and the AIC scores of the model.

regime	type	$X0$	Σ_u	Σ
:global:	NA	$\begin{bmatrix} 3.4904 \\ 1.2142 \end{bmatrix}$		
1	B		$\begin{bmatrix} 0.0224 & 0.0743 \\ 0.0000 & 0.0518 \end{bmatrix}$	$\begin{bmatrix} 0.0060 & 0.0038 \\ 0.0038 & 0.0027 \end{bmatrix}$
2	B		$\begin{bmatrix} 0.0471 & 0.1392 \\ 0.0000 & 0.0926 \end{bmatrix}$	$\begin{bmatrix} 0.0216 & 0.0129 \\ 0.0129 & 0.0086 \end{bmatrix}$
3	B		$\begin{bmatrix} 0.0270 & 0.1062 \\ 0.0000 & 0.0664 \end{bmatrix}$	$\begin{bmatrix} 0.0120 & 0.0071 \\ 0.0071 & 0.0044 \end{bmatrix}$
4	B		$\begin{bmatrix} 0.0435 & 0.0778 \\ 0.0000 & 0.0728 \end{bmatrix}$	$\begin{bmatrix} 0.0079 & 0.0057 \\ 0.0057 & 0.0053 \end{bmatrix}$
5	B		$\begin{bmatrix} 0.0339 & 0.0991 \\ 0.0000 & 0.0727 \end{bmatrix}$	$\begin{bmatrix} 0.0110 & 0.0072 \\ 0.0072 & 0.0053 \end{bmatrix}$
6	B		$\begin{bmatrix} 0.0361 & 0.0431 \\ 0.0000 & 0.0285 \end{bmatrix}$	$\begin{bmatrix} 0.0032 & 0.0012 \\ 0.0012 & 0.0008 \end{bmatrix}$
7	B		$\begin{bmatrix} 0.0337 & 0.0658 \\ 0.0000 & 0.0473 \end{bmatrix}$	$\begin{bmatrix} 0.0055 & 0.0031 \\ 0.0031 & 0.0022 \end{bmatrix}$
8	B		$\begin{bmatrix} 0.0367 & 0.0647 \\ 0.0000 & 0.0561 \end{bmatrix}$	$\begin{bmatrix} 0.0055 & 0.0036 \\ 0.0036 & 0.0032 \end{bmatrix}$
9	B		$\begin{bmatrix} 0.0716 & 0.1082 \\ 0.0000 & 0.0723 \end{bmatrix}$	$\begin{bmatrix} 0.0168 & 0.0078 \\ 0.0078 & 0.0052 \end{bmatrix}$

Table S10. Inferred parameters of model MGPM*, i.e. the best MGPM (A-F) fit to body- and brain-mass data from 629 mammal species. Schur and upper triangular forms of the parameters $\mathbf{H}$ (OU models only), $\mathbf{\Sigma}$ are denoted by H_S and $\mathbf{\Sigma}_u$, respectively. The regime column denotes the scope of the parameters with :global: denoting the global scope, i.e. parameter values inherited by all regimes, and integer numbers denoting each of the model regimes. The type column denotes the model type associated with each regime. See also Table 1 in the main text for the log-likelihood and the AIC scores of the model.

