Cophylogeny reconstruction allowing for multiple associations through approximate Bayesian computation – Supplementary Material

Blerina Sinaimeri[1,2], Laura Urbini[2*], Marie-France Sagot[2] and Catherine Matias[3]

[1] *LUISS University, Rome, Italy*

[2] *Inria Lyon, 56 Bd Niels Bohr, 69100 Villeurbanne, France, and Université de Lyon, F-69000, Lyon; Université Lyon 1; CNRS, UMR5558; 43 Boulevard du 11 Novembre 1918, 69622 Villeurbanne cedex, France*

[3] *Sorbonne Université, Université de Paris Cité, Centre National de la Recherche Scientifique, Laboratoire de Probabilités, Statistique et Modélisation, Paris, France*

Corresponding author: Blerina Sinaimeri, LUISS University, Rome, Italy; E-mail: bsinaimeri@luiss.it.

Contents

*First co-authors.

A The event-based model

AmoCoala relies on the event-based model presented in Charleston (2002); Tofigh *et al.* (2011). For the sake of completeness, we detail the model here. We first start with some basic definitions related to phylogenetic trees.

A.1 Tree-related basic definitions

A rooted phylogenetic tree is a leaf-labelled tree that models the evolution of a set of taxa from their most recent common ancestor (placed at the root). The internal vertices of the tree correspond to the speciation events. In a rooted phylogenetic tree, a direction is assumed from the root to the leaves that corresponds to the direction of evolutionary time. Specifically, a phylogenetic tree is a rooted tree with labelled leaves where the root has in-degree 0 and out-degree 2, the leaves have in-degree 1 and out-degree 0 and every internal vertex has in-degree 1 and out-degree 2. For such a tree T, the set of vertices is denoted by $V(T)$, the set of arcs by $A(T)$, and the set of leaves by $L(T)$. The cardinality of set A is denoted by $|A|$. The root of T is denoted by $r(T)$. For a vertex v in a tree T, we denote by T_v the subtree of T rooted in v (often referred to as a *clade*), and we write $L(v)$ for the set $L(T_v)$. For a vertex $v \in V(T)$, we denote by $Des(v)$ the set of *descendants* of v, *i.e.* the set of vertices in the subtree of T_v. Similarly, we denote by $Anc(v)$ the set of *ancestors* of v, that is the set of vertices in the unique path from $r(T)$ to v (including the end points). For a vertex $v \in V(T)$ different from the root, we call its *parent*, denoted by $par(v)$, the vertex x for which there is the arc $(x, v) \in A(T)$. We denote by $\mathsf{mrca}(v, w)$ the most recent common ancestor of v and w in T. Finally, we denote by $\leq$ the partial order induced by the ancestry relation in the tree. Formally, for $x, y \in V(T)$, we say that $x \leq y$ if $x \in Anc(y)$. If neither $x \in Anc(y)$ nor $y \in Anc(x)$, the vertices x and y are said to be *incomparable*.

For any tree T and any set of leaves $t_1, \ldots, t_n$, we denote by $T_{|\{t_1,\ldots,t_n\}}$ the phylogenetic subtree of T induced by the leaves $t_1, \ldots, t_n$ and eventually

suppressing the vertices of out-degree 1. When a vertex u with parent vertex v and child vertex w is suppressed, both vertex u and arcs $(v,u),(u,w)$ are removed and the arc (v,w) is added to the tree.

A.2 Reconciliation model from Tofigh et al.

In this section, we describe the classical reconciliation model, where 4 co-evolutionary events are allowed, producing no multiple associations. Let H and S be respectively the rooted phylogenetic trees of the host and symbiont species, both binary and full (*i.e.* each internal vertex has exactly two children). Let ϕ be a function from $L(S)$ to $L(H)$, representing the symbiont/host associations between extant species. A reconciliation is a function λ that assigns, for each symbiont vertex $s \in V(S)$, a host vertex $\lambda(h) \in V(H)$, and satisfies the conditions stated in Definition A.1.

In its classical form, a reconciliation associates to each vertex s in $V(S)$ an event $E(\lambda(s))$ among cospeciation ($\mathbb{C}$), duplication ($\mathbb{D}$) and host switch ($\mathbb{S}$).

Definition A.1. *Given two phylogenetic trees S and H, and a function $\phi : L(S) \to L(H)$, a reconciliation of (S,H,ϕ) is a function $\lambda : V(S) \to V(H)$ satisfying the following:*

1. *For every leaf vertex $s \in L(S)$, we have $\lambda(s) = \phi(s)$.*
2. *For every internal vertex $s \in V(S) \setminus L(S)$ with children s_1, s_2, exactly one of the following applies:*
   (a) *$E(\lambda(s)) = \mathbb{S}$, that is, either $\lambda(s_1)$ and $\lambda(s)$ are incomparable and $\lambda(s_2)$ is a descendant of $\lambda(s)$, or $\lambda(s_2)$ and $\lambda(s)$ are incomparable and $\lambda(s_1)$ is a descendant of $\lambda(s)$,*
   (b) *$E(\lambda(s)) = \mathbb{C}$, that is, $\mathsf{mrca}(\lambda(s_1),\lambda(s_2)) = \lambda(s)$, and $\lambda(s_1)$ and $\lambda(s_2)$ are incomparable,*
   (c) *$E(\lambda(s)) = \mathbb{D}$, that is, $\lambda(s_1)$ and $\lambda(s_2)$ are both descendants of $\lambda(s)$, and the previous two cases do not apply.*

The loss event is denoted by $\mathbb{L}$ and is identified by a multiset (generalisation of a set where the elements are allowed to appear more than once) whose elements are in $V(H)$ containing all the vertices $h \in V(H)$ that are in the path between the image of a vertex $s \in V(S)$ and the image of one of

its children. The images themselves are not included in the count, except for the duplication event, where one of the images is included.

The function λ partitions the set of internal symbiont tree vertices into three disjoint subsets according to the coevolutionary event occurring at that vertex. The number of occurrences of each of the three events and the number of losses make up the *event vector* of the reconciliation. The *event vector* of a reconciliation is a vector of integers consisting of the total number of each type of events $\mathbb{C}$, $\mathbb{D}$, $\mathbb{S}$, $\mathbb{L}$.

We say that a reconciliation is *time-feasible* if it does not violate the time-feasibility constraints. The exact criterion we use to assess time-feasibility is the one defined in Stolzer *et al.* (2012) and that was already in force in Coala.

A.3 Reconciliation model allowing for spreads

The introduction of spread events modifies the previous setting in the following way. Let again H and S be respectively the rooted phylogenetic trees of the host and symbiont species, both binary and full (*i.e.* every internal vertex has exactly two children). Now, let ϕ be a relation between $L(S)$ and $L(H)$, representing the symbiont/host associations between extant species. More precisely, let us denote $\mathcal{P}(L(H))$ the set of all subsets of $L(H)$. Then ϕ is now a function from $L(S)$ to $\mathcal{P}(L(H))$. For any extant symbiont species $s \in L(S)$, whenever the cardinality $|\phi(s)| \geq 2$ (*i.e.* whenever the symbiont is associated to more than one host), we say that this symbiont has multiple associations and we count the total number of multiple associations in the dataset as:

$$\text{Nb of multiple associations } = \sum_{s \in L(S)} (|\phi(s)| - 1).$$

A reconciliation is now a function λ from $V(S)$ to $\mathcal{P}(V(H))$ that assigns, for each symbiont vertex $s \in V(S)$, a set of host vertices $\lambda(s) \subset V(H)$, and satisfies the conditions stated in Definition A.2. A reconciliation now associates to each vertex s in $V(S)$ an event $E(\lambda(s))$ among cospeciation ($\mathbb{C}$), duplication ($\mathbb{D}$), host switch ($\mathbb{S}$), vertical spread ($\mathbb{VS}$) and horizontal spread ($\mathbb{HS}$).

Definition A.2. *Given two phylogenetic trees S and H, and a function $\phi : L(S) \to \mathcal{P}(L(H))$, a reconciliation of (S, H, ϕ) is a function $\lambda : V(S) \to \mathcal{P}(V(H))$ satisfying the following:*

1. *For every leaf vertex $s \in L(S)$, we have $\lambda(s) = \phi(s)$.*

2. *For every internal vertex $s \in V(S) \setminus L(S)$ with children s_1, s_2, such that $\lambda(s)$ is a singleton, exactly one of the following applies:*

   (a) *$E(\lambda(s)) = \mathbb{S}$, that is, either $\lambda(s)$ and one element of $\lambda(s_1)$ are incomparable and $\lambda(s_2)$ contains a descendant of $\lambda(s)$, or $\lambda(s)$ and one element of $\lambda(s_2)$ are incomparable and $\lambda(s_2)$ contains a descendant of $\lambda(s)$,*

   (b) *$E(\lambda(s)) = \mathbb{C}$, that is, there is some $h_1 \in \lambda(s_1)$ (resp. $h_2 \in \lambda(s_2)$) such that $\mathsf{mrca}(h_1, h_2) = \lambda(s)$, and h_1 and h_2 are incomparable,*

   (c) *$E(\lambda(s)) = \mathbb{D}$, that is, there is some $h_1 \in \lambda(s_1)$ (resp. $h_2 \in \lambda(s_2)$) such that both h_1, h_2 are descendants of $\lambda(s)$, and the previous two cases do not apply.*

3. *For every internal vertex $s \in V(S) \setminus L(S)$ such that $\lambda(s)$ is not a singleton, exactly one of the following applies:*

   (a) *$E(\lambda(s)) = \mathbb{VS}$, that is $\lambda(s)$ is a clade in H, and all the descendants s' of s are also associated to the same clade,* i.e. *$\lambda(s') = \lambda(s)$.*

   (b) *$E(\lambda(s)) = \mathbb{HS}$, that is $\lambda(s)$ is the union of two clades in H whose respective roots are incomparable. Moreover, all the descendants s' of s are also associated to the same clades,* i.e. *$\lambda(s') = \lambda(s)$.*

   (c) *s is the descendant of a node s' where a spread (either vertical or horizontal) occurred (cases (3a) and (3b)). Then $\lambda(s) = \lambda(s')$. In that case, no additional coevolutionary event is recorded at that vertex.*

The loss event denoted by $\mathbb{L}$ is identified by a multiset (generalisation of a set where the elements are allowed to appear more than once) whose elements are in $V(H)$ containing all the vertices $h \in V(H)$ that are in the path between the image of a vertex $s \in V(S)$ which is a singleton and the image of one of its children. Note that no other event and thus no losses can happen below spread events.

Now, the function λ partitions the set of internal symbiont tree vertices into five disjoint subsets according to the coevolutionary event occurring at that vertex, plus an additional subset of all internal symbiont vertices that descend from a vertex where a spread occurred. The number of occurrences

of each of the five events and the number of losses make up the *event vector* of the reconciliation. The *event vector* of a reconciliation is a vector of integers consisting of the total number of each type of events $\mathbb{C}$, $\mathbb{D}$, $\mathbb{S}$, $\mathbb{L}$, $\mathbb{VS}$, $\mathbb{HS}$. Note that in the case of spread events (either vertical or horizontal) occurring at internal vertex $s \in V(S) \setminus L(S)$, the event is counted only once and the internal vertices s' descendants of s have no coevolutionary event associated to them.

The time feasibility condition is unchanged when adding spreads in the list of coevolutionary events.

A.4 Pre-estimating probabilities for the spread events

Given an input dataset (H, S, ϕ), we rely on frequency estimators for the spread probabilities that will be used in our algorithm. Note that the "classical events" (cospeciation, duplication, host switch and loss) have the same probability to occur everywhere in the tree, while the probability of a vertical or horizontal spread is specific to each vertex of the host tree. These probabilities are pre-estimated based on the input (H, S, ϕ) as described below rather than in the full ABC procedure. They are estimated through heuristic frequencies observed in the associations of the two trees. In Section D.3, we explore the robustness of our results with respect to these pre-computed estimators.

Probability that a vertical spread occurs at host h. A probability $p_{\text{vs}}(h)$ is associated to a vertical spread event at host h as follows. If $h \in L(H)$, then $p_{\text{vs}}(h)$ is estimated to 1. Otherwise, for any internal vertex h of the host tree H, the probability $p_{\text{vs}}(h)$ is estimated to

$$p_{\text{vs}}(h) = \left(\frac{1}{|S^{L(h)}|}\right) \frac{\sum_{s \in S^{L(h)}} |\phi(s) \cap L(h)| - 1}{|L(h)| - 1} \tag{A.1}$$

where $L(h)$ is the set of leaves in H_h (the subtree of H rooted in h), $S^{L(h)}$ is the set of leaves in the symbiont tree S that are associated with at least one leaf of H_h (formally $S^{L(h)} = \{s \in L(S) : \phi(s) \cap L(h) \neq \emptyset\}$), and $|\phi(s) \cap L(h)|$ is the number of host leaves in H_h associated with a symbiont s.

Intuitively, the probability $p_{\text{vs}}(h)$ is large whenever a large proportion of the symbionts in $S^{L(h)}$ are associated to a large proportion of the hosts $L(h)$ (*i.e.* most of the symbionts are generalists) and is low when most of

those symbionts are associated only with a few hosts of $L(h)$ (*i.e.* most of the symbionts are specialists). Notice that for a host h that is high in the tree, *i.e.* that is near to the root of H, the set $L(h)$ is large. Thus, a vertical spread to occur at h with high probability requires that some symbiont leaves are associated to an unrealistically large set of hosts $L(h)$. Hence usually the probability of a vertical spread is lower in hosts that are high in the tree. As explained in the next paragraph, the same holds for the horizontal spread event.

Probability that a symbiont present in h invades an incomparable host h'. For two incomparable vertices h and h', a probability $p_{\text{jump}}(h \to h')$ is estimated as follows

$$p_{\text{jump}}(h \to h') = \frac{|S^{L(h)} \cap S^{L(h')}|}{|S^{L(h)} \cup S^{L(h')}|}. \tag{A.2}$$

The notion of "jump" does not refer to a coevolutionary event and should not be confused with a host switch. The jump probability is specific to each pair of vertices of the host tree. It is a symmetric quantity, *i.e.* $p_{\text{jump}}(h \to h') = p_{\text{jump}}(h' \to h)$. It is high whenever the leaves of the subtrees H_h and $H_{h'}$ share a large proportion of associated symbionts. In particular, it is zero when they do not share any associated symbiont, and 1 when they have exactly the same set of associated symbionts.

Probability that a horizontal spread occurs at host h. From the probabilities $p_{\text{jump}}(h \to h')$, we estimate a probability of horizontal spread at each vertex h. The associated probability depends on all the vertices h' that are incomparable with h. Indeed, such vertices are all those that may be reached from h through a horizontal spread event. In fact, a horizontal spread corresponds to a jump combined with two vertical spreads. We thus associate a probability of horizontal spread $p_{\text{hs}}(h)$ to each vertex h of the host tree that takes into account both a jump and two vertical spreads and is set as

$$p_{\text{hs}}(h) = \min\{1, p^*(h)\}, \tag{A.3}$$

where

$$p^*(h) = p_{\text{vs}}(h) \sum_{\substack{h' \in V(H) \\ h,h' \text{ incomparable}}} p_{\text{vs}}(h') p_{\text{jump}}(h \to h').$$

The probability of a horizontal spread $p_{\text{hs}}(h)$ is high whenever $p_{\text{vs}}(h)$ is high and there exist vertices h' incomparable to h with large $p_{\text{vs}}(h)$ and large value $p_{\text{jump}}(h \to h')$ (so that the leaves below h and h' share many symbionts). Observe that $p^*(h)$ is not a probability but a positive value, that in particular may be larger than 1.

