Supplementary Material for: **Alignment of RNA Secondary Structures with Arbitrary Pseudoknots using Structural Sequences**

Luca Tesei, Francesca Levi, Michela Quadrini, Emanuela Merelli

Introduction

This Supplementary Material is organized into a set of Supplementary Notes (S1–S3), each addressing specific methodological or experimental aspects that complement the main text. It is intended to complement the mathematical definitions and experimental evaluations reported in the main article.

Supplementary Note S1: Step-by-step Dynamic Programming and Traceback Example

Definition of the recurrence

Let $x = x_1 \dots x_n$ and $y = y_1 \dots y_m$ be structural sequences with prefix constraints $x_i \in C_i = \{1, \dots, 2i-1\}$ and $y_j \in C_j = \{1, \dots, 2j-1\}$. The dynamic programming (DP) matrix $W \in \mathbb{N}^{(n+1)\times(m+1)}$ is defined by:

$$W_{0,0} = 0, \qquad W_{i,0} = i \ (i \geq 1), \qquad W_{0,j} = j \ (j \geq 1),$$
$$W_{i,j} = \min \begin{cases} W_{i-1,j} + \gamma_{\text{del}}(x_i \mid j) & \text{(delete } x_i\text{)}, \\ W_{i,j-1} + \gamma_{\text{ins}}(y_j \mid i) & \text{(insert } y_j\text{)}, \\ W_{i-1,j-1} + \gamma_{\text{sub}}(x_i, y_j \mid i, j) & \text{(match/mismatch)}. \end{cases}$$

Here $\gamma_{\text{del}}, \gamma_{\text{ins}}, \gamma_{\text{sub}}$ are the edit costs subject to alignment correctness (Definition 10 in the main text): an edit is admissible only if the resulting prefixes remain well-formed. If an edit would violate a prefix constraint, its cost is set to $+\infty$ (i.e., it is not eligible in the min). In the example below, unit gap costs are used for admissible insertions/deletions, matches cost 0, and admissible mismatches have unit cost.

DP Table for Example $(1, 1, 5, 3)$ vs $(1, 1, 5, 5, 2)$

Consider $x = (1, 1, 5, 3)$ and $y = (1, 1, 5, 5, 2)$ (structures S_1 and S_2 in the main text). The filled matrix reported in the paper is:

	j=0	y_1=1	y_2=1	y_3=5	y_4=5	y_5=2
i=0	0	1	2	3	4	5
x_1=1	1	0	1	4	5	6
x_2=1	2	1	0	5	6	6
x_3=5	3	4	5	0	1	2
x_4=3	4	5	5	1	1	2

The optimal alignment obtained by traceback is:

$$\begin{array}{ccccc} 1 & 1 & 5 & 3 & - \\ 1 & 1 & 5 & 5 & 2 \end{array}$$

which coincides with Eq. (1) in the main text.

Forward Filling of the DP Matrix

We illustrate here how the DP matrix W is filled using the recurrence of Definition (X). As usual, the first row and first column are initialized through pure insertion/deletion operations:

$$W_{0,j} = j, \qquad W_{i,0} = i.$$

Thus,

	$j = 0$	1	2	3	4	5
$i = 0$	0	1	2	3	4	5
1	1					
2	2					
3	3					
4	4					

We now show a few representative cases of the DP update process.

Cell $(1,1)$**:** x_1=1, y_1=1 A match is admissible since $1 \in C_1$ and $1 \in C_1$:

$$W_{1,1} = \min\{W_{0,0}+0,\ W_{1,0}+1,\ W_{0,1}+1\} = 0.$$

Cell $(1,3)$**:** x_1=1, y_3=5 Substitution is valid, but incurs a mismatch penalty:

$$W_{1,3} = \min\{W_{0,2}+3,\ W_{1,2}+1,\ W_{0,3}+1\} = \min\{2+3,\ 2+1,\ 3+1\} = 4.$$

Cell $(2,3)$**:** x_2=1, y_3=5 Deletion or insertion are possible, as well as substitution (admissible since $1 \in C_2 = \{1,2,3\}$ and $5 \in C_3 = \{1,\ldots,5\}$):

$$W_{2,3} = \min\{W_{1,2}+\gamma_{\text{sub}}(1,5),\ W_{2,2}+1,\ W_{1,3}+1\} = \min\{0+5,\ 0+1,\ 4+1\} = 1.$$

Cell $(4,5)$ Continuing in this manner for every (i,j) in row-major order yields:

	$j=0$	y_1=1	y_2=1	y_3=5	y_4=5	y_5=2
$i=0$	0	1	2	3	4	5
x_1=1	1	0	1	4	5	6
x_2=1	2	1	0	5	6	6
x_3=5	3	4	5	0	1	2
x_4=3	4	5	5	1	1	2

The value in $(4,5)$ provides the optimal score $W_{4,5}=2$. The next section details the traceback procedure extracting the corresponding alignment.