regime	type	$X0$	H_S	H	Θ	Σ_u	Σ
:global:	NA	$\begin{bmatrix} 1.9062 \\ -0.9203 \end{bmatrix}$					
1	F		$\begin{bmatrix} 0.0233 & -2.9041 \\ 0.0586 & 0.0000 \end{bmatrix}$	$\begin{bmatrix} -0.0190 & 0.0219 \\ -0.0368 & 0.0423 \end{bmatrix}$	$\begin{bmatrix} 2.8272 \\ 0.8035 \end{bmatrix}$	$\begin{bmatrix} 0.0539 & 0.0719 \\ 0.0000 & 0.0445 \end{bmatrix}$	$\begin{bmatrix} 0.0081 & 0.0032 \\ 0.0032 & 0.0020 \end{bmatrix}$
2	E		$\begin{bmatrix} 0.0103 & 1.8517 \\ 0.0000 & 0.0000 \end{bmatrix}$	$\begin{bmatrix} 0.0053 & -0.0051 \\ -0.0051 & 0.0050 \end{bmatrix}$	$\begin{bmatrix} 0.6914 \\ -1.1544 \end{bmatrix}$	$\begin{bmatrix} 0.0282 & 0.0746 \\ 0.0000 & 0.0444 \end{bmatrix}$	$\begin{bmatrix} 0.0064 & 0.0033 \\ 0.0033 & 0.0020 \end{bmatrix}$
3	F		$\begin{bmatrix} 0.0459 & -0.8641 \\ -0.8132 & 0.0518 \end{bmatrix}$	$\begin{bmatrix} 0.4308 & -0.5461 \\ 0.2671 & -0.3331 \end{bmatrix}$	$\begin{bmatrix} 3.3797 \\ 1.6457 \end{bmatrix}$	$\begin{bmatrix} 0.0342 & 0.1551 \\ 0.0000 & 0.0649 \end{bmatrix}$	$\begin{bmatrix} 0.0252 & 0.0101 \\ 0.0101 & 0.0042 \end{bmatrix}$
4	E		$\begin{bmatrix} 1.3978 & 2.0122 \\ 0.0000 & 0.0432 \end{bmatrix}$	$\begin{bmatrix} 0.6380 & -0.6723 \\ -0.6723 & 0.8030 \end{bmatrix}$	$\begin{bmatrix} 1.0516 \\ -0.5143 \end{bmatrix}$	$\begin{bmatrix} 0.1142 & 0.0013 \\ 0.0000 & 0.0945 \end{bmatrix}$	$\begin{bmatrix} 0.0130 & 0.0001 \\ 0.0001 & 0.0089 \end{bmatrix}$
5	F		$\begin{bmatrix} 0.0022 & -0.0163 \\ 0.0098 & 0.0087 \end{bmatrix}$	$\begin{bmatrix} 0.0021 & 0.0097 \\ -0.0001 & 0.0088 \end{bmatrix}$	$\begin{bmatrix} 5.2387 \\ 3.8002 \end{bmatrix}$	$\begin{bmatrix} 0.0460 & 0.1445 \\ 0.0000 & 0.0894 \end{bmatrix}$	$\begin{bmatrix} 0.0230 & 0.0129 \\ 0.0129 & 0.0080 \end{bmatrix}$
6	D		$\begin{bmatrix} 1.8857 & 0.0000 \\ 0.0000 & 0.7420 \end{bmatrix}$	$\begin{bmatrix} 1.8857 & 0.0000 \\ 0.0000 & 0.7420 \end{bmatrix}$	$\begin{bmatrix} 0.8734 \\ -0.7228 \end{bmatrix}$	$\begin{bmatrix} 0.0000 & 0.5514 \\ 0.0000 & 0.2413 \end{bmatrix}$	$\begin{bmatrix} 0.3040 & 0.1330 \\ 0.1330 & 0.0582 \end{bmatrix}$
7	F		$\begin{bmatrix} 0.0258 & -9.9582 \\ 0.0331 & 0.0101 \end{bmatrix}$	$\begin{bmatrix} 0.0060 & 0.0027 \\ -0.0304 & 0.0299 \end{bmatrix}$	$\begin{bmatrix} 4.7903 \\ 3.1074 \end{bmatrix}$	$\begin{bmatrix} 0.0598 & 0.0933 \\ 0.0000 & 0.0433 \end{bmatrix}$	$\begin{bmatrix} 0.0123 & 0.0040 \\ 0.0040 & 0.0019 \end{bmatrix}$
8	E		$\begin{bmatrix} 0.1381 & 2.8358 \\ 0.0000 & 0.0000 \end{bmatrix}$	$\begin{bmatrix} 0.0305 & -0.0573 \\ -0.0573 & 0.1075 \end{bmatrix}$	$\begin{bmatrix} 3.1301 \\ 1.0128 \end{bmatrix}$	$\begin{bmatrix} 0.0266 & 0.0208 \\ 0.0000 & 0.0383 \end{bmatrix}$	$\begin{bmatrix} 0.0011 & 0.0008 \\ 0.0008 & 0.0015 \end{bmatrix}$
9	B					$\begin{bmatrix} 0.0147 & 0.0650 \\ 0.0000 & 0.0395 \end{bmatrix}$	$\begin{bmatrix} 0.0044 & 0.0026 \\ 0.0026 & 0.0016 \end{bmatrix}$
10	E		$\begin{bmatrix} 0.1557 & -0.8257 \\ 0.0000 & 0.0730 \end{bmatrix}$	$\begin{bmatrix} 0.1307 & 0.0380 \\ 0.0380 & 0.0981 \end{bmatrix}$	$\begin{bmatrix} 3.7885 \\ 1.9502 \end{bmatrix}$	$\begin{bmatrix} 0.0619 & 0.0914 \\ 0.0000 & 0.0560 \end{bmatrix}$	$\begin{bmatrix} 0.0122 & 0.0051 \\ 0.0051 & 0.0031 \end{bmatrix}$
11	B					$\begin{bmatrix} 0.0372 & 0.0638 \\ 0.0000 & 0.0556 \end{bmatrix}$	$\begin{bmatrix} 0.0055 & 0.0036 \\ 0.0036 & 0.0031 \end{bmatrix}$
12	F		$\begin{bmatrix} 0.1785 & -0.5817 \\ -1.9349 & 0.1476 \end{bmatrix}$	$\begin{bmatrix} 0.8654 & -1.6329 \\ 0.3020 & -0.5393 \end{bmatrix}$	$\begin{bmatrix} 2.4176 \\ 0.4985 \end{bmatrix}$	$\begin{bmatrix} 0.1382 & 0.1469 \\ 0.0000 & 0.0426 \end{bmatrix}$	$\begin{bmatrix} 0.0407 & 0.0063 \\ 0.0063 & 0.0018 \end{bmatrix}$

M.2. Inferred scores and parameters for the model fits to the phylogenetic principal component (pPC) scores of the mammal data.