Probability for sampling a horizontal spread to some specific host h'. In the simulation process, once a horizontal spread is sampled for symbiont s at vertex h, we need to choose an incomparable vertex h' where the symbiont s has to jump to. In this case, we need to guarantee that the jump satisfies the time-feasibility constraints as given in Stolzer *et al.* (2012) and Baudet *et al.* (2015). This constraint depends on the symbionts mapped so far (see Section *Simulation algorithm in* AmoCoala below). For a current partial mapping λ from the vertices of S to the subsets of vertices of H, the probability $p_{\text{invasion}}(h \to h', \lambda)$ of a vertex h' to be invaded by a symbiont s mapped in h is estimated as

$$\begin{aligned} p_{\text{invasion}}(h \to h', \lambda) &= \frac{p_{\text{jump}}(h \to h') 1\{E_{h,h',\lambda}\} p_{\text{vs}}(h) p_{\text{vs}}(h')}{p_{\text{vs}}(h) \sum_{h''} p_{\text{vs}}(h'') p_{\text{jump}}(h \to h'') 1\{E_{h,h'',\lambda}\}}, \\ &= \frac{p_{\text{jump}}(h \to h') 1\{E_{h,h',\lambda}\} p_{\text{vs}}(h')}{\sum_{h''} p_{\text{vs}}(h'') p_{\text{jump}}(h \to h'') 1\{E_{h,h'',\lambda}\}}, \end{aligned} \tag{A.4}$$

where $1\{E_{h,h',\lambda}\} = 1$ whenever the horizontal spread of the symbiont mapped in h to the new host h' induces a time feasible reconciliation, and the sum in the denominator is restricted to the vertices h'' that are incomparable to h. If no vertex induces a time feasible reconciliation (namely $p_{\text{invasion}}(h \to h', \lambda) = 0$ for any h' incomparable to h), the horizontal spread is not applied and another event is sampled. Otherwise, as the probabilities $p_{\text{invasion}}(h \to h', \lambda)$ sum up to one, a vertex h' is necessarily chosen.

Computing the pre-estimated spread probabilities. The estimated spread probabilities are calculated at the beginning of the algorithm. These values depend only on the host tree H, the symbiont tree S and the associations between the leaves ϕ. In a first step, we start by setting to 1 the probabilities p_{vs} for the leaves. Then, for the internal vertices h, these probabilities are computed as in Equation (A.1). In a second step, the probabilities of a jump are calculated for each pair of incomparable vertices h and h' as in Equation (A.2). In the last step, the probabilities of a horizontal

spread for vertex h are computed as in Equation (A.3). Observe that the probabilities of invasion (Equation (A.4)) depend on the current simulation. Indeed, one has to take into account the time-feasibility in order to choose the target h' of a horizontal spread. Therefore, it may happen that the invasion $p_{\text{invasion}}(h \to h', \lambda) > 0$ for the current partial mapping λ but after some steps $p_{\text{invasion}}(h \to h', \lambda') = 0$ for the new mapping λ'. These probabilities are then updated, during the simulation algorithm, each time a horizontal spread is selected.

B AmoCoala algorithm

B.1 Simulation algorithm in AmoCoala

The simulation of a symbiont tree $\tilde{S}$ together with its reconciliation $\tilde{\lambda}$ starts with the creation of its root vertex $\tilde{s}_{root}$. This vertex is positioned before the root of H on the arc $a = (\rho, H_{root})$. We add the arc (ρ, H_{root}) to allow the simulation of events that happened in the symbiont tree before the most recent common ancestor of all host species in H. Figure **??** in main text depicts this starting configuration.

For any vertex $\tilde{s}$ of $\tilde{S}$ that is not yet mapped and whose position is $\langle \tilde{s} : a \rangle$ (see Figure **??** in main text), AmoCoala successively considers the six allowed operations, and chooses one depending on the probability of each event (once an event is picked, the others are not considered). In what follows, we denote by a_1, a_2 the arcs outgoing from the head $h(a)$ of the arc a.

I. If $h(a)$ is a leaf, we *STOP* the evolution of $\tilde{s}$.

II. We first sample a horizontal spread according to the probability $p_{\text{hs}}\big(h(a)\big)$. When a horizontal spread occurs (Figure **??** in main text), we apply the mapping $\tilde{\lambda}(\tilde{s}) = H_{h(a)} \cup H_{h(a')}$. The choice of the incomparable vertex $h(a')$ varies in order to preserve time feasibility (Stolzer *et al.*, 2012; Baudet *et al.*, 2015), thus the probabilities described in Equation (A.4) are updated according to the new set of incomparable vertices. If there is no incomparable vertex, it is not possible for a horizontal spread to occur and we go to Step III. To select the ghost subtree rooted in $\tilde{s}$, we mimic the real symbiont tree as shown in Figure **??** in main text.

III. If a horizontal spread did not occur, we sample a vertical spread according to the probability $p_{\text{vs}}\big(h(a)\big)$. When a vertical spread occurs (Figure **??** in main text), we apply the mapping $\tilde{\lambda}(\tilde{s}) = H_{h(a)}$. To select the ghost subtree rooted in $\tilde{s}$, we mimic the real symbiont tree as shown in Figure **??** from main text.

In both cases of vertical and horizontal spreads, the evolution of $\tilde{s}$ stops after the creation of the ghost subtree and its descendants are not processed anymore.

IV. If a spread was not sampled, then we sample with a multinomial distribution a classical event according to the probabilities $\theta = \langle p_c, p_d, p_s, p_l \rangle$. Notice that $p_c + p_d + p_s + p_l = 1$ so that one of the four events is selected. This case is handled identically as in Coala and the symbiont is associated to a single host. We briefly recall the procedure below.

- Cospeciation (Figure **??**(b) in main text): We apply the mapping $\tilde{\lambda}(\tilde{s}) = \{h(a)\}$ and we create the vertices $\tilde{s}_1$ and $\tilde{s}_2$ as children of $\tilde{s}$. We position them as follows: $\langle \tilde{s}_1 : a_1 \rangle$ and $\langle \tilde{s}_2 : a_2 \rangle$. This operation is executed with probability p_c.
- Duplication (Figure **??**(c) in main text): We apply the mapping $\tilde{\lambda}(\tilde{s}) = \{h(a)\}$ and we create the vertices $\tilde{s}_1$ and $\tilde{s}_2$ as children of $\tilde{s}$. Both $\tilde{s}_1$ and $\tilde{s}_2$ are positioned on a. This operation is executed with probability p_d.
- Host switch (Figure **??**(e) in main text): We apply the mapping $\tilde{\lambda}(\tilde{s}) = \{h(a)\}$ and we create the vertices $\tilde{s}_1$ and $\tilde{s}_2$ as children of $\tilde{s}$. We then randomly choose one of the two children and position it on a. Finally, we randomly choose an arc a' that does not violate the time feasibility of the reconstruction so far (Stolzer *et al.*, 2012; Baudet *et al.*, 2015). If such an arc does not exist, it is not possible for a host switch to take place. In this case, we choose between the three remaining events with probability $p_i/(p_c + p_d + p_l)$ with $i \in \{c, d, l\}$. Otherwise, we position $\tilde{s}_2$ on a'. This operation is executed with probability p_s.
- Loss (Figure **??**(e) in main text): This operation consists of randomly choosing an arc outgoing from the head $h(a)$ of a and positioning $\tilde{s}$ on it. This operation is executed with probability p_l.

In any of these four cases, the simulation process recursively continues with the new vertices created (back to Step I).

Note that in our modelling, losses never occur after a spread event. Indeed, in the case of a vertical spread, a symbiont and its entire clade are associated to one host clade, while in the case of a horizontal spread, they are then associated to two host clades. This might appear unrealistic. However, this choice is made for computational reasons. Indeed, as mentioned in the Main Manuscript, there is no simple way of simulating the symbiont tree below a symbiont where a spread occurs.

B.2 ABC-SMC inference method in AmoCoala

AmoCoala is based on the same ABC-SMC method as the one developed in Coala (Baudet *et al.*, 2015). For the sake of completeness, we now recall the procedure.

The ABC-SMC procedure is composed of a sequence of $R > 1$ rounds. At each round, parameter vectors θ are sampled in a specific way, symbiont trees $\tilde{S}_\theta$ are generated under the reconciliation model allowing for spreads with parameter values given by θ (and relying on the simulation algorithm described in the previous section). Then, these symbiont trees are compared to the original dataset through a summary distance d whose details are given in the next section. The parameters with the smallest discrepancies are selected.

For each of these rounds, we define a tolerance value τ_r ($1 \leq r \leq R$) which determines the percentage of parameter vectors to be accepted. Associated with a tolerance value τ_r, we have a threshold ϵ_r which is the largest value of the summary distance associated with the accepted parameter vectors.

- Initial round ($r = 1$):
  - Draw an initial set of N parameter vectors $\{\theta_1^i\}_{(1\leq i\leq N)}$ from the prior π.
  - Then, for each θ_1^i, simulate M trees $\{\tilde{S}_j(\theta_1^i)\}_{(1\leq j\leq M)}$. Compute the corresponding discrepancies $\{d_j(\theta_1^i)\}_{(1\leq j\leq M)}$ and summarise them into the summary discrepancy $d_{\theta_1^i}$ through the mean value.
  - Select $Q_1 = \tau_1 \times N$ parameter vectors θ_1 that have the smallest value d_{θ_1}, thus defining the threshold ϵ_1 and the set A_1 of accepted parameter vectors.

- Following rounds ($2 \leq r \leq R$):
  1. Sample a parameter vector $\theta^\star$ from the set $A_{(r-1)}$.
  2. Create a parameter vector $\theta^{\star\star}$ by perturbing $\theta^\star$ (through a kernel proposal).
  3. Simulate M trees relying on the parameter value $\theta^{\star\star}$ and compute $d_{\theta^{\star\star}}$. If $d_{\theta^{\star\star}} \leq \epsilon_{(r-1)}$, add $\theta^{\star\star}$ into the quantile set $\mathcal{Q}_r$. If $|\mathcal{Q}_r| < Q_{r-1}$, return to Step 1.
  4. Based on the set $\mathcal{Q}_r$, select $Q_r = \tau_r \times Q$ parameter vectors θ_r that have the smallest d_{θ_r}, thus defining the threshold ϵ_r and the set A_r of accepted parameters.

Prior distribution. We sample from a uniform distribution on the simplex $\mathcal{S}_3 = \{(p_1, p_2, p_3, p_4); p_i \geq 0 \text{ and } \sum_i p_i = 1\}$ (we recall that $p_c + p_d + p_s + p_l = 1$).

Kernel proposal. We add to each coordinate of θ a randomly chosen value in $[-0.01, +0.01]$ and normalise the result. The final set of accepted parameter vectors is the result of the ABC-SMC procedure and characterises the list of vectors that may explain the evolution of the pair of host and symbiont trees given as input. Observe that, since in all experiments a uniform prior distribution is assumed and also the perturbations are performed in a uniform way, the weights induced by the proposals will also appear to be uniform (Beaumont *et al.*, 2009). However, in the case of a different prior, weights should be used in the process in order to correct the posterior distribution according to the perturbation made.

Clustering of the vectors. The final list of accepted vectors are clustered using a hierarchical clustering procedure implemented in Coala (Baudet *et al.*, 2015). As final result, we therefore obtain a list of clusters to each one of which a representative vector is associated.

B.3 Distance measure in AmoCoala

The discrepancy between the simulated and the original datasets is measured through a distance between set-labelled phylogenetic trees which can be calculated in polynomial time. Similarly as in Coala, this distance contains

two components: (i) d_1, that describes how much the simulated tree $\tilde{S}_\theta$ is representative of the vector θ, and (ii) d_2 that measures how much is $\tilde{S}_\theta$ (and its labels) topologically similar to S (and its labels).

Let us recall the definition of this first component. For a given vector $\theta = \langle p_c, p_d, p_s, p_l \rangle$ and for each simulated tree $\tilde{S}_\theta$ that was simulated according to this vector, we keep track of the vector of the number of classical cophylogeny events $\langle o_c, o_d, o_s, o_l \rangle$ associated to this simulation. We compute the corresponding expected vector $\langle e_c, e_d, e_s, e_l \rangle$ as follows

$$\forall \text{event} \in \{c, d, s, l\}, \quad e_{\text{event}} = |S| \times \theta_{\text{event}} = |S| \times p_{\text{event}},$$

where $|S|$ is the size of the symbiont tree, *i.e.* its number of internal leaves. Then by comparing the observed and expected vectors, we define a measure $d_1(S, \tilde{S}_\theta)$ as follows:

$$d_1(S, \tilde{S}_\theta) = \frac{1}{4} \times \sum_{\text{event} \in \{c,d,s,l\}} \frac{|e_{\text{event}} - o_{\text{event}}|}{\max\{e_{\text{event}}, o_{\text{event}}\}}.$$

Note that we did not consider the number of observed spread events, which does not depend on the choice of θ as the corresponding probabilities are pre-estimated before applying the ABC-SMC approach.

As concerns point (ii), we extend the well-known *maximum agreement subtree* (MAST) distance (Finden and Gordon, 1985; Farach-Colton *et al.*, 1995) to handle set-labelled trees. This part is the novelty with respect to the proposal in Coala and details were given in the Main Manuscript. We establish in the next sections that d_{MASST} is a distance and that it can be computed in polynomial time.

We use a normalised version of d_{MASST} and define the distance d_2 (see Main Manuscript). The two components are then combined to form the following distance

$$d_\theta = \alpha_1 d_1(S, \tilde{S}_\theta) + \alpha_2 d_2(S, \tilde{S}_\theta).$$

According to our experiments and also the ones presented in Coala, the most appropriate values are $\alpha_1 = 0.7$ and $\alpha_2 = 0.3$.

B.4 A proof that dMASST is a distance

We show that the distance d_{MASST} is a metric. For this, we check that d_{MASST} satisfies the following properties:

1. $d_{MASST}(T_1, T_2) \geq 0$ for all T_1,T_2: this is trivial.

2. $d_{MASST}(T_1, T_2) = 0$ if and only if $T_1 = T_2$. Clearly if $T_1 = T_2$ then $d_{MASST}(T_1, T_2) = 0$. Otherwise, let $d_{MASST}(T_1, T_2) = 0$. Then $\max\{w(T_1), w(T_2)\} = MASST(T_1, T_2)$. The proof follows by observing that if T^* is a subtree of T such that $w(T^*) = w(T)$ then $T^* = T$.

3. $d_{MASST}(T_1, T_2) = d_{MASST}(T_2, T_1)$: this is trivial.

4. For any triplet of trees T_1, T_2, T_3, it holds that $d_{MASST}(T_1, T_2) + d_{MASST}(T_2, T_3) \geq d_{MASST}(T_1, T_3)$. For simplicity, we set $w_i = w(T_i)$ and $w_{i,j} = w(MASST(T_i, T_j))$. Hence $d_{MASST}(T_i, T_j) = \max\{w_i, w_j\} - w_{i,j}$. Furthermore, we denote by $w_{1,2,3}$ the weight of the maximum agreement subtree that is common to the three trees T_1, T_2, T_3. We then have:

$$
\begin{aligned}
&d_{MASST}(T_1, T_2) + d_{MASST}(T_2, T_3) \\
&= \max\{w_1, w_2\} - w_{1,2} + \max\{w_2, w_3\} - w_{2,3} \\
&= \max\{w_1, w_2\} + \max\{w_2, w_3\} - (w_{1,2} + w_{2,3} - w_{1,2,3} + w_{1,2,3}) \\
&\geq \max\{w_1, w_2, w_3\} + w_2 - (w_2 + w_{1,2,3}) \\
&\geq \max\{w_1, w_3\} - w_{1,3},
\end{aligned}
$$

   where for the first inequality, we use the fact that $\max\{w_1, w_2\} + \max\{w_2, w_3\} \geq \max\{w_1, w_2, w_3\} + w_2$ and we show in the next Lemma that $w_{1,2} + w_{2,3} - w_{1,2,3}$ is at most w_2. The last inequality uses $w_{1,2,3} \leq w_{1,3}$.

This concludes the proof.