Optimal traceback: step-by-step explanation

We reconstruct the solution *step by step* from $(i,j)=(4,5)$ to $(0,0)$. When multiple predecessors yield the same optimal value, we apply the tie-breaking order *diag* > *left* > *up*. The rule applied at each step is stated explicitly.

1. $(i,j)=(4,5)$: $W_{4,5}=2$. Eligible predecessors:

   - left $(4,4)$ with $W_{4,4}$=1 and admissible insertion of y_5=2: $1+\gamma_{\text{ins}}(2 \mid i{=}4)=2$;
   - diag $(3,4)$ with $W_{3,4}$=1 and admissible substitution (x_4=3, y_5=2): $1+\gamma_{\text{sub}}(3,2 \mid 4,5)=2$;
   - up $(3,5)$ with $W_{3,5}$=2 and admissible deletion of x_4=3: $2+1=3$ (worse).

   We select **left** (insert y_5): emit pair $(-,2)$ and move to $(4,4)$.

2. $(4,4)$: $W_{4,4}=1$. Eligible predecessors:

   - diag $(3,3)$ with $W_{3,3}$=0 and admissible substitution (x_4=3, y_4=5): $0+\gamma_{\text{sub}}(3,5 \mid 4,4)=1$;
   - left $(4,3)$ has $W_{4,3}$=1 (insertion) $\Rightarrow 2$;
   - up $(3,4)$ has $W_{3,4}$=1 (deletion) $\Rightarrow 2$.

   We select **diag** (mismatch): emit $(3,5)$ and move to $(3,3)$.

3. $(3,3)$: $W_{3,3}=0$. Only **diag** is optimal with match (x_3=5, y_3=5): $W_{2,2}$=$0+\gamma_{\text{sub}}(5,5)=0$. Emit $(5,5)$ and move to $(2,2)$.

4. $(2,2)$: $W_{2,2}=0$. **Diag** with match $(1,1)$: $W_{1,1}$=$0+0=0$. Emit $(1,1)$ and move to $(1,1)$.

5. $(1,1)$: $W_{1,1}=0$. **Diag** with match $(1,1)$: $W_{0,0}$=$0+0=0$. Emit $(1,1)$ and move to $(0,0)$.

Reading the emitted pairs in reverse order yields the optimal alignment

$$\begin{matrix} 1 & 1 & 5 & 3 & - \\ 1 & 1 & 5 & 5 & 2 \end{matrix}.$$

also written $(1,1),(1,1),(5,5),(3,5),(-,2)$.

Provenance Map (DP Rule Selection)

For completeness, the figure below annotates each internal cell with the chosen predecessor (↖ diagonal; ← left; ↑ up) used by our implementation *for this instance* after correctness filtering and tie-breaking. Cells not used in the optimal path are still annotated with the rule that minimized their value during the fill.

	0	1	1	5	5	2
0	·	←	←	←	←	←
1	↑	↖	↖	←	←	←
1	↑	↑	↖	←	←	←
5	↑	↑	↑	↖	↖	←
3	↑	↑	↑	↖	↖	←

(Top row and left column are base cases.)

Remarks. (i) If an edit violates alignment correctness (prefix constraint), the corresponding move is discarded by setting its cost to $+\infty$. (ii) When multiple admissible moves achieve the same value, the tie-breaking order (diag > left > up) ensures deterministic traceback, but the final score is independent of tie-breaking. (iii) The same procedure applies to the unconstrained variant (without correctness checks), in which case no move is filtered out.

Supplementary Note S2: Proofs of Theoretical Results

Proof of Theorem 2

For the sake of readability, we recall here Theorem 2 of the main text.

Theorem. *Let $x = x_1x_2\dots x_n$ and $y = y_1y_2\dots y_m$ be two structural sequences. Let $\hat{x}_i$ be the prefix $x_1, x_2, \dots, x_i$ and let $\hat{y}_j$ be the prefix $y_1, y_2, \dots, y_j$. For all i, j such that $0 \leq i \leq n$ and $0 \leq j \leq m$, let us define the following function*

$$W_{i,j} \triangleq W(\hat{x}_i, \hat{y}_j)$$

where by $\hat{x}_0$ and by $\hat{y}_0$ we denote the empty sequences.