Table S11. Competing model fits to the phylogenetic principal component scores (pPC scores) of the mammal data. The notation is the same as in table 1, main text. Note that, after pPCA transformation, the pPC scores are phylogenetically uncorrelated (assuming BM-process). Hence, the models Global BM_A, Global OU_C and SURFACE OU, which assume trait independence, perform comparatively better in terms of AIC-score (ΔAIC values much smaller compared to the model fits on the original data (table 1, main text). Note also that, due to the same reason, the maximum log-likelihood values for the global BM_A and BM_B models are exactly equal and they match the maximum log-likelihood value of the BM_B model fit to the original data (table 1 main text; see also SI Appendix, Section K). The maximum log-likelihood parameter estimates for the Global OU_C and the Global OU_D models converged to the parameters of the Global BM_B model, hence the maximum log-likelihood values are the same. This reveals that global OU models with diagonal selection strength matrix $\mathbf{H}$ do not fit better to the pPC scores relative to the Global BM_B model.

model	q	R	p	$\ell\ell$	AIC	ΔAIC
Global BM_A	n.a.	1	4	30.60	-53.19	183.91
Global BM_B	n.a.	1	5	30.60	-51.19	185.91
Global OU_C	n.a.	1	8	30.60	-45.19	191.91
Global OU_D	n.a.	1	9	30.59	-43.19	193.91
Global OU_E	n.a.	1	10	47.91	-75.82	161.28
Global OU_F	n.a.	1	11	66.30	-110.61	126.49
SURFACE OU	20	2	11	34.78	-47.55	189.55
SCALAR OU	20	10	62	125.01	-126.02	111.08
RATEMATRIX BM	20	9	37	115.92	-157.84	79.26
MGPM* (A-F)	20	13	96	214.55	-237.10	0.00

Table S12. Inferred parameters of model Global BM_A to pPCA-rotated body- and brain-mass data from 629 mammal species. See legend for SI Appendix, table S1.

regime	type	$X0$	Σ_u	Σ
:global:	NA	$\begin{bmatrix} 0.0051 \\ -0.0033 \end{bmatrix}$		
1	A		$\begin{bmatrix} 0.1081 & 0.0000 \\ 0.0000 & 0.0177 \end{bmatrix}$	$\begin{bmatrix} 0.0117 & 0.0000 \\ 0.0000 & 0.0003 \end{bmatrix}$

Table S13. Inferred parameters of model Global BM_B to pPCA-rotated body- and brain-mass data from 629 mammal species. See legend for SI Appendix, table S1.

regime	type	$X0$	Σ_u	Σ
:global:	NA	$\begin{bmatrix} 0.0112 \\ 0.0027 \end{bmatrix}$		
1	B		$\begin{bmatrix} 0.1081 & -0.0001 \\ 0.0000 & 0.0177 \end{bmatrix}$	$\begin{bmatrix} 0.0117 & -0.0000 \\ -0.0000 & 0.0003 \end{bmatrix}$

Table S14. Inferred parameters of model Global OU_C to pPCA-rotated body- and brain-mass data from 629 mammal species. See legend for SI Appendix, table S1.

regime	type	$X0$	H_S	H	Θ	Σ_u	Σ
:global:	NA	$\begin{bmatrix} 0.0132 \\ -0.0017 \end{bmatrix}$					
1	C		$\begin{bmatrix} 0.0000 & 0.0000 \\ 0.0000 & 0.0000 \end{bmatrix}$	$\begin{bmatrix} 0.0000 & 0.0000 \\ 0.0000 & 0.0000 \end{bmatrix}$	$\begin{bmatrix} -0.9312 \\ -0.2963 \end{bmatrix}$	$\begin{bmatrix} 0.1081 & 0.0000 \\ 0.0000 & 0.0177 \end{bmatrix}$	$\begin{bmatrix} 0.0117 & 0.0000 \\ 0.0000 & 0.0003 \end{bmatrix}$

Table S15. Inferred parameters of model Global OU_D to pPCA-rotated body- and brain-mass data from 629 mammal species. See legend for SI Appendix, table S1.