Lemma. *For any three set-labelled trees T_1, T_2, T_3 (using the notation from the above proof) it holds that $w_{1,2} + w_{2,3} - w_{1,2,3} \leq w_2$.*

Proof. Let $T_{1,2}$ and $T_{2,3}$ be maximum agreement set-labelled subtrees (MASST) of T_1, T_2 and T_2, T_3, respectively. Consider any pair of leaf, label that belongs to T_2, *i.e.* $(l, lab) \in T_2$. There are only four possibilities: (i) $(l, lab) \in T_{1,2}$ and $(l, lab) \notin T_{2,3}$ (we call these leaves of type A), (ii) $(l, lab) \notin T_{1,2}$ and $(l, lab) \in T_{2,3}$ (we call these leaves of type B), (iii) $(l, lab) \in T_{1,2}$ and $(l, lab) \in T_{2,3}$ (we call these leaves of type C), (iv) $(l, lab) \notin T_{1,2}$ and

$(l, lab) \notin T_{2,3}$ (we call these leaves of type D). Then we have

$$\begin{aligned} w_2 &= |A| + |B| + |C| + |D| \\ &= w_{12} - |C| + w_{23} - |C| + |C| + |D| \\ &= w_{12} + w_{23} - |C| + |D|. \end{aligned}$$

Or equivalently

$$w_{12} + w_{23} = w_2 + |C| - |D|. \tag{B.1}$$

Moreover, we define the tree $\tilde{T}$ as the subtree obtained from T_2 by taking all the pairs of leaf, label that belong to T_{12} and T_{23}. Notice that $\tilde{T}$ is also a subtree of T_1 and of T_3. Thus, $\tilde{T}$ is included in T_{123}. This implies that $|C| \leq w_{123}$. Going back to (B.1), we thus obtain

$$\begin{aligned} w_{12} + w_{23} &= w_2 + |C| - |D| \\ &\leq w_2 + |C| \\ &\leq w_2 + w_{123}. \end{aligned}$$

This concludes the proof of the lemma. □

Remark. *The previous proof and comments show that the MASST distance d_{MASST} is very similar to the MAAC one (Ganapathy* et al.*, 2005) for multi-labelled trees. Thus, it is natural to ask whether comparing two set-labelled trees can be reduced to comparing two multi-labelled trees. One idea is to transform a set-labelled tree into a multi-labelled tree. However, the straightforward transformation seems not to work well for our purpose. For instance, we can transform each set-labelled tree into a multi-labelled tree by substituting each set-labelled leaf by a subtree with a fixed topology (say a complete binary tree, or a multifurcating vertex) as in Figure A. However, in these cases the two trees in Figure A would be considered equivalent, but in our context they are different. In fact, the set-labelled tree in Figure A(a) indicates that there is a symbiont that infects 4 different hosts h_1, h_2, h_3, h_4, while in Figure A(b), we will have 4 different symbionts infecting each a different host.*

B.5 Polynomial time algorithm for computing the dMASST distance

We show that it is possible to calculate the distance $d_{MASST}(T_1, T_2)$ in polynomial time with respect to the size of the trees. This boils down to computing the weight of the maximum agreement subtree $w(MASST(T_1, T_2))$

in polynomial time. The algorithm is based on dynamic programming and extends quite straightforwardly the algorithm for calculating the MAAC distance (Ganapathy *et al.*, 2005). We abbreviate to $w(v_1, v_2)$ the weight of the maximum agreement subtree between the two trees T_1 and T_2 rooted in v_1 and v_2, respectively. For a leaf v, we denote by $l(v)$ the set of labels associated with it. Finally, for an internal vertex v, we denote by $ch_1(v)$ and $ch_2(v)$ the two children of v.

The dynamic programming algorithm starts from the leaves and ends in the roots of T_1 and T_2 following a recursion. We have that $w(v_1, v_2)$ is given by:

- If v_1 and v_2 are both leaves then $w(v_1, v_2) = |l(v_1) \cup l(v_2)|$
- If v_1 or v_2 (could be both) are internal vertices, $w(v_1, v_2)$ is the maximum value among the following three quantities
  1. $\max\{w(ch_1(v_1), v_2), w(ch_2(v_1), v_2)\}$;
  2. $\max\{w(v_1, ch_1(v_2)), w(v_1, ch_2(v_2)\}$;
  3. $\max\{w(ch_1(v_1), ch_1(v_2)) + w(ch_2(v_1), ch_2(v_2)),\ w(ch_1(v_1), ch_2(v_2))$ $+w(ch_2(v_1), ch_1(v_2))\}$.

C Additional results for the self-test

The results for parameter values $\theta_2^\star$ to $\theta_8^\star$ are presented in Figures Ba to Cd.

D Biological datasets

We provide here a description of the 4 datasets used. The corresponding phylogenetic trees are shown in Figures D - G.

Dataset 1*: AP - Acacia & Pseudomyrmex.* This dataset was extracted from Gómez-Acevedo *et al.* (2010) and displays the interaction between *Acacia* plants and *Pseudomyrmex* species of ants. The host and symbiont trees include 9 and 7 leaves, respectively. The dataset has 22 multiple-associations.

Dataset 2*: MP - Myrmica & Phengaris.* This dataset was extracted from Jansen *et al.* (2011) and is composed of a pair of host and symbiont trees which have each 8 leaves. The dataset has 8 multiple-associations.

Dataset 3*: SBL - Seabirds & Lice.* This dataset was extracted from Paterson *et al.* (1997). The host and symbiont trees include 15 and 8 leaves, respectively. The dataset has 15 multiple-associations.

Dataset 4*: SFC - Smut Fungi & Caryophillaceus plants.* This dataset was extracted from Refrégier *et al.* (2008). The host and symbiont trees include 15 and 16 leaves, respectively. The dataset has 4 multiple-associations.

D.1 Results on biological datasets

We ran AmoCoala on all the real datasets and plotted in Figures H to S the histograms of the summary discrepancies and event probabilities (except for the spread probabilities which are not inferred) obtained at the end of each one of the 3 rounds, for each of the 4 datasets. We see on the histograms that the summary discrepancies for the accepted parameter vectors decrease after each round. We recall that the summary discrepancy measures the similarity between the simulated trees and the original symbiont tree, and hence is related to the quality of the vectors. Thus, our result shows that the set of accepted vectors is refined at each round, leading to vectors which can generate trees that are increasingly more similar to the original symbiont tree (and its host associations).

D.2 Running times

Table D.1 shows the running times obtained on the 4 biological datasets, together with their sizes (as expressed by the number of leaves in the host and symbiont trees) and the number of multiple associations. The results have been obtained on a computer with a AMD EPYC 7542 32-Core processor and 128 CPU (2 sockets of 32 double threads cores) and 675Gb RAM. We used just one core ('nthreads 1', though AmoCoala has a parallelized version) and AmoCoala was run with default values on these datasets.

We also performed an artificial experiment on a host tree with 204 leaves, a symbiont tree with 128 leaves, and six multiple associations. Relying on the above machine and using now 60 threads (which might not have been fully used during the entire computation), the running time of AmoCoala (used with default options except for the number of initial vectors N that was set to 1000) was approximately 27.5 hours.

Dataset	(Host,Symbiont) leaves	Multiple associations	Running time
AP	(9,7)	22	23m20.859s
MP	(8,8)	8	21m25.631s
SBL	(15,8)	15	28m53.597s
SFC	(15,16)	4	117m45.919s

Table D.1: For each of the 4 biological datasets, we indicate the pairs of numbers of host and symbiont trees leaves (2nd column), the number of multiple associations (3rd column) and the running time of AmoCoala on this dataset (4th column).

D.3 Robustness analysis wrt the pre-estimated spread probabilities

In this section, we explore the robustness of our results with respect to the pre-estimated values of the spread events probabilities. On each of the 4 biological datasets, we ran AmoCoala with perturbated values of $p_{\text{hs}}(h), p_{\text{vs}}(h)$. More precisely, to each non zero probability $p_{\text{hs}}(h)$ or $p_{\text{vs}}(h)$, we added a noise value uniformly drawn in $[-0.1; 0.1]$ (and then took the infimum with 1 and the supremum with 0, in order to ensure the modified probabilities remain in $[0, 1]$). With these perturbed values, we ran AmoCoala and output (after 3 rounds) 50 accepted vectors $\theta = \langle p_c, p_d, p_s, p_l \rangle$. The results are presented in Figures T to W. Let us recall that AmoCoala is a stochastic algorithm and any two runs will give similar but not identical results. The results obtained adding these perturbations are qualitatively the same for the first 3 datasets (namely AP, MP and SBL) as the ones without perturbations (see Figures J to P). The results for dataset SFC show more variability wrt those of the unperturbed version (Figure S). Thus we also looked at the clusters output by AmoCoala in this case in Table D.2. We recall that in Refrégier *et al.* (2008), the different analyses performed indicated that the most plausible reconciliations presented for the SFC dataset have from 0 to 3 cospeciations, no duplication, 12 to 15 host switches and 0 to 2 losses. Here we find that the first main cluster (31 vectors out of 50) has a representative vector with around 50% of cospeciations (about 7 or 8 events), almost no duplication (about 0 or 1 event), 31% of host switches (about 4 or 5 events) and 18% of losses (about 2 or 3 losses). The second main cluster has a higher probability of cospeciation and less switches. Only the third cluster could correspond to Refrégier *et al.* (2008)'s scenario, with 1 or 2 cospeciations, no duplication, 8

or 9 host switches and 4 to 5 losses; though it is supported by only 3 selected vectors out of 50. Thus for the SFC dataset, the detection of the biological scenario presented in Refrégier *et al.* (2008) is more difficult to detect with perturbed values of the spread probabilities. To conclude, our results are overall robust with respect to potential errors in the estimation of the spread events probabilities.

References

Baudet, C., Donati, B., Sinaimeri, B., Crescenzi, P., Gautier, C., Matias, C., and Sagot, M.-F. 2015. Cophylogeny reconstruction via an Approximate Bayesian Computation. *Systematic Biology*, 64(3): 416–31.

Beaumont, M. A., Cornuet, J.-M., Marin, J.-M., and Robert, C. P. 2009. Adaptive approximate Bayesian computation. *Biometrika*, 96: 983–990.

Charleston, M. A. 2002. *Biological Evolution and Statistical Physics*, volume 585 of *Lecture Notes in Physics*, chapter Principles of cophylogenetic maps, pages 122–147. Springer Berlin Heidelberg.

Farach-Colton, M., Przytycka, T. M., and Thorup, M. 1995. On the agreement of many trees. *Inform. Process. Lett.*, 55: 297–301.

Finden, C. R. and Gordon, A. D. 1985. Obtaining common pruned trees. *J. Classif.*, 2: 255–276.

Ganapathy, G., Goodson, B., Jansen, R., Ramachandran, V., and Warnow, T. 2005. Pattern Identification in Biogeography. In R. Casadio and G. Myers, editors, *Algorithms in Bioinformatics*, volume 3692 of *Lecture Notes in Computer Science*, pages 116–127. Springer Berlin Heidelberg.

Gómez-Acevedo, S., Rico-Arce, L., Delgado-Salinas, A., Magallón, S., and Eguiarte, L. E. 2010. Neotropical mutualism between Acacia and Pseudomyrmex: Phylogeny and divergence times. *Molecular Phylogenetics and Evolution*, 56(1): 393–408.

Jansen, G., Vepsäläinen, K., and Savolainen, R. 2011. A phylogenetic test of the parasite-host associations between *Maculinea* butterflies (Lepidoptera: Lycaenidae) and *Myrmica* ants (Hymenoptera: Formicidae). *European Journal of Entomology*, 108(1): 53–62.

Paterson, A., Gray, R. D., Clayton, D. H., and Moore, J. 1997. Host-parasite co-speciation, host switching, and missing the boat. In D. H. Clayton and J. Moore, editors, *Host-parasite evolution: General principles and avian models*, pages 236–250. Oxford University Press.

Refrégier, G., Le Gac, M., Jabbour, F., Widmer, A., Shykoff, J. A., Yockteng, R., Hood, M. E., and Giraud, T. 2008. Cophylogeny of the anther smut fungi and their caryophyllaceous hosts: Prevalence of host shifts and importance of delimiting parasite species for inferring cospeciation. *BMC Evolutionary Biology*, 8(1): 100.

Stolzer, M. L., Lai, H., Xu, M., Sathaye, D., Vernot, B., and Durand, D. 2012. Inferring duplications, losses, transfers and incomplete lineage sorting with nonbinary species trees. *Bioinformatics*, 28(18): i409–i415.

Tofigh, A., Hallett, M. T., and Lagergren, J. 2011. Simultaneous identification of duplications and lateral gene transfers. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 8(2): 517–535.

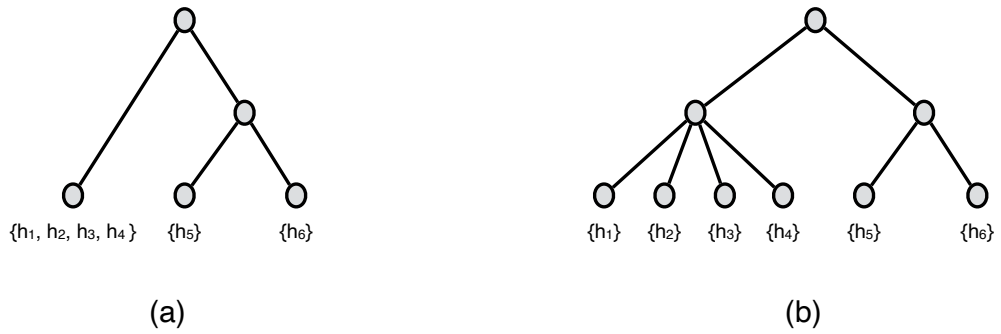
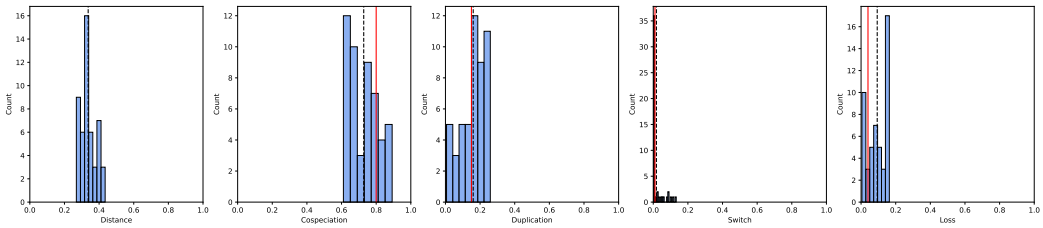
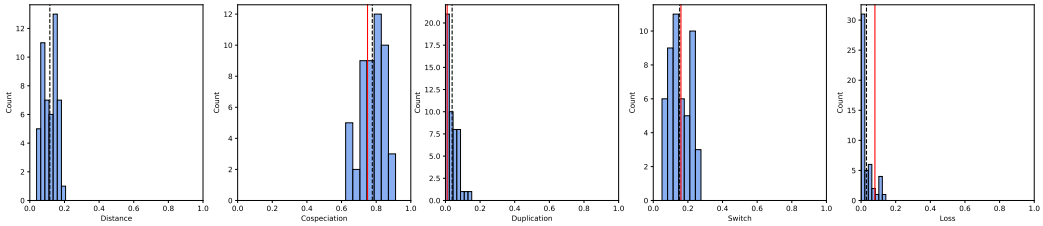


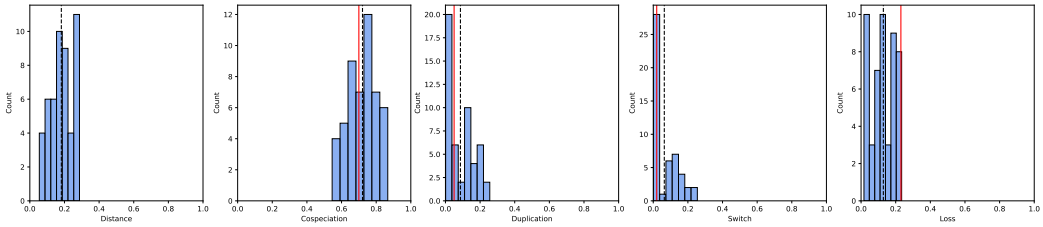
Figure A: The two phylogenetic trees will be considered at distance 0 if we substitute the vertex labelled by the set h_1, h_2, h_3, h_4 by a multifurcated vertex.



(a) Results for $\theta_2^\star = \langle 0.80, 0.15, 0.01, 0.04 \rangle$.

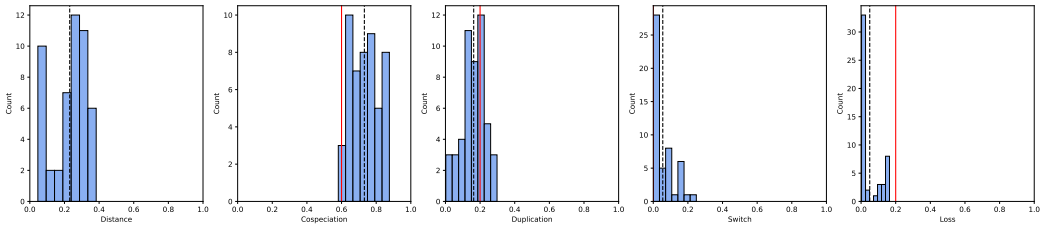


(b) Results for $\theta_3^\star = \langle 0.75, 0.01, 0.16, 0.08 \rangle$

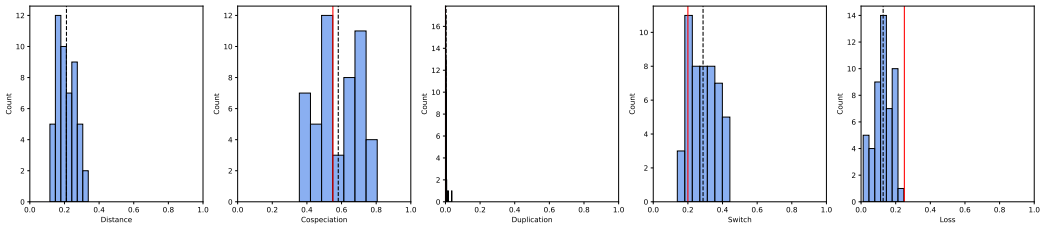


(c) Results for $\theta_4^\star = \langle 0.70, 0.05, 0.02, 0.23 \rangle$.