regime	type	$X0$	H_S	H	Θ	Σ_u	Σ
:global:	NA	$\begin{bmatrix} -0.0083 \\ 0.0020 \end{bmatrix}$					
1	D		$\begin{bmatrix} 0.0000 & 0.0000 \\ 0.0000 & 0.0000 \end{bmatrix}$	$\begin{bmatrix} 0.0000 & 0.0000 \\ 0.0000 & 0.0000 \end{bmatrix}$	$\begin{bmatrix} 0.3522 \\ -0.0392 \end{bmatrix}$	$\begin{bmatrix} 0.1081 & -0.0002 \\ 0.0000 & 0.0177 \end{bmatrix}$	$\begin{bmatrix} 0.0117 & -0.0000 \\ -0.0000 & 0.0003 \end{bmatrix}$

Table S16. Inferred parameters of model Global OU_E to pPCA-rotated body- and brain-mass data from 629 mammal species. See legend for SI Appendix, table S1.

regime	type	$X0$	H_S	H	Θ	Σ_u	Σ
:global:	NA	$\begin{bmatrix} -0.0223 \\ 0.2737 \end{bmatrix}$					
1	E		$\begin{bmatrix} 0.0000 & -0.3850 \\ 0.0000 & 0.0080 \end{bmatrix}$	$\begin{bmatrix} 0.0005 & -0.0020 \\ -0.0020 & 0.0074 \end{bmatrix}$	$\begin{bmatrix} 0.2810 \\ -0.0637 \end{bmatrix}$	$\begin{bmatrix} 0.1055 & -0.0317 \\ 0.0000 & 0.0201 \end{bmatrix}$	$\begin{bmatrix} 0.0121 & -0.0006 \\ -0.0006 & 0.0004 \end{bmatrix}$

Table S17. Inferred parameters of model Global OU_F to pPCA-rotated body- and brain-mass data from 629 mammal species. See legend for SI Appendix, table S1.

regime	type	$X0$	H_S	H	Θ	Σ_u	Σ
:global:	NA	$\begin{bmatrix} -2.5749 \\ 0.2290 \end{bmatrix}$					
1	F		$\begin{bmatrix} 0.0000 & 0.0551 \\ -0.1140 & 0.0009 \end{bmatrix}$	$\begin{bmatrix} -0.0042 & -0.1138 \\ 0.0002 & 0.0051 \end{bmatrix}$	$\begin{bmatrix} -0.5620 \\ -0.1117 \end{bmatrix}$	$\begin{bmatrix} 0.0897 & -0.0325 \\ 0.0000 & 0.0209 \end{bmatrix}$	$\begin{bmatrix} 0.0091 & -0.0007 \\ -0.0007 & 0.0004 \end{bmatrix}$

Table S18. Inferred parameters of model SURFACE OU to pPCA-rotated body- and brain-mass data from 629 mammal species. See legend for SI Appendix, table S1.

regime	type	$X0$	H_S	H	Σ_u	Σ	Θ
:global:	NA	$\begin{bmatrix} -0.0181 \\ -0.2177 \end{bmatrix}$	$\begin{bmatrix} 0.0000 & 0.0000 \\ 0.0000 & 0.0021 \end{bmatrix}$	$\begin{bmatrix} 0.0000 & 0.0000 \\ 0.0000 & 0.0021 \end{bmatrix}$	$\begin{bmatrix} 0.1081 & 0.0000 \\ 0.0000 & 0.0182 \end{bmatrix}$	$\begin{bmatrix} 0.0117 & 0.0000 \\ 0.0000 & 0.0003 \end{bmatrix}$	
1	SURFACE OU						$\begin{bmatrix} 0.3880 \\ 0.5590 \end{bmatrix}$
2	SURFACE OU						$\begin{bmatrix} 1.5132 \\ -0.8685 \end{bmatrix}$

Table S19. Inferred parameters of model SCALAR OU to pPCA-rotated body- and brain-mass data from 629 mammal species. See legend for SI Appendix, table S1.