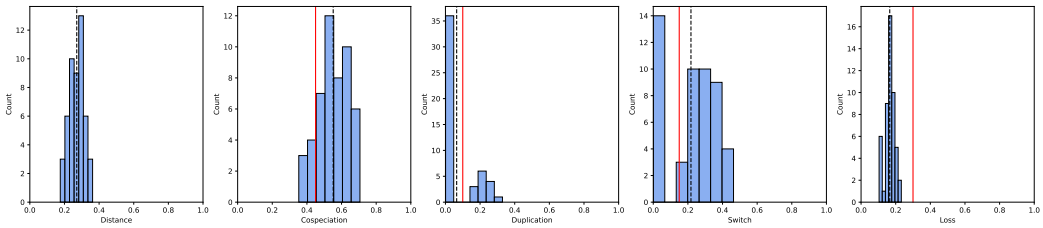
Figure B: For each simulated dataset with true parameter value $\theta_i^\star$ and $2 \leq i \leq 8$, we ran AmoCoala 50 times and, at the end of the third round, we took note of the cluster whose representative parameter vector had the smallest euclidean distance (histograms shown in the first column) to $\theta_i^\star$. Columns 2 to 5 show the histograms of the distributions of the event probabilities in these "best" clusters. The dashed vertical black line indicates the mean value. The solid vertical red line indicates the true parameter value.



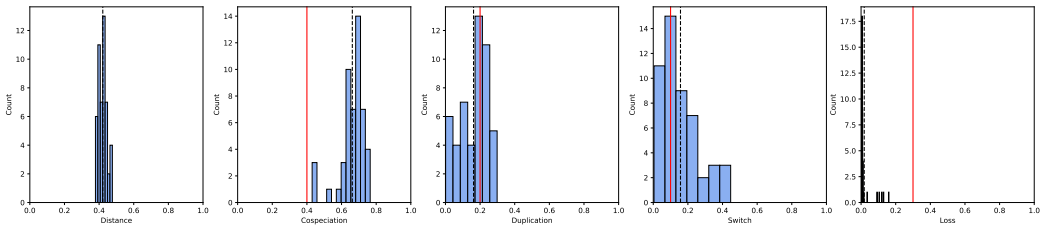
(a) Results for $\theta_5^\star = \langle 0.60, 0.20, 0.00, 0.20 \rangle$



(b) Results for $\theta_6^\star = \langle 0.55, 0.00, 0.20, 0.25 \rangle$.



(c) Results for $\theta_7^\star = \langle 0.45, 0.10, 0.15, 0.30 \rangle$



(d) Results for $\theta_8^\star = \langle 0.40, 0.20, 0.10, 0.30 \rangle$.

Figure C: For each simulated dataset with true parameter value $\theta_i^\star$, we ran AmoCoala 50 times and, at the end of the third round, we took note of the cluster whose representative parameter vector had the smallest euclidean distance (histograms shown in the first column) to $\theta_i^\star$. Columns 2 to 5 show the histograms of the distributions of the event probabilities in these "best" clusters. The dashed vertical black line indicates the mean value. The solid vertical red line indicates the true parameter value.

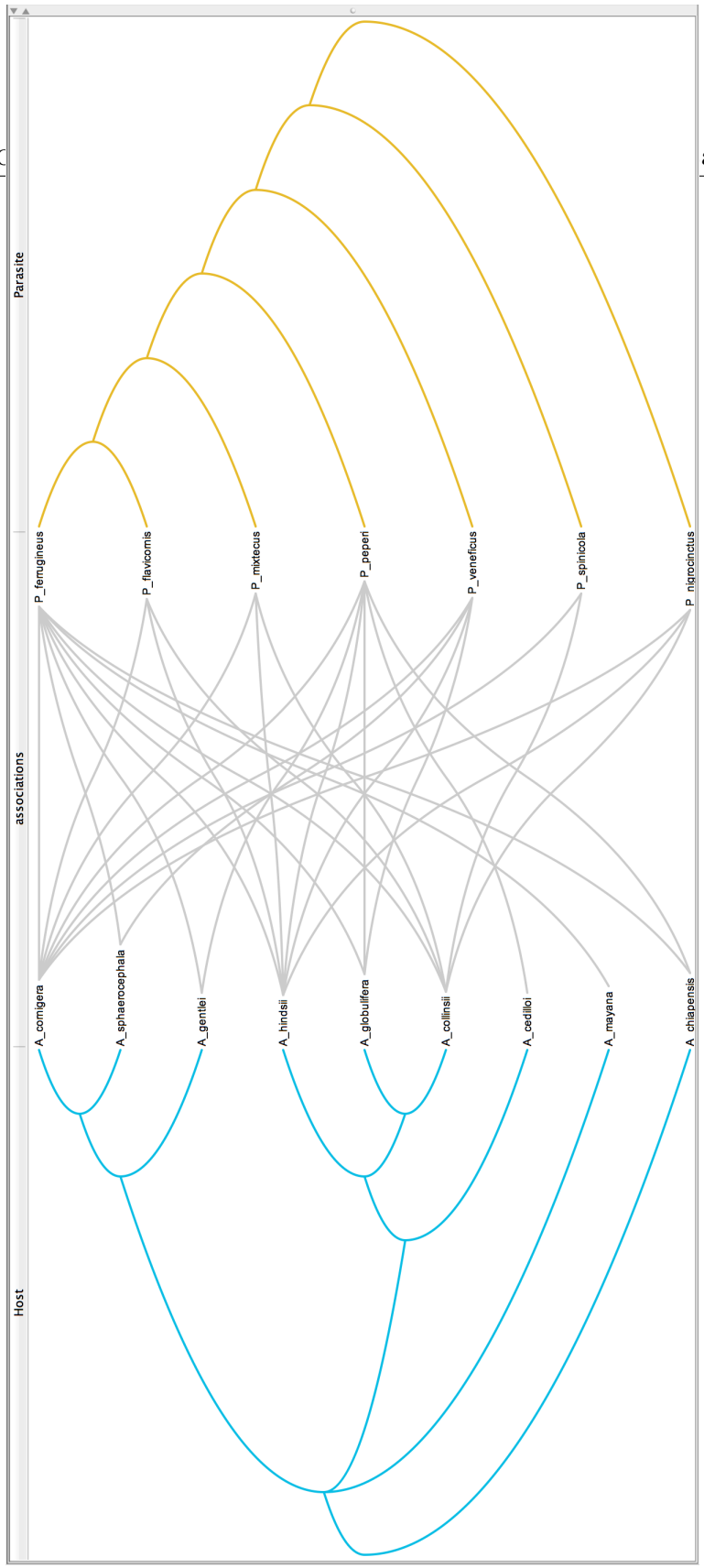


Figure D: AP dataset.

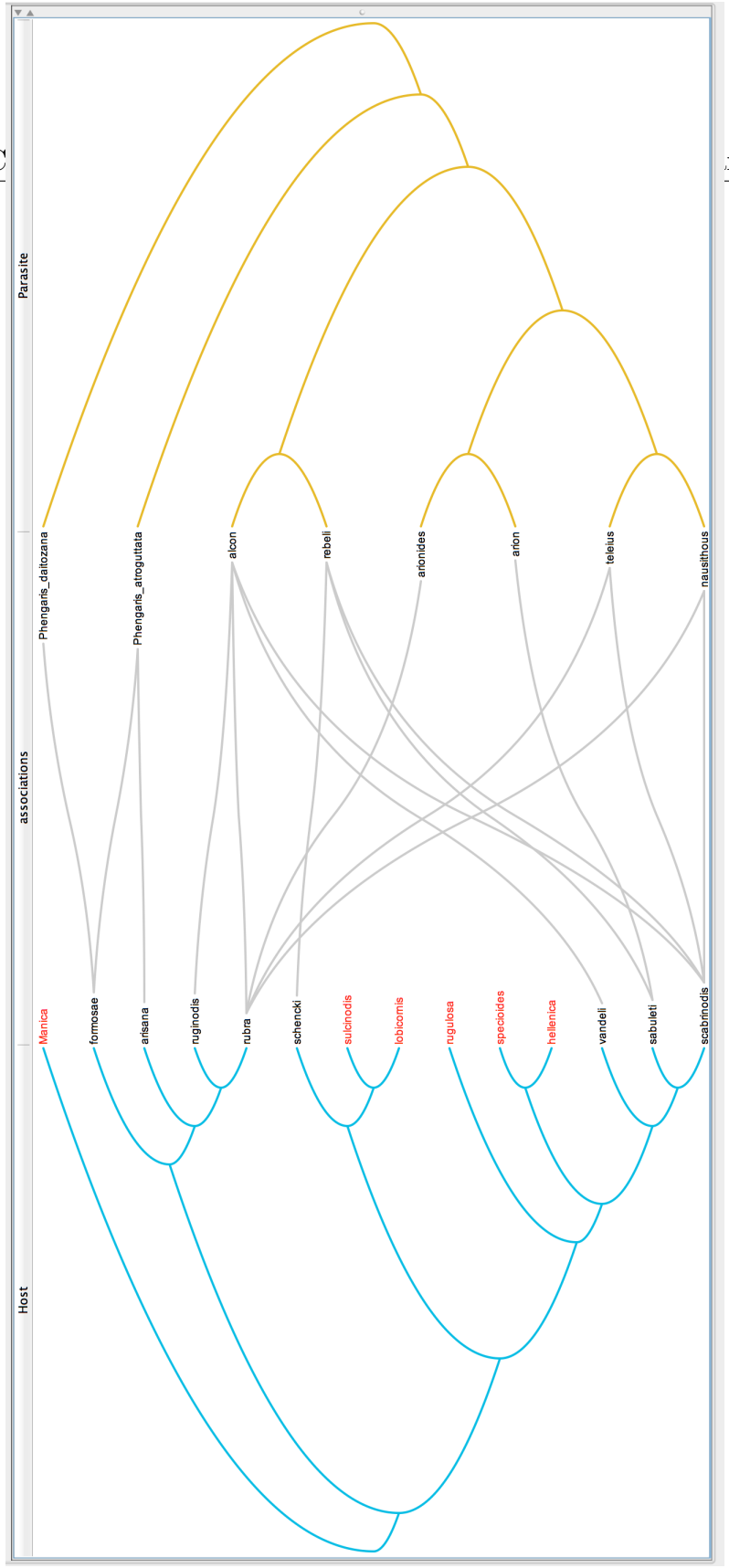


Figure E: MP dataset.

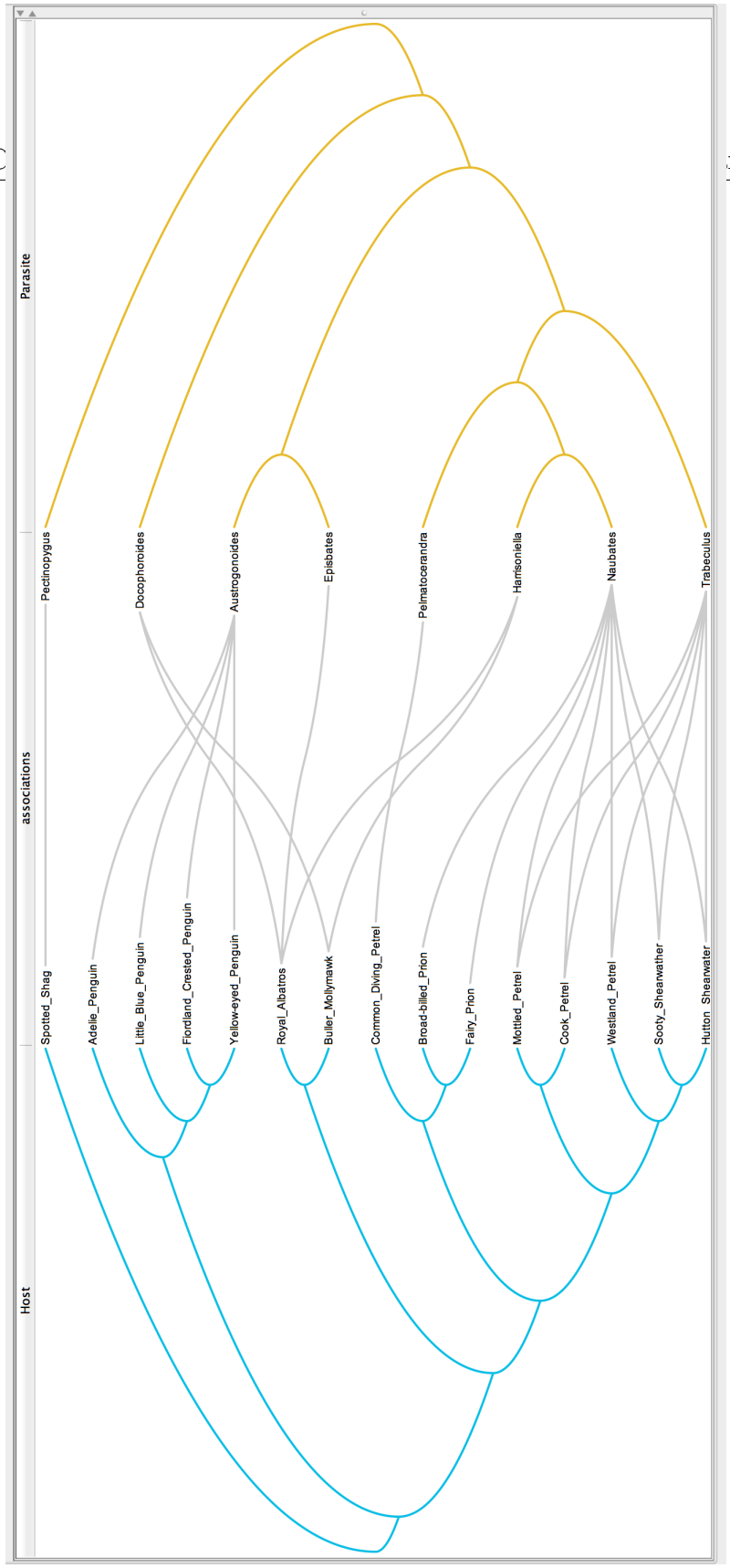


Figure F: SBL dataset.

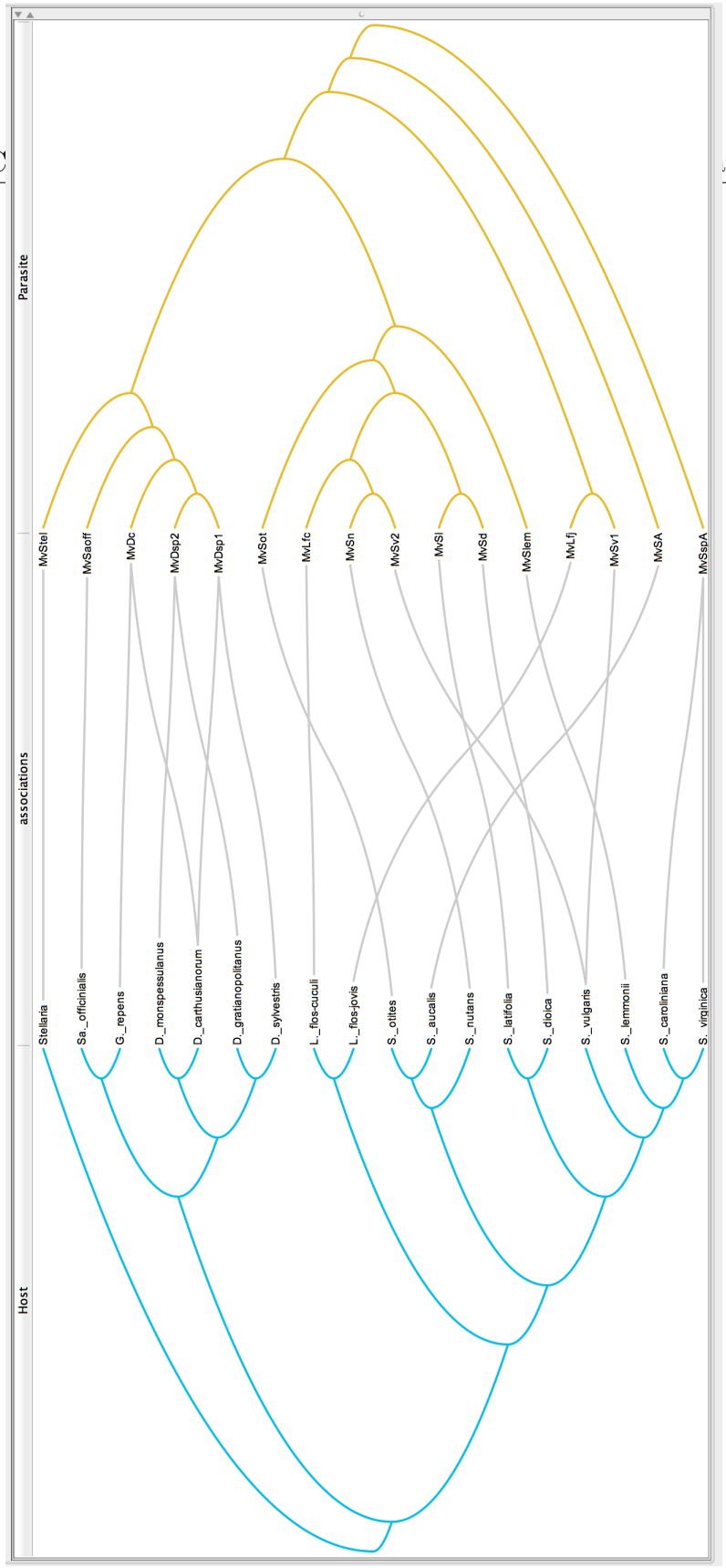


Figure G: SFC dataset.

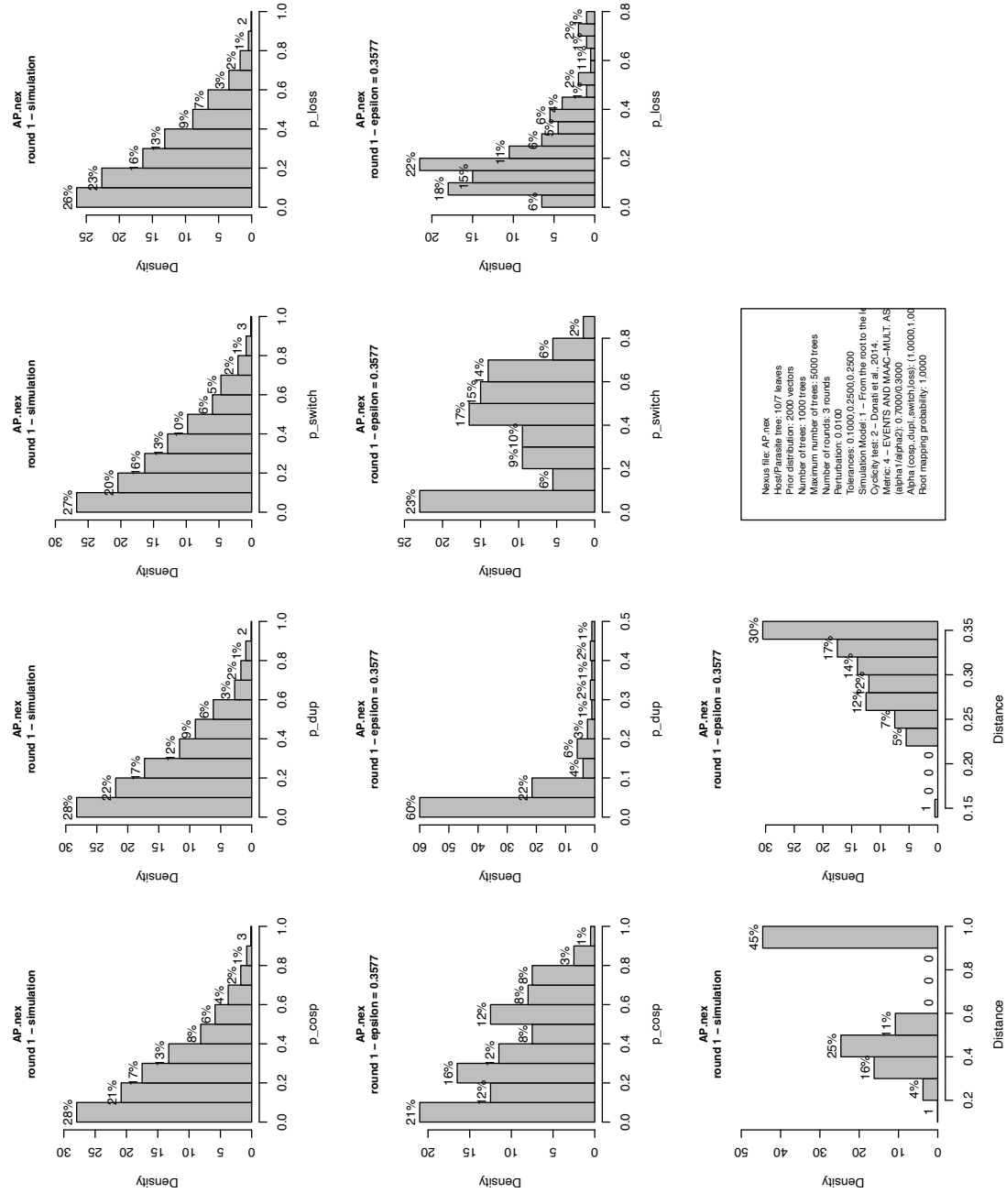


Figure H: AP dataset. First row: histograms of the input parameters. Second row: histograms of the parameters after round 1. Third row: summary discrepancies of the input parameters and of the parameters after round 1.

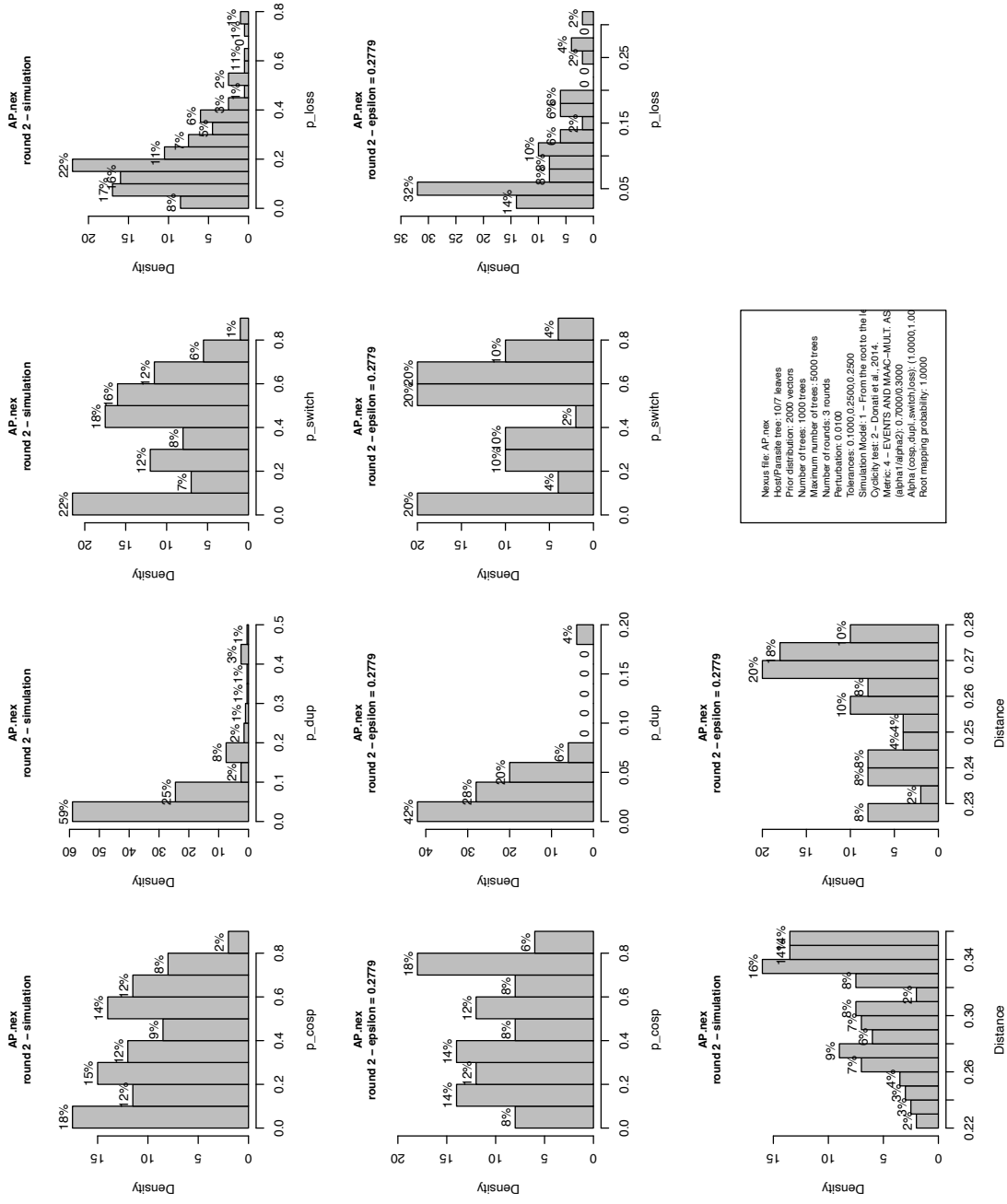


Figure I: AP dataset. First row: histograms of the input parameters. Second row: histograms of the parameters after round 2. Third row: summary discrepancies of the input parameters and of the parameters after round 2.

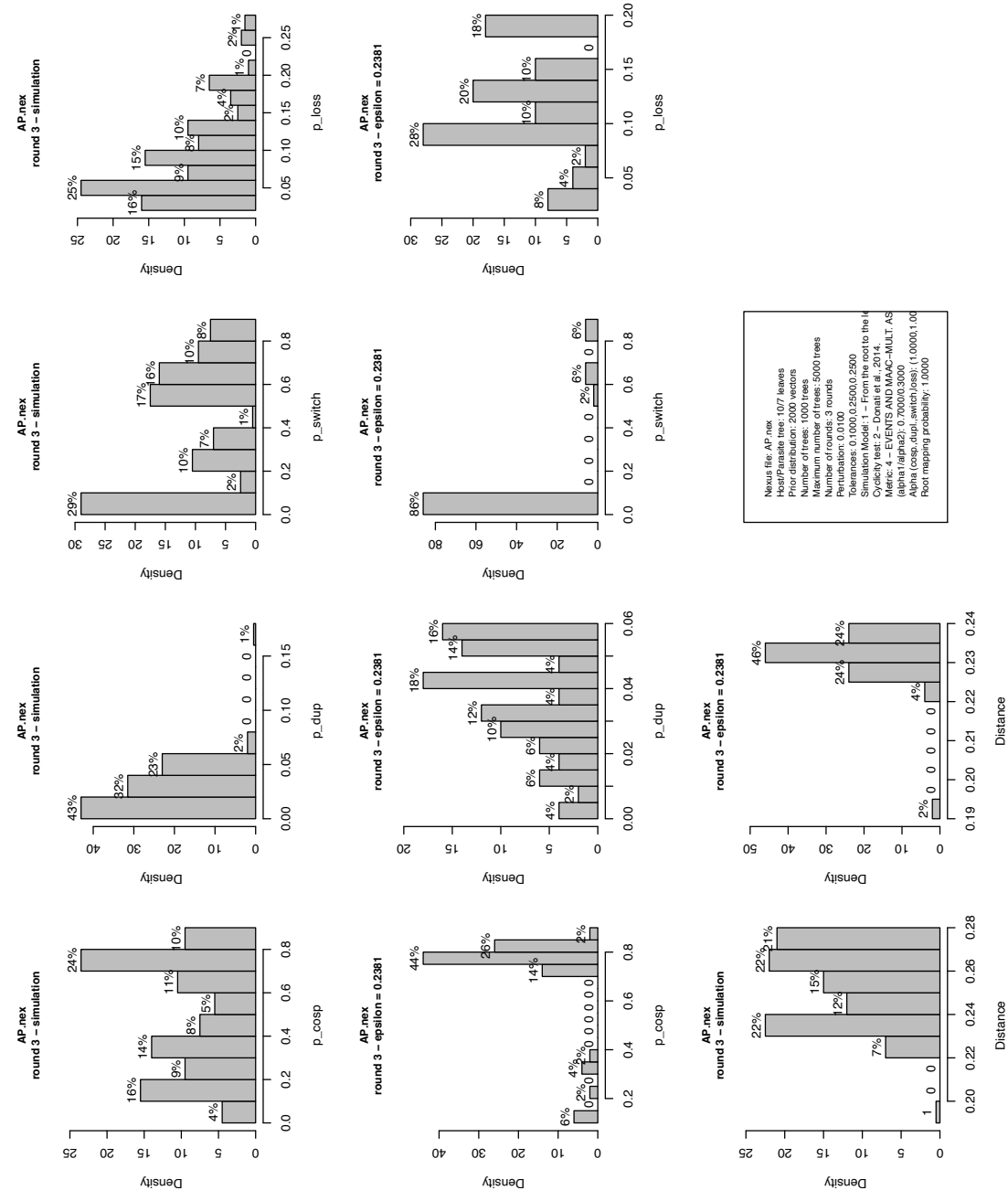


Figure J: AP dataset. First row: histograms of the input parameters. Second row: histograms of the parameters after round 3. Third row: summary discrepancies of the input parameters and of the parameters after round 3.

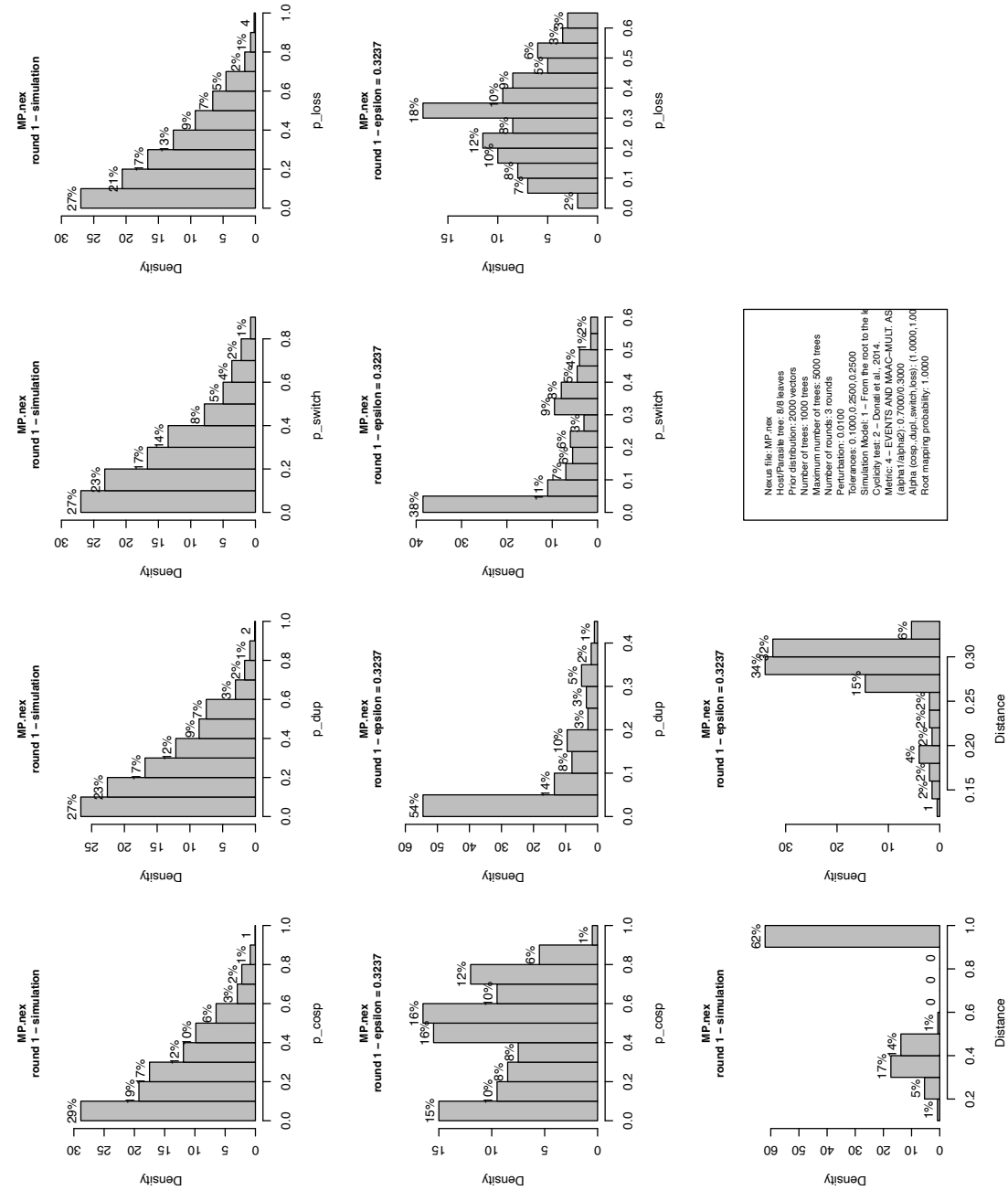


Figure K: MP dataset. First row: histograms of the input parameters. Second row: histograms of the parameters after round 1. Third row: summary discrepancies of the input parameters and of the parameters after round 1.