regime	type	$X0$	H_S	H	Θ	Σ_u	Σ
:global:	NA	$\begin{bmatrix} 0.4095 \\ -0.0730 \end{bmatrix}$	$\begin{bmatrix} 0.0014 & 0.0000 \\ 0.0000 & 0.0014 \end{bmatrix}$	$\begin{bmatrix} 0.0014 & 0.0000 \\ 0.0000 & 0.0014 \end{bmatrix}$			
1	SCALAR OU				$\begin{bmatrix} -3.1113 \\ 0.1298 \end{bmatrix}$	$\begin{bmatrix} 0.1113 & 0.0104 \\ 0.0000 & 0.0189 \end{bmatrix}$	$\begin{bmatrix} 0.0125 & 0.0002 \\ 0.0002 & 0.0004 \end{bmatrix}$
2	SCALAR OU				$\begin{bmatrix} -0.1741 \\ 0.1864 \end{bmatrix}$	$\begin{bmatrix} 0.1117 & -0.0345 \\ 0.0000 & 0.0143 \end{bmatrix}$	$\begin{bmatrix} 0.0137 & -0.0005 \\ -0.0005 & 0.0002 \end{bmatrix}$
3	SCALAR OU				$\begin{bmatrix} 1.9239 \\ 0.1277 \end{bmatrix}$	$\begin{bmatrix} 0.0436 & 0.0875 \\ 0.0000 & 0.0143 \end{bmatrix}$	$\begin{bmatrix} 0.0096 & 0.0013 \\ 0.0013 & 0.0002 \end{bmatrix}$
4	SCALAR OU				$\begin{bmatrix} 2.6041 \\ 0.2360 \end{bmatrix}$	$\begin{bmatrix} 0.0737 & 0.0692 \\ 0.0000 & 0.0143 \end{bmatrix}$	$\begin{bmatrix} 0.0102 & 0.0010 \\ 0.0010 & 0.0002 \end{bmatrix}$
5	SCALAR OU				$\begin{bmatrix} 0.3153 \\ -0.0463 \end{bmatrix}$	$\begin{bmatrix} 0.0736 & -0.0156 \\ 0.0000 & 0.0098 \end{bmatrix}$	$\begin{bmatrix} 0.0057 & -0.0002 \\ -0.0002 & 0.0001 \end{bmatrix}$
6	SCALAR OU				$\begin{bmatrix} -0.9812 \\ -0.0969 \end{bmatrix}$	$\begin{bmatrix} 0.0917 & 0.0147 \\ 0.0000 & 0.0264 \end{bmatrix}$	$\begin{bmatrix} 0.0086 & 0.0004 \\ 0.0004 & 0.0007 \end{bmatrix}$
7	SCALAR OU				$\begin{bmatrix} -2.7486 \\ -0.0321 \end{bmatrix}$	$\begin{bmatrix} 0.1679 & -0.0684 \\ 0.0000 & 0.0254 \end{bmatrix}$	$\begin{bmatrix} 0.0329 & -0.0017 \\ -0.0017 & 0.0006 \end{bmatrix}$
8	SCALAR OU				$\begin{bmatrix} 1.5765 \\ -0.0379 \end{bmatrix}$	$\begin{bmatrix} 0.0723 & -0.0321 \\ 0.0000 & 0.0075 \end{bmatrix}$	$\begin{bmatrix} 0.0063 & -0.0002 \\ -0.0002 & 0.0001 \end{bmatrix}$
9	SCALAR OU				$\begin{bmatrix} 2.1031 \\ -0.3080 \end{bmatrix}$	$\begin{bmatrix} 0.0860 & -0.0389 \\ 0.0000 & 0.0076 \end{bmatrix}$	$\begin{bmatrix} 0.0089 & -0.0003 \\ -0.0003 & 0.0001 \end{bmatrix}$
10	SCALAR OU				$\begin{bmatrix} 0.9019 \\ -0.1403 \end{bmatrix}$	$\begin{bmatrix} 0.1337 & -0.0512 \\ 0.0000 & 0.0391 \end{bmatrix}$	$\begin{bmatrix} 0.0205 & -0.0020 \\ -0.0020 & 0.0015 \end{bmatrix}$

Table S20. Inferred parameters of model RATEMATRIX BM (BM_B with shifts) to pPCA-rotated body- and brain-mass data from 629 mammal species. See legend for SI Appendix, table S1.