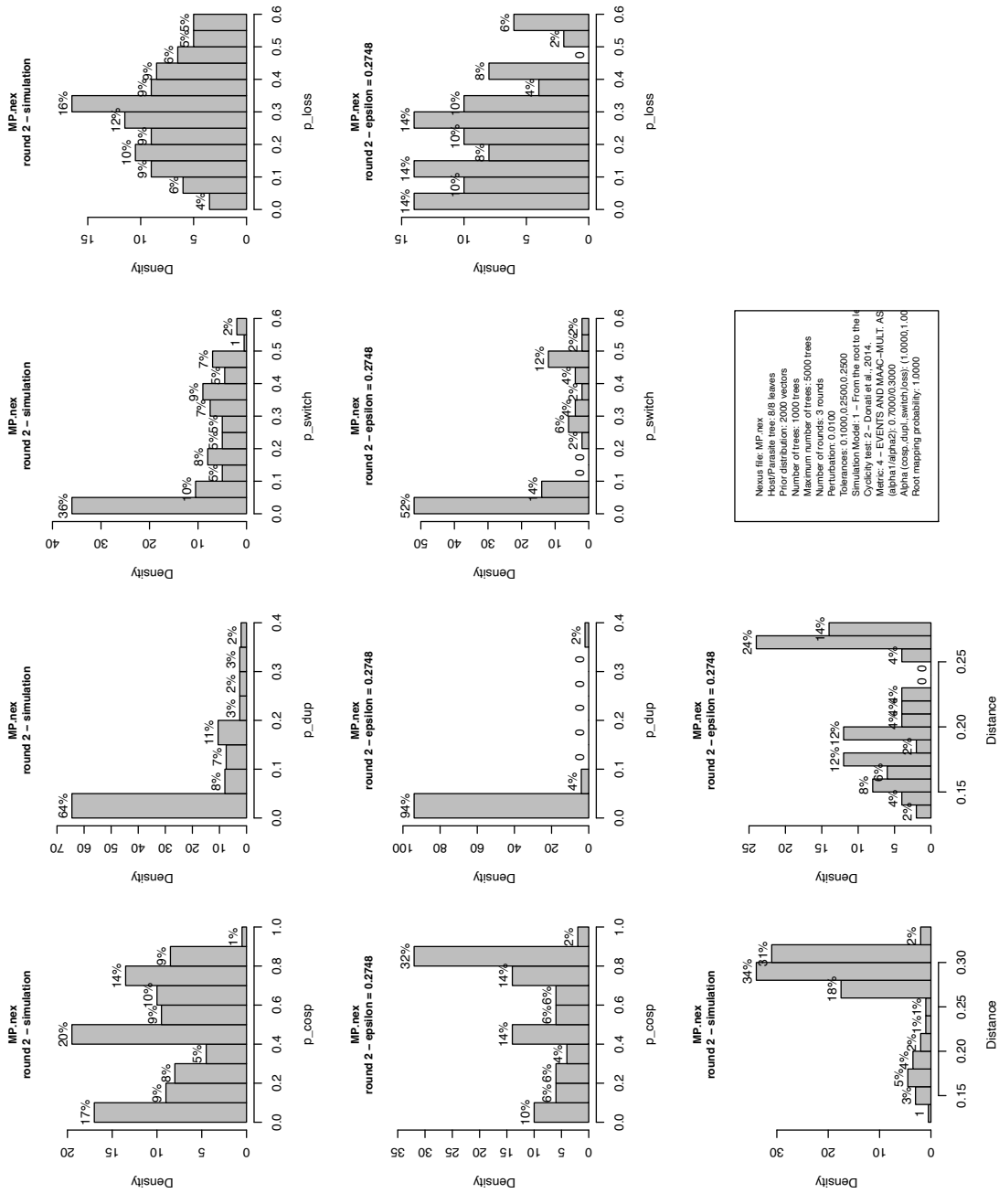


Figure L: MP dataset. First row: histograms of the input parameters. Second row: histograms of the parameters after round 2. Third row: summary discrepancies of the input parameters and of the parameters after round 2.

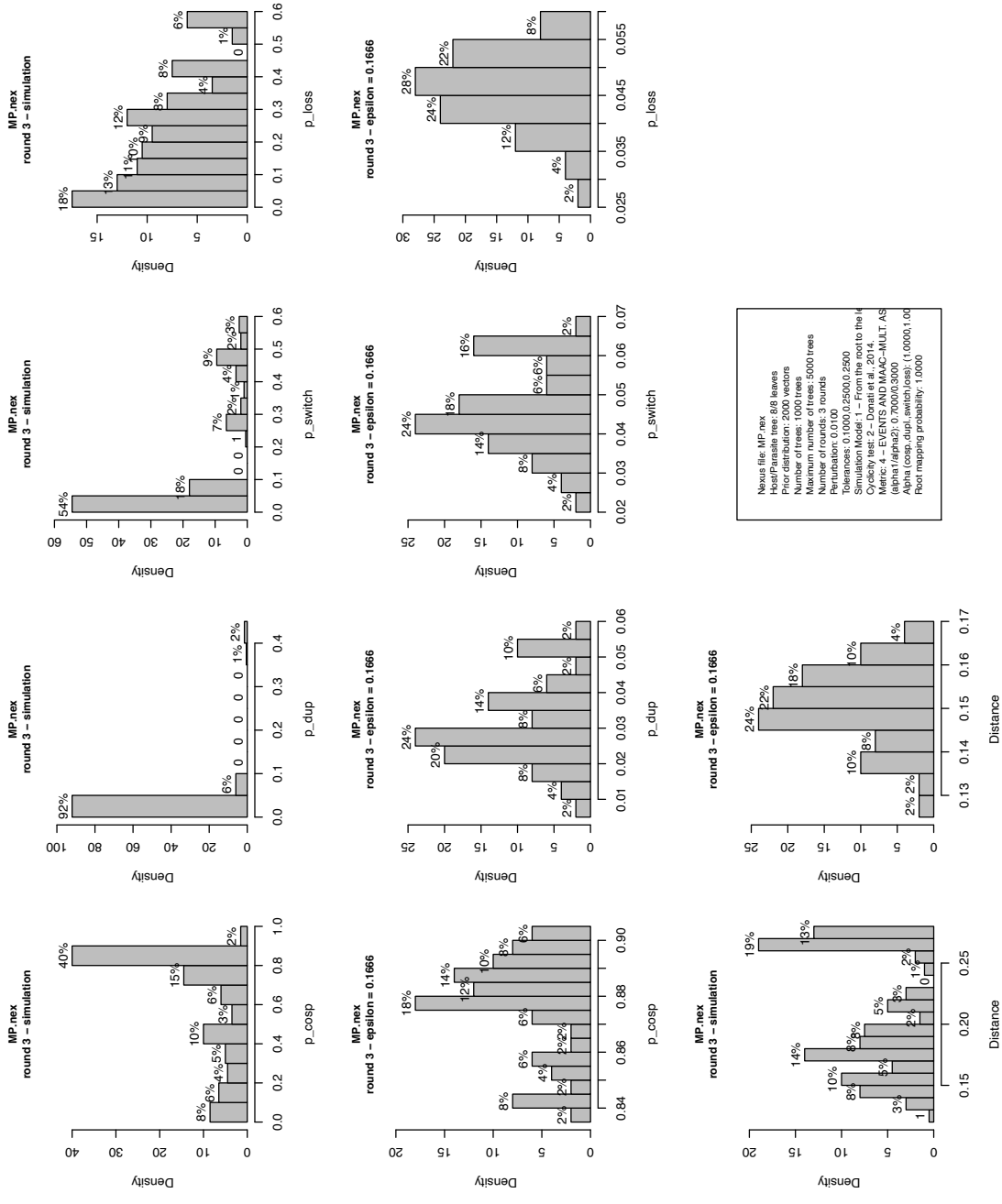


Figure M: MP dataset. First row: histograms of the input parameters. Second row: histograms of the parameters after round 3. Third row: summary discrepancies of the input parameters and of the parameters after round 3.

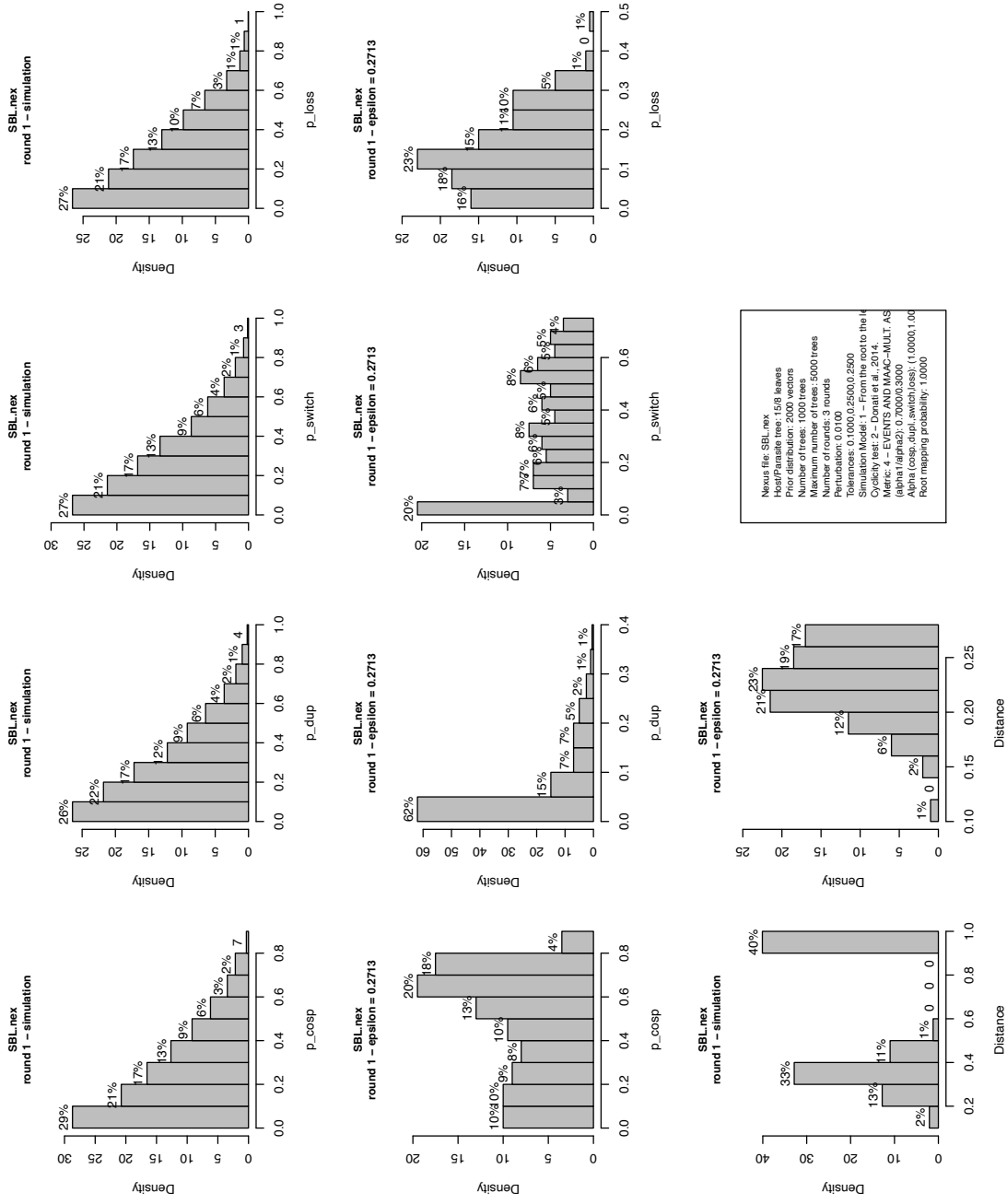


Figure N: SBL dataset. First row: histograms of the input parameters. Second row: histograms of the parameters after round 1. Third row: summary discrepancies of the input parameters and of the parameters after round 1.

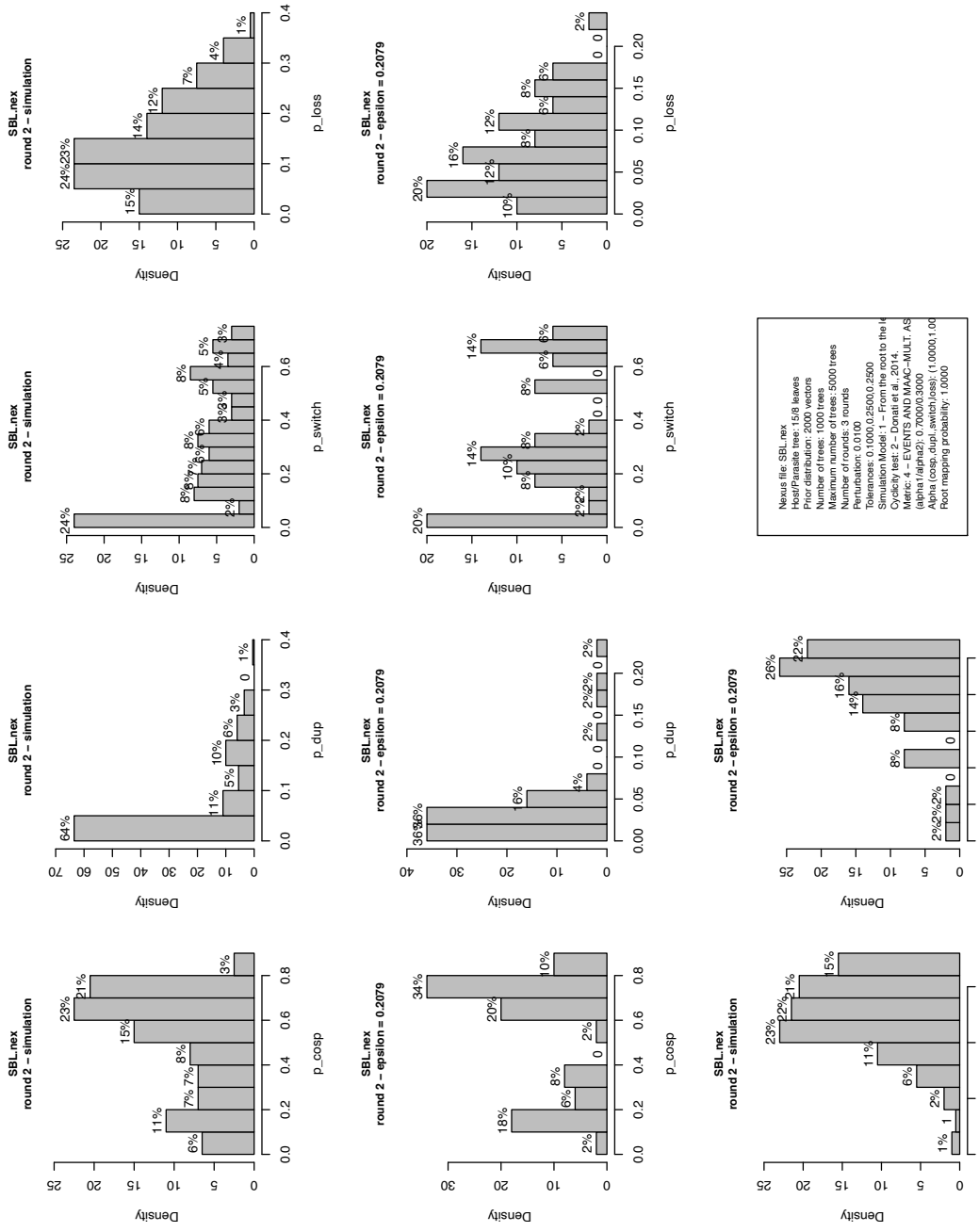


Figure O: SBL dataset. First row: histograms of the input parameters. Second row: histograms of the parameters after round 2. Third row: summary discrepancies of the input parameters and of the parameters after round 2.

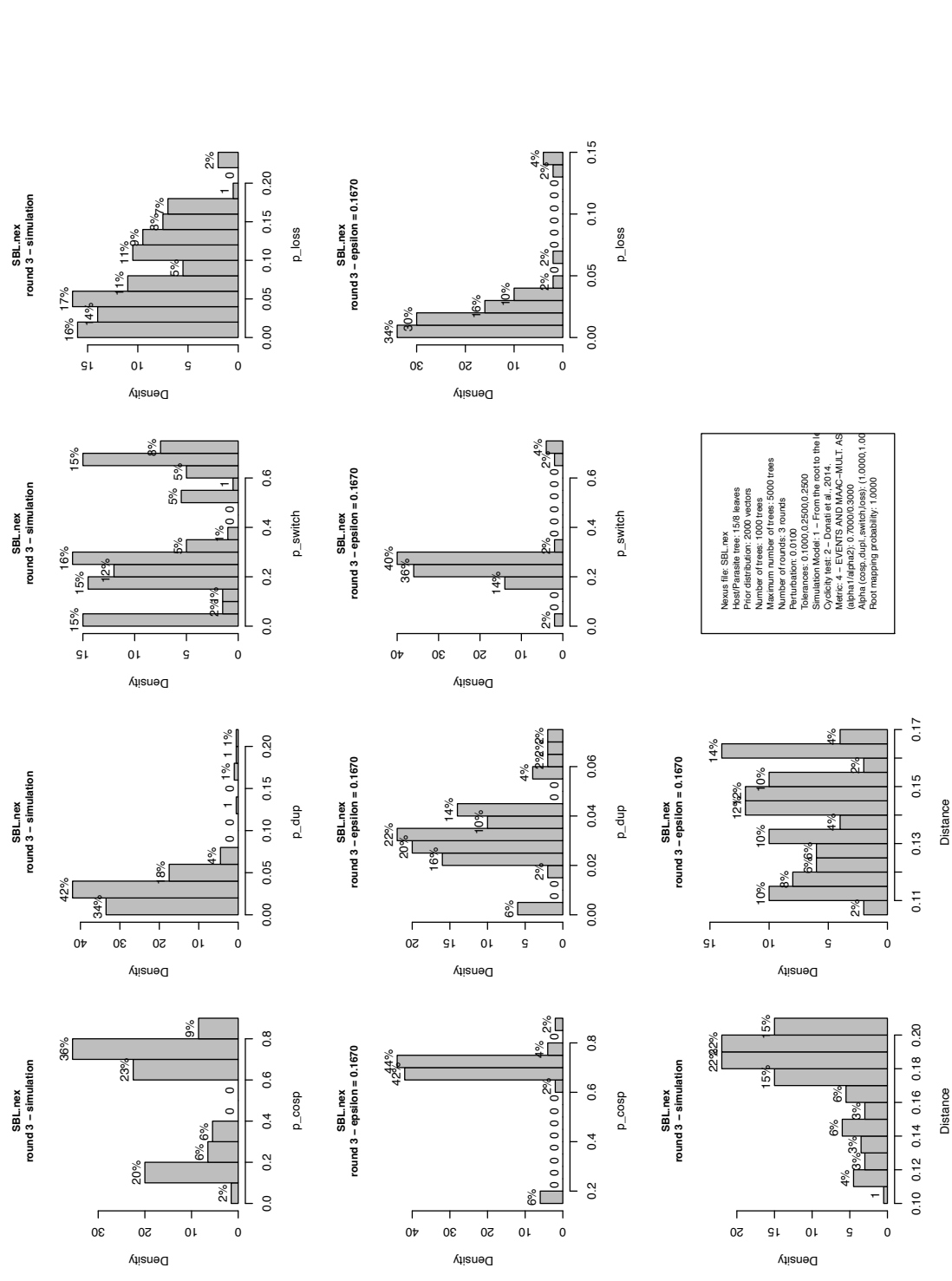


Figure P: SBL dataset. First row: histograms of the input parameters. Second row: histograms of the parameters after round 3. Third row: summary discrepancies of the input parameters and of the parameters after round 3.

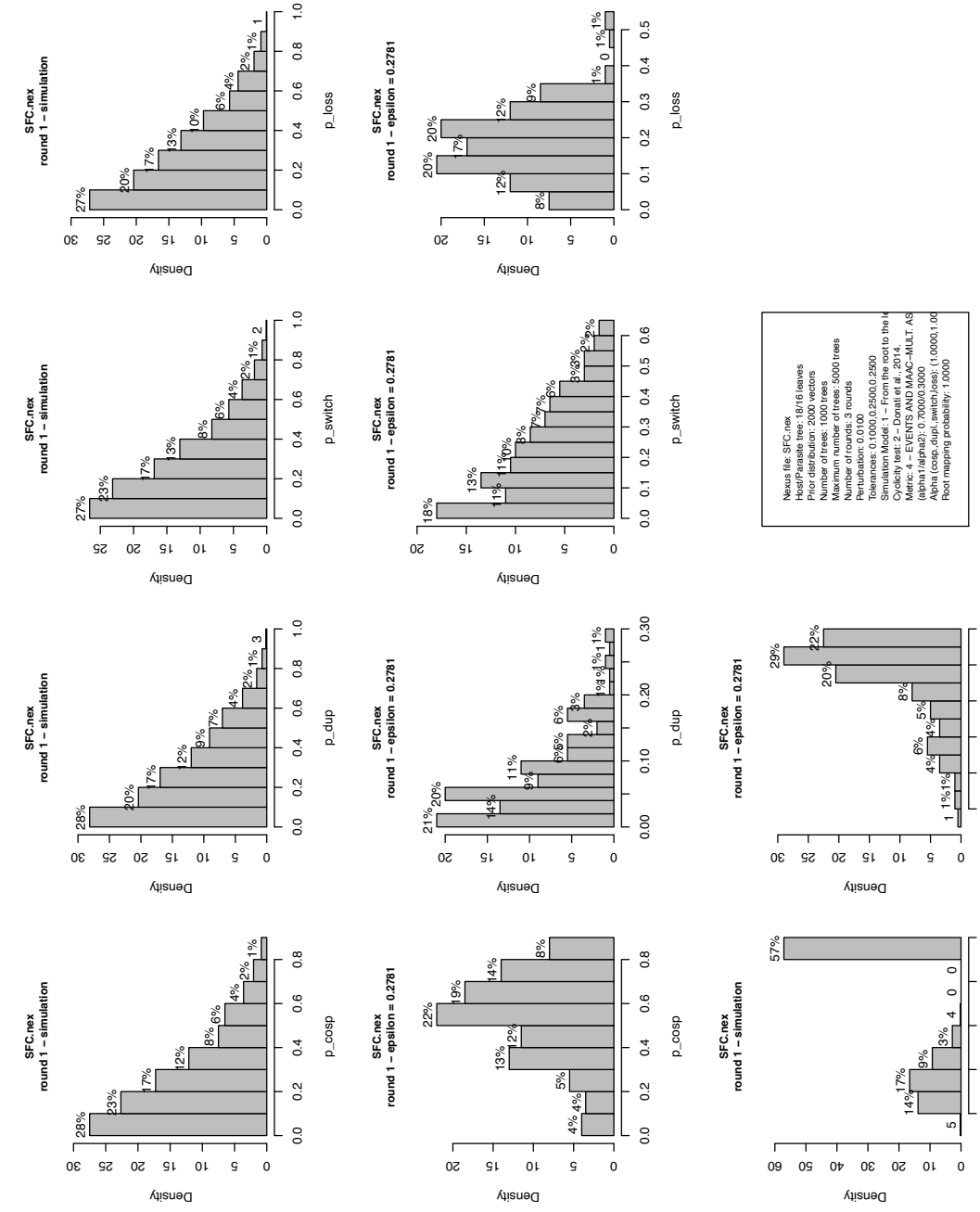


Figure Q: SFC dataset. First row: histograms of the input parameters. Second row: histograms of the parameters after round 1. Third row: summary discrepancies of the input parameters and of the parameters after round 1.

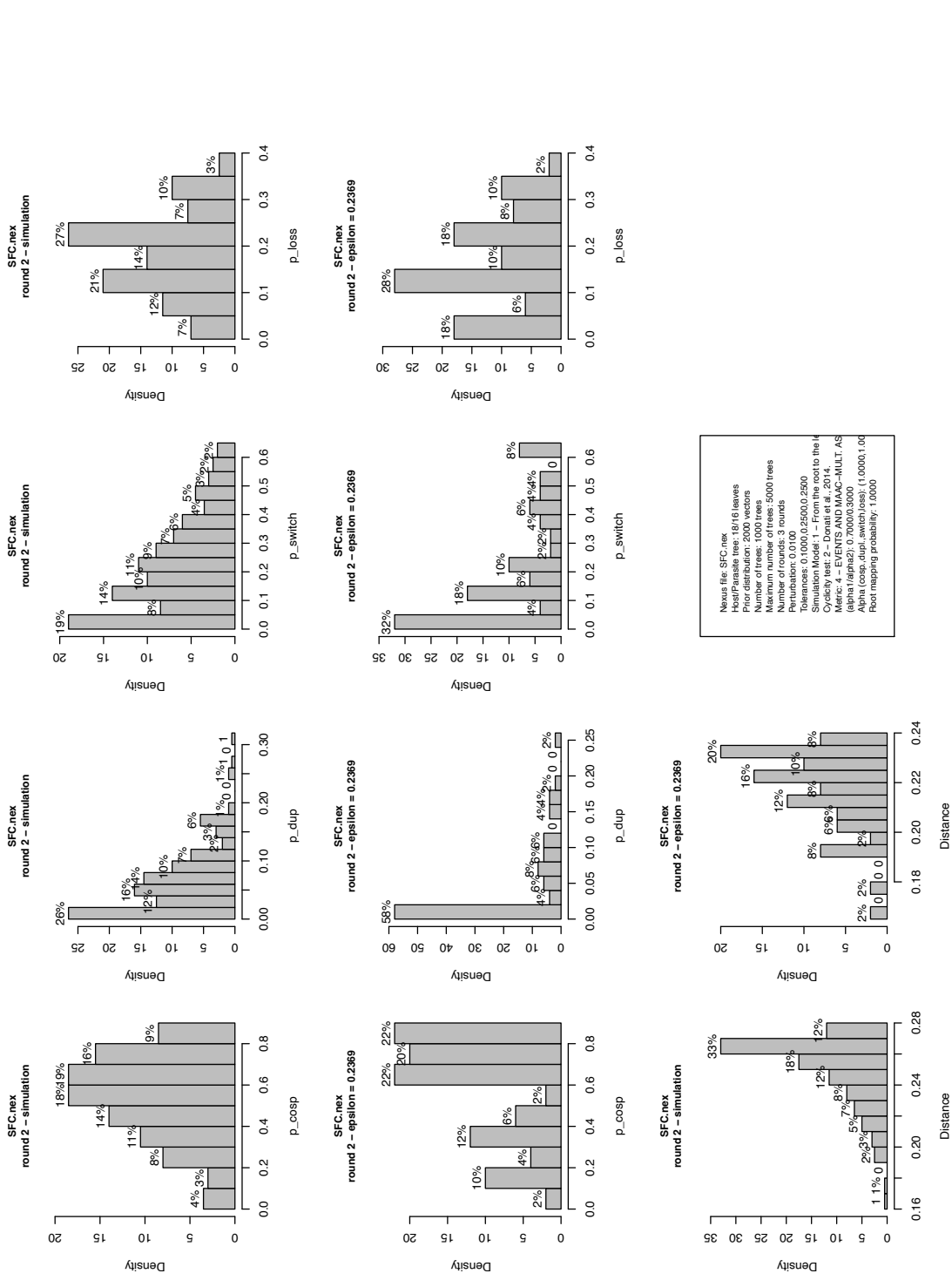


Figure R: SFC dataset. First row: histograms of the input parameters. Second row: histograms of the parameters after round 2. Third row: summary discrepancies of the input parameters and of the parameters after round 2.

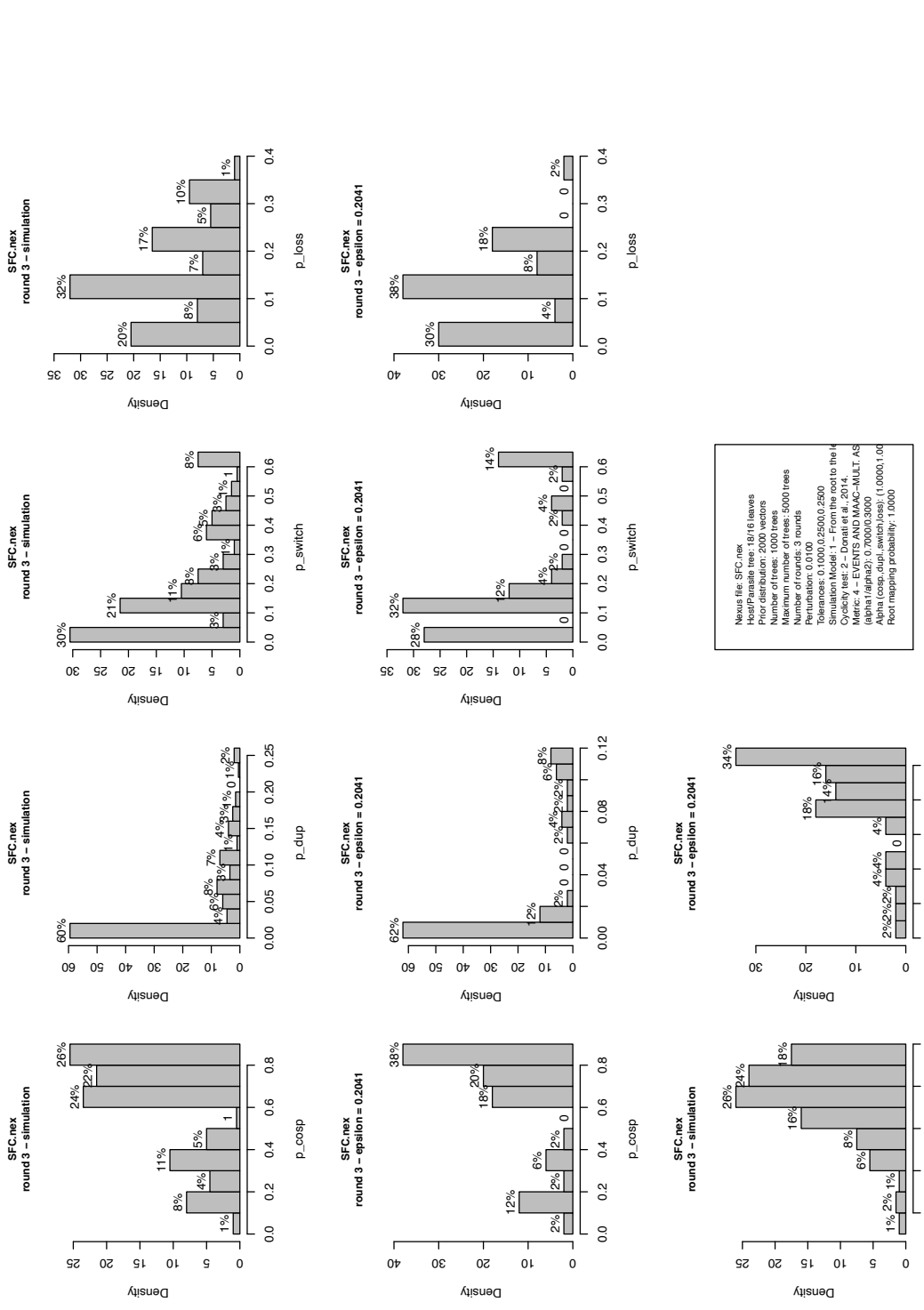


Figure S: SFC dataset. First row: histograms of the input parameters. Second row: histograms of the parameters after round 3. Third row: summary discrepancies of the input parameters and of the parameters after round 3.