regime	type	$X0$	Σ_u	Σ
:global:	NA	$\begin{bmatrix} -0.3650 \\ -0.0042 \end{bmatrix}$		
1	B		$\begin{bmatrix} 0.0924 & 0.0119 \\ 0.0000 & 0.0126 \end{bmatrix}$	$\begin{bmatrix} 0.0087 & 0.0002 \\ 0.0002 & 0.0002 \end{bmatrix}$
2	B		$\begin{bmatrix} 0.1085 & -0.0337 \\ 0.0000 & 0.0139 \end{bmatrix}$	$\begin{bmatrix} 0.0129 & -0.0005 \\ -0.0005 & 0.0002 \end{bmatrix}$
3	B		$\begin{bmatrix} 0.1817 & -0.0153 \\ 0.0000 & 0.0229 \end{bmatrix}$	$\begin{bmatrix} 0.0332 & -0.0004 \\ -0.0004 & 0.0005 \end{bmatrix}$
4	B		$\begin{bmatrix} 0.1057 & 0.0368 \\ 0.0000 & 0.0295 \end{bmatrix}$	$\begin{bmatrix} 0.0125 & 0.0011 \\ 0.0011 & 0.0009 \end{bmatrix}$
5	B		$\begin{bmatrix} 0.1042 & -0.0206 \\ 0.0000 & 0.0262 \end{bmatrix}$	$\begin{bmatrix} 0.0113 & -0.0005 \\ -0.0005 & 0.0007 \end{bmatrix}$
6	B		$\begin{bmatrix} 0.1250 & 0.0210 \\ 0.0000 & 0.0199 \end{bmatrix}$	$\begin{bmatrix} 0.0161 & 0.0004 \\ 0.0004 & 0.0004 \end{bmatrix}$
7	B		$\begin{bmatrix} 0.0880 & 0.0192 \\ 0.0000 & 0.0232 \end{bmatrix}$	$\begin{bmatrix} 0.0081 & 0.0004 \\ 0.0004 & 0.0005 \end{bmatrix}$
8	B		$\begin{bmatrix} 0.0528 & -0.0283 \\ 0.0000 & 0.0195 \end{bmatrix}$	$\begin{bmatrix} 0.0036 & -0.0006 \\ -0.0006 & 0.0004 \end{bmatrix}$
9	B		$\begin{bmatrix} 0.1322 & -0.0475 \\ 0.0000 & 0.0384 \end{bmatrix}$	$\begin{bmatrix} 0.0197 & -0.0018 \\ -0.0018 & 0.0015 \end{bmatrix}$

Table S21. Inferred parameters of model MGPM (A-F) to pPCA-rotated body- and brain-mass data from 629 mammal species. See legend for SI Appendix, table S1.

regime	type	$X0$	H_S	H	Θ	Σ_u	Σ
:global:	NA	$\begin{bmatrix} -0.1208 \\ -0.0625 \end{bmatrix}$					
1	B					$\begin{bmatrix} 0.1187 & 0.0394 \\ 0.0000 & 0.0156 \end{bmatrix}$	$\begin{bmatrix} 0.0156 & 0.0006 \\ 0.0006 & 0.0002 \end{bmatrix}$
2	F		$\begin{bmatrix} 0.0407 & -0.1318 \\ 0.5611 & 0.0469 \end{bmatrix}$	$\begin{bmatrix} -0.0084 & 0.5563 \\ -0.0049 & 0.0960 \end{bmatrix}$	$\begin{bmatrix} 0.5442 \\ 0.0786 \end{bmatrix}$	$\begin{bmatrix} 0.0118 & -0.0076 \\ 0.0000 & 0.0363 \end{bmatrix}$	$\begin{bmatrix} 0.0002 & -0.0003 \\ -0.0003 & 0.0013 \end{bmatrix}$
3	A					$\begin{bmatrix} 0.0936 & 0.0000 \\ 0.0000 & 0.0153 \end{bmatrix}$	$\begin{bmatrix} 0.0088 & 0.0000 \\ 0.0000 & 0.0002 \end{bmatrix}$