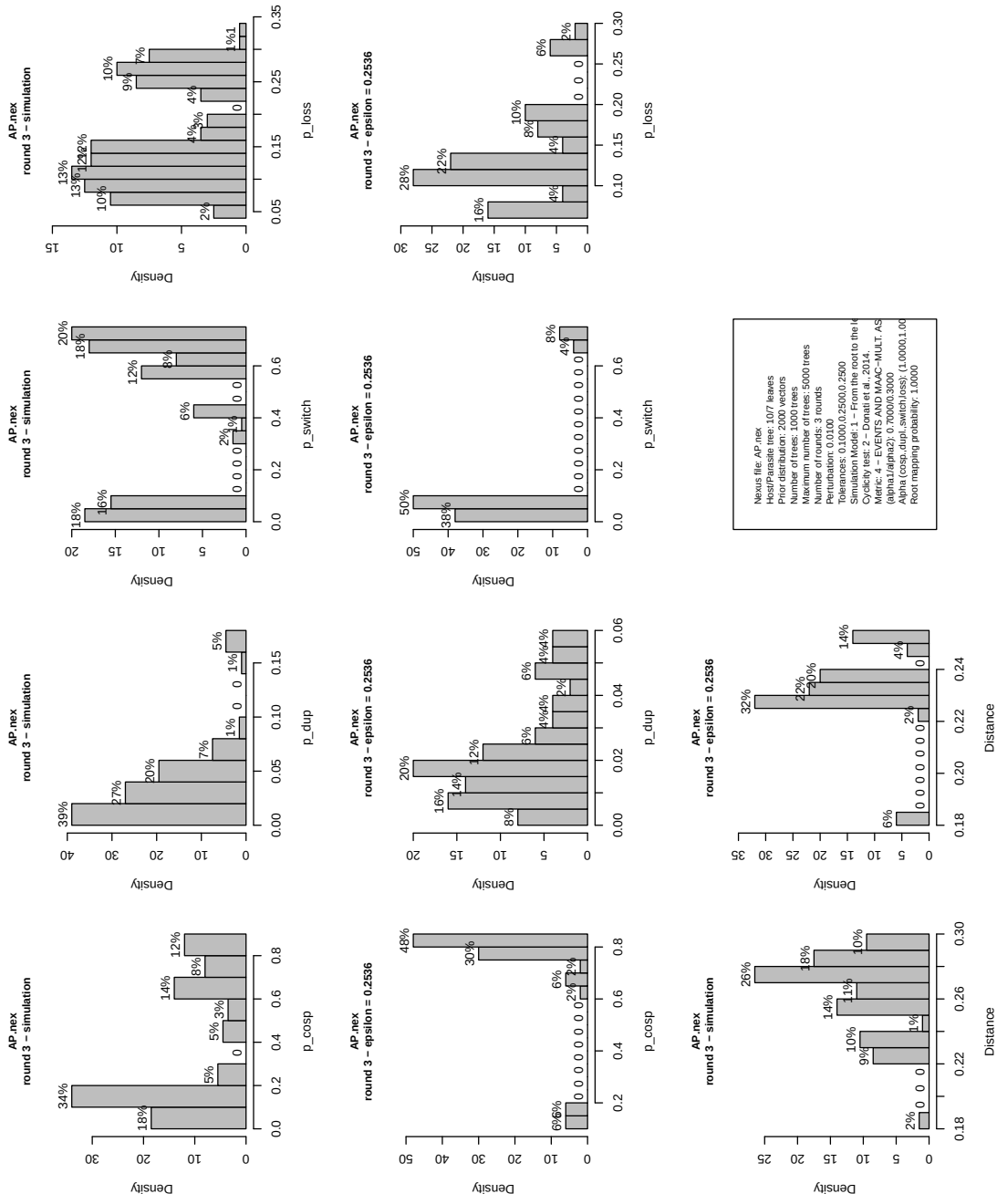


Figure T: AP dataset with perturbated spread probabilities. First row: histograms of the input parameters. Second row: histograms of the parameters after round 1. Third row: summary discrepancies of the input parameters and of the parameters after round 1.

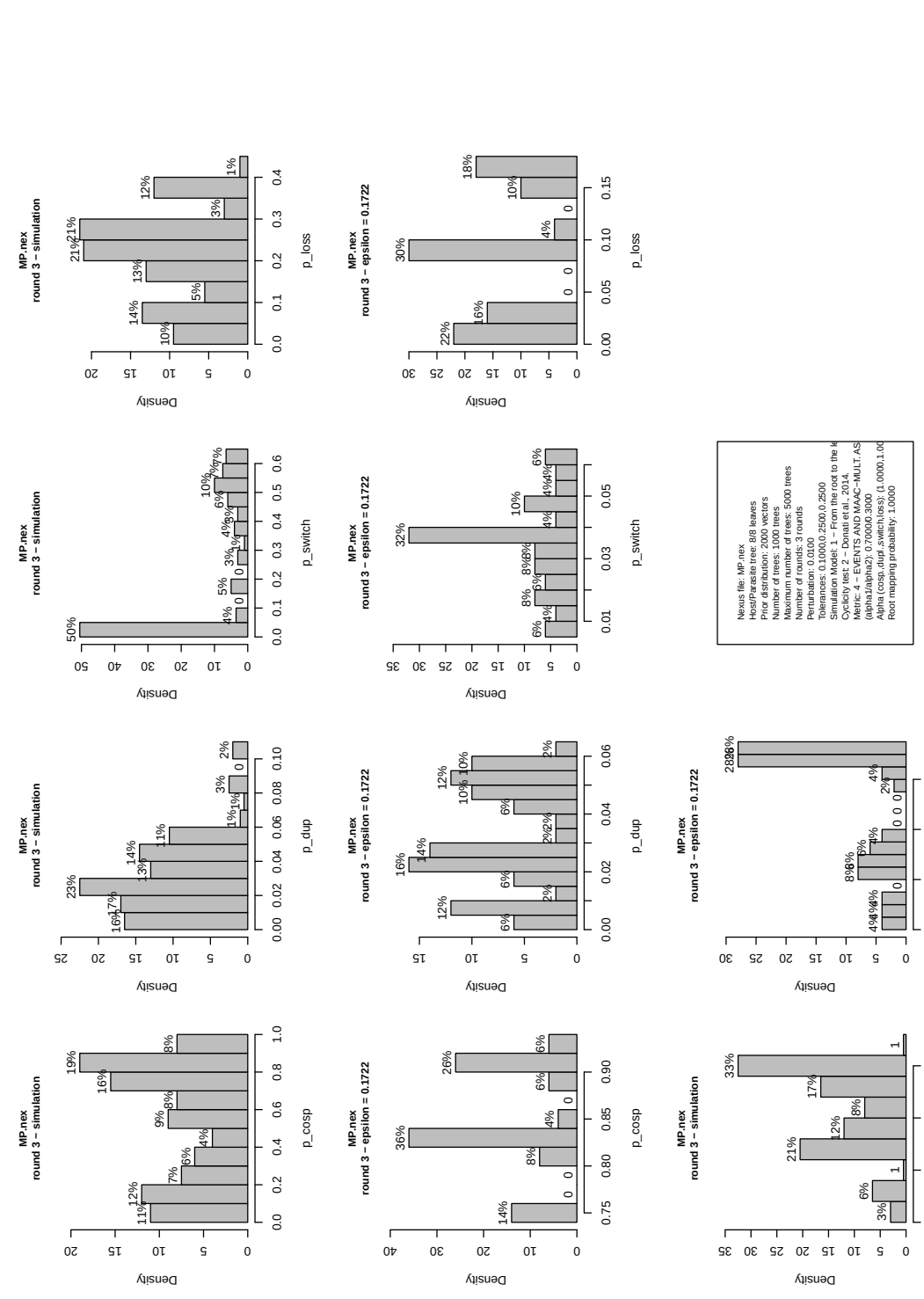


Figure U: MP dataset with perturbated spread probabilities. First row: histograms of the input parameters. Second row: histograms of the parameters after round 1. Third row: summary discrepancies of the input parameters and of the parameters after round 1.

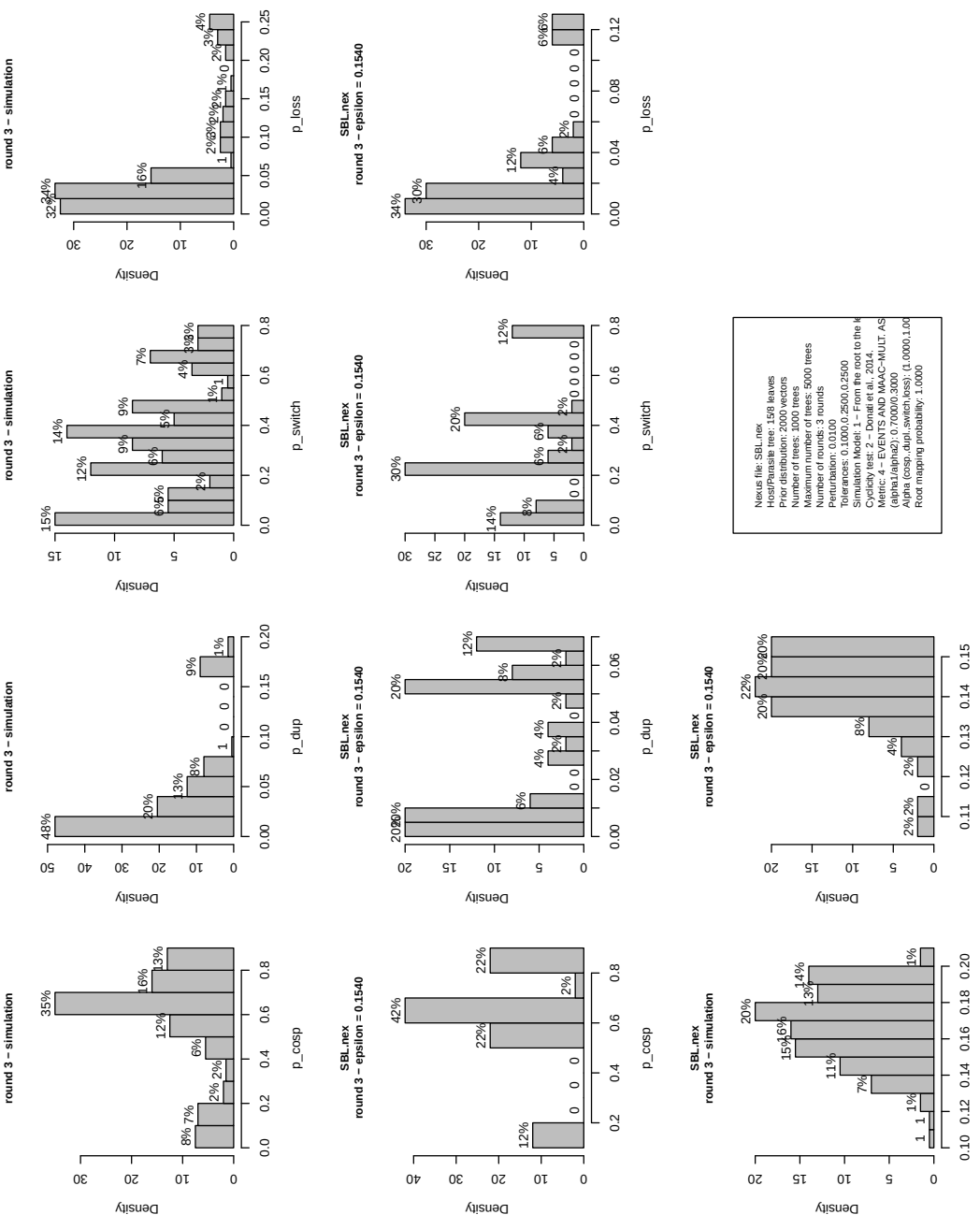


Figure V: SBL dataset with perturbated spread probabilities. First row: histograms of the input parameters. Second row: histograms of the parameters after round 1. Third row: summary discrepancies of the input parameters and of the parameters after round 1.

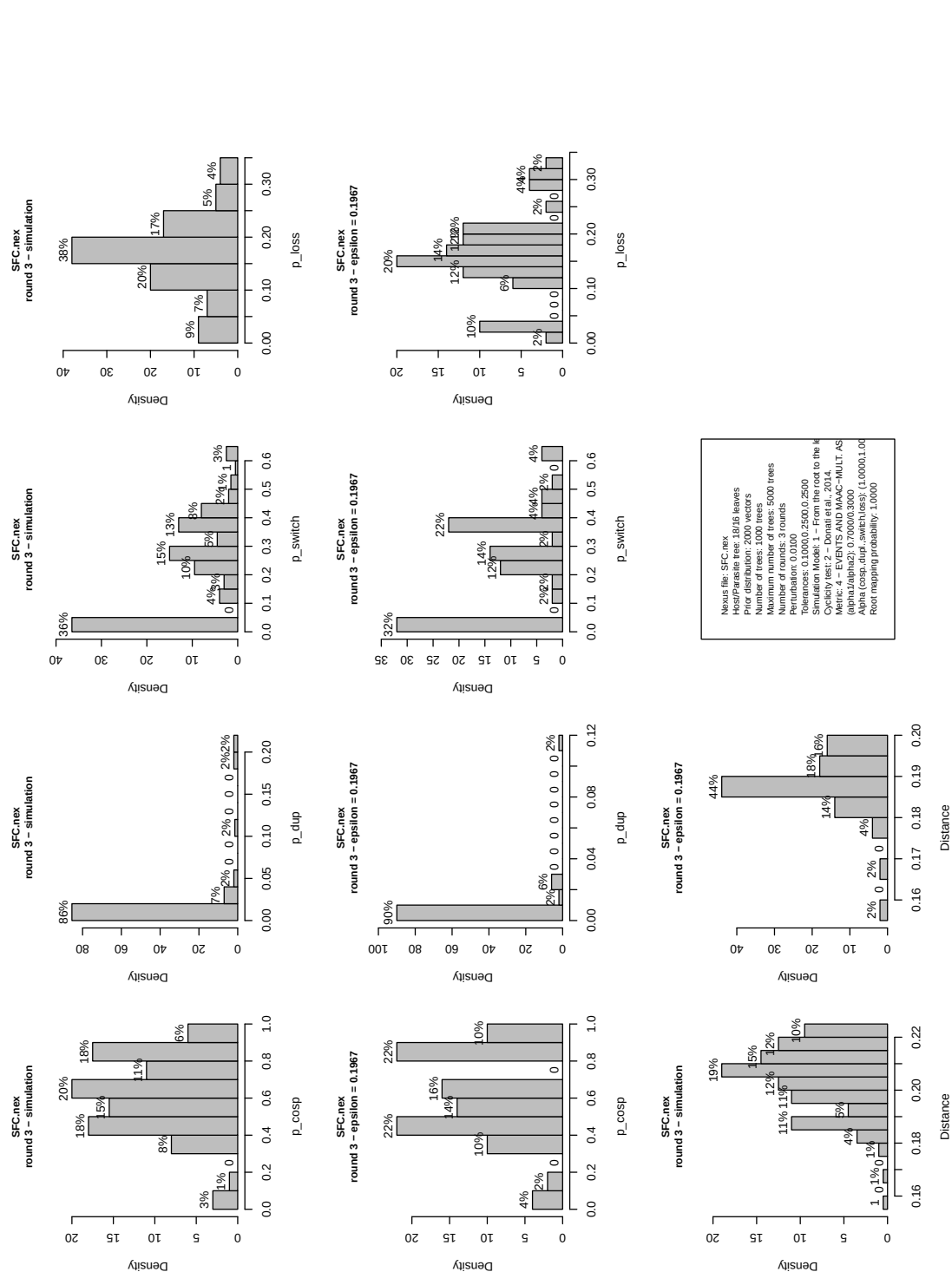


Figure W: SFC dataset with perturbated spread probabilities. First row: histograms of the input parameters. Second row: histograms of the parameters after round 1. Third row: summary discrepancies of the input parameters and of the parameters after round 1.

Table D.2: Representative vectors of the clusters produced by AmoCoala with perturbations for the SFC dataset. The column *#vectors* indicates the number of vectors in the cluster.

Dataset	*Cluster*	p_c	p_d	p_s	p_l	*#vectors*
SFC	1	0.4985	0.0024	0.3162	0.1829	31
	2	0.8738	0.0147	0.0180	0.0935	16
	3	0.1087	0.0012	0.5770	0.3131	3