4	E		$\begin{bmatrix} 0.0246 & -0.5016 \\ 0.0000 & 0.0496 \end{bmatrix}$	$\begin{bmatrix} 0.0274 & -0.0079 \\ -0.0079 & 0.0468 \end{bmatrix}$	$\begin{bmatrix} 2.9717 \\ 0.3522 \end{bmatrix}$	$\begin{bmatrix} 0.0725 & 0.0453 \\ 0.0000 & 0.0110 \end{bmatrix}$	$\begin{bmatrix} 0.0073 & 0.0005 \\ 0.0005 & 0.0001 \end{bmatrix}$
5	F		$\begin{bmatrix} 0.0947 & 0.0612 \\ -0.7737 & 0.0180 \end{bmatrix}$	$\begin{bmatrix} 0.0630 & -0.7756 \\ -0.0018 & 0.0496 \end{bmatrix}$	$\begin{bmatrix} -2.2553 \\ -0.5231 \end{bmatrix}$	$\begin{bmatrix} 0.0437 & -0.1372 \\ 0.0000 & 0.0348 \end{bmatrix}$	$\begin{bmatrix} 0.0207 & -0.0048 \\ -0.0048 & 0.0012 \end{bmatrix}$
6	F		$\begin{bmatrix} 0.0000 & -0.0549 \\ 0.2484 & 0.1164 \end{bmatrix}$	$\begin{bmatrix} -0.0089 & 0.2438 \\ -0.0046 & 0.1254 \end{bmatrix}$	$\begin{bmatrix} 0.0249 \\ -0.1398 \end{bmatrix}$	$\begin{bmatrix} 0.0769 & -0.0227 \\ 0.0000 & 0.0489 \end{bmatrix}$	$\begin{bmatrix} 0.0064 & -0.0011 \\ -0.0011 & 0.0024 \end{bmatrix}$
7	C		$\begin{bmatrix} 0.4736 & 0.0000 \\ 0.0000 & 0.0528 \end{bmatrix}$	$\begin{bmatrix} 0.4736 & 0.0000 \\ 0.0000 & 0.0528 \end{bmatrix}$	$\begin{bmatrix} 2.6038 \\ 0.2529 \end{bmatrix}$	$\begin{bmatrix} 0.3320 & 0.0000 \\ 0.0000 & 0.0199 \end{bmatrix}$	$\begin{bmatrix} 0.1103 & 0.0000 \\ 0.0000 & 0.0004 \end{bmatrix}$
8	A					$\begin{bmatrix} 0.1036 & 0.0000 \\ 0.0000 & 0.0112 \end{bmatrix}$	$\begin{bmatrix} 0.0107 & 0.0000 \\ 0.0000 & 0.0001 \end{bmatrix}$
9	B					$\begin{bmatrix} 0.1393 & -0.0715 \\ 0.0000 & 0.0243 \end{bmatrix}$	$\begin{bmatrix} 0.0245 & -0.0017 \\ -0.0017 & 0.0006 \end{bmatrix}$
10	A					$\begin{bmatrix} 0.0905 & 0.0000 \\ 0.0000 & 0.0232 \end{bmatrix}$	$\begin{bmatrix} 0.0082 & 0.0000 \\ 0.0000 & 0.0005 \end{bmatrix}$
11	D		$\begin{bmatrix} 0.1184 & 0.0000 \\ 0.0000 & 0.0247 \end{bmatrix}$	$\begin{bmatrix} 0.1184 & 0.0000 \\ 0.0000 & 0.0247 \end{bmatrix}$	$\begin{bmatrix} -1.3026 \\ -0.3427 \end{bmatrix}$	$\begin{bmatrix} 0.0890 & -0.0505 \\ 0.0000 & 0.0228 \end{bmatrix}$	$\begin{bmatrix} 0.0105 & -0.0011 \\ -0.0011 & 0.0005 \end{bmatrix}$
12	C		$\begin{bmatrix} 2.8411 & 0.0000 \\ 0.0000 & 7.1116 \end{bmatrix}$	$\begin{bmatrix} 2.8411 & 0.0000 \\ 0.0000 & 7.1116 \end{bmatrix}$	$\begin{bmatrix} 1.7223 \\ 0.0599 \end{bmatrix}$	$\begin{bmatrix} 0.7108 & 0.0000 \\ 0.0000 & 0.0421 \end{bmatrix}$	$\begin{bmatrix} 0.5053 & 0.0000 \\ 0.0000 & 0.0018 \end{bmatrix}$
13	B					$\begin{bmatrix} 0.1314 & -0.0460 \\ 0.0000 & 0.0373 \end{bmatrix}$	$\begin{bmatrix} 0.0194 & -0.0017 \\ -0.0017 & 0.0014 \end{bmatrix}$

References

1. Khabbazian M, Kriebel R, Rohe K, Ané C (2016) Fast and accurate detection of evolutionary shifts in Ornstein-Uhlenbeck models. *Methods in Ecology and Evolution* 7(7):811–824.
2. Bastide P, Ané C, Robin S, Mariadassou M (2018) Inference of Adaptive Shifts for Multivariate Correlated Traits. *Systematic Biology* 113(4):2158–680.
3. Alfaro ME, et al. (2009) Nine exceptional radiations plus high turnover explain species diversity in jawed vertebrates. *PNAS* 106(32):13410–13414.
4. Ingram T, Mahler DL (2013) SURFACE: detecting convergent evolution from comparative data by fitting Ornstein-Uhlenbeck models with stepwise Akaike Information Criterion. *Methods in Ecology and Evolution* 4(5):416–425.
5. Microsoft, Weston S (2017) foreach: Foreach Looping Construct for R.
6. Revolution Analytics, Weston, Steve (2017) iterators: Iterator Construct for R.
7. Weston S (2017) doMPI: Foreach Parallel Adaptor for the Rmpi Package.
8. Bartoszek K, Pienaar J, Mostad P, Andersson S, Hansen TF (2012) A phylogenetic comparative method for studying multivariate adaptation. *Journal of theoretical biology* 314:204–215.
9. Clavel J, Escarguel G, Merceron G (2015) mvmorph: an r package for fitting multivariate evolutionary models to morphometric data. *Methods in Ecology and Evolution* 6(11):1311–1319.
10. Golub GH, Van Loan CF (2012) *Matrix Computations*. (JHU Press).
11. Byrd RH, Lu P, Nocedal J, Zhu CY (1995) A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing* 16(5):1190–1208.
12. Mitov V, Bartoszek K, Asimomitis G, Stadler T (2018) Fast likelihood calculation for multivariate phylogenetic comparative methods: the PCMBase R package. *arXiv.org* p. arXiv:1809.09014.
13. Felsenstein J (1985) Phylogenies and the Comparative Method. *The American Naturalist* 125(1):1–15.
14. Boddy AM, et al. (2012) Comparative analysis of encephalization in mammals reveals relaxed constraints on anthropoid primate and cetacean brain scaling. *Journal of Evolutionary Biology* 25(5):981–994.
15. Paradis E, Claude J, Strimmer K (2004) APE: Analyses of Phylogenetics and Evolution in R language. *Bioinformatics* 20(2):289–290.
16. Sanderson C, Curtin R (2016) Armadillo: a template-based C++ library for linear algebra. *Journal of Open Source Software* 1(2).
17. Goulet V, et al. (2018) expm: Matrix Exponential, Log.
18. Genz A, Bretz F (2009) *Computation of Multivariate Normal and t Probabilities*. (Springer Science & Business Media).
19. Dowle M, Short T, Liangolou S, Srinivasan A (2014) data.table: Extension of data.frame. *R package* p. 9.
20. Wickham H (2009) ggplot2 - Elegant Graphics for Data Analysis. *Use R*.
21. Yu G, Smith DK, Zhu H, Guan Y, Lam TTY (2017) GGTREE: an R package for visualization and annotation of phylogenetic trees with their covariates and other associated data. *Methods in Ecology and Evolution* 8(1):28–36.
22. Yu G (2018) ggimage: Use Image in 'ggplot2'.
23. Eddelbuettel D (2018) digest: Create Compact Hash Digests of R Objects.
24. Eddelbuettel D (2013) *Seamless R and C++ Integration with Rcpp*. (Springer Science & Business Media, New York, NY).
25. Revell LJ (2011) phytools: an R package for phylogenetic comparative biology (and other things). *Methods in Ecology and Evolution* 3(2):217–223.
26. Wilke CO (2019) cowplot: Streamlined Plot Theme and Plot Annotations for 'ggplot2' [R package cowplot version 0.9.3].
27. Xie Y (2017) *Dynamic Documents with R and knitr, Second Edition*. (CRC Press).
28. Allaire JJ, et al. (2014) rmarkdown: Dynamic Documents for R.
29. Dahl DB (2018) xtable: Export Tables to LaTeX or HTML.
30. Bininda-Emonds ORP, et al. (2007) The delayed rise of present-day mammals. *Nature* 446(7135):507–512.
31. Boddy A, et al. (2012) Data from: Comparative analysis of encephalization in mammals reveals relaxed constraints on anthropoid primate and cetacean brain scaling.
32. Baker RWR (1963) Expressions for Combining Standard Errors of Two Groups and for Sequential Standard Error. *Nature* 198(4):1020–.
33. Quan H, Zhang J (2003) Estimate of standard deviation for a log-transformed variable using arithmetic means and standard deviations. *Statistics in medicine* 22(17):2723–2736.
34. FitzJohn RG (2012) Diversitree: comparative phylogenetic analyses of diversification in R. *Methods in Ecology and Evolution* 3(6):1084–1092.
35. Hennig C (2018) *fpc: Flexible Procedures for Clustering*. R package version 2.1-11.1.
36. Uyeda JC, Pennell MW, Miller ET, Maia R, McClain CR (2017) The Evolution of Energetic Scaling across the Vertebrate Tree of Life. *American Naturalist* 190(2):185–199.
37. Goolsby EW, Bruggeman J, Ané C (2016) Rphylopars: fast multivariate phylogenetic comparative methods for missing data and within-species variation. *Methods in Ecology and Evolution* 8(1):22–27.
38. Ho LsT, Ané C (2014) A linear-time algorithm for Gaussian and non-Gaussian trait evolution models. *Systematic Biology* 63(3):397–408.
39. Caetano DS, Harmon LJ (2017) ratematrix: An Rpackage for studying evolutionary integration among several traits on phylogenetic trees. *Methods in Ecology and Evolution* 8(12):1920–1927.

40. Adams DC, Collyer ML (2018) Multivariate Phylogenetic Comparative Methods: Evaluations, Comparisons, and Recommendations. *Systematic Biology* 67(1):14–31.
41. Revell LJ (2009) Size-correction and principal components for interspecific comparative studies. *Evolution* 63(12):3258–3268.
42. Uyeda JC, Caetano DS, Pennell MW (2015) Comparative Analysis of Principal Components Can be Misleading. *Systematic Biology* 64(4):677–689.
43. Revell LJ, Harmon LJ (2008) Testing quantitative genetic hypotheses about the evolutionary rate matrix for continuous characters. *Evolutionary Ecology Research* 10(3):311–331.