Then,

- $W_{0,0} \triangleq 0$
- $W_{0,j} \triangleq \sum_{k=1}^{j} w(-, y_k)$, *for all* $j = 1, \dots, m$
- $W_{i,0} \triangleq \sum_{k=1}^{i} w(x_k, -)$, *for all* $i = 1, \dots, n$

For all $i = 1, \dots, n$ and $j = 1, \dots, m$

$$W_{i,j} \triangleq \min \begin{cases} W_{i-1,j} + w(x_i, -) & \textit{if} \quad x_i \in C_j & \textit{deletion} \\ W_{i-1,j-1} + w(x_i, y_j) & \textit{if} \quad x_i \in C_j \wedge y_j \in C_i & \textit{match/mismatch} \\ W_{i,j-1} + w(-, y_j) & \textit{if} \quad y_j \in C_i & \textit{insertion} \end{cases}$$

Proof. Let $0 \leq i \leq n$ and $0 \leq j \leq m$ be two indexes. If $i = 0$ then $W(\hat{x}_i, \hat{y}_j)$ denotes the optimal correct alignment of the empty sequence with $\hat{y}_j$. In this case such alignment is

$$\begin{array}{lcllll} \hat{x}_0 & = & - & - & \dots & - \\ \hat{y}_j & = & y_1 & y_2 & \dots & y_j \end{array}$$

with cost $W(\hat{x}_0, \hat{y}_j) = \sum_{k=1}^{j} w(-, y_k)$ by Definition of SERNA distance in which $L = j$, also equal to $W_{0,j}$. The case $j = 0$ is analogous and we conclude, using the same definitions, that $W_{i,0} = W(\hat{x}_i, \hat{y}_0)$.

We now follow the lines of the proof for standard edit distance given in Waterman (1995).

If $i > 0$ and $j > 0$, by Definition of SERNA distance, we let v denote the minimum cost over all correct alignments of $\hat{x}_i$ with $\hat{y}_j$. Let

$$A \ : \ \begin{array}{lcllll} \hat{x}_i^* & = & x_1^* & x_2^* & \dots & x_L^* \\ \hat{y}_j^* & = & y_1^* & y_2^* & \dots & y_L^* \end{array}$$

be one of these alignments with minimum cost, i.e., $\sum_{k=1}^{L} w(x_k^*, y_k^*) = v$. Therefore, we have that $W(\hat{x}_i, \hat{y}_j) = v = \sum_{k=1}^{L} w(x_k^*, y_k^*)$.

We can write $W(\hat{x}_i, \hat{y}_j) = (\sum_{k=1}^{L-1} w(x_k^*, y_k^*) + w(x_L^*, y_L^*))$.

There are three cases, depending on the form of the edit operation at index L:

(Case 1)

$$\begin{array}{lcl} \hat{x}_L^* & = & x_i \\ \hat{y}_L^* & = & - \end{array}$$

In this case x_i is deleted. By the fact that the chosen alignment is correct, we also know that $x_i \in C_j$. We can say that

$$A' \ : \ \begin{array}{llll} x_1^* & x_2^* & \dots & x_{L-1}^* \\ y_1^* & y_2^* & \dots & y_{L-1}^* \end{array}$$

is a an alignment with minimum cost of $\hat{x}_{i-1}$ and $\hat{y}_j$ with cost $\sum_{k=1}^{L-1} w(x_k^*, y_k^*)$. To see this, suppose by contradiction, that there exists another alignment A'' of $\hat{x}_{i-1}$ and $\hat{y}_j$ with a cost lower than that of A'. Then, the alignment A'' plus the last element

$$\begin{array}{lcl} \hat{x}_L^* & = & x_i \\ \hat{y}_L^* & = & - \end{array}$$

would be an alignment of $\hat{x}_i$ and $\hat{y}_j$ with a cost lower than that of A, which, by hypothesis, was a minimum cost.

Thus, in this case, we have that $x_i \in C_j$ and

$$\begin{array}{lcl} W_{i,j} & \triangleq & W(\hat{x}_i, \hat{y}_j) = \sum_{k=1}^{L-1} w(x_k^*, y_k^*) + w(x_i, -) = W(\hat{x}_{i-1}, \hat{y}_j) + w(x_i, -) \\ & \triangleq & W_{i-1,j} + w(x_i, -) \end{array} \tag{1}$$

(Case 2)

$$\begin{array}{lcl} \hat{x}_L^* & = & x_i \\ \hat{y}_L^* & = & y_j \end{array}$$

In this case there is a match/mismatch between x_i and y_j. Again, by the fact that the chosen alignment is correct, we also know that $x_i \in C_j$ and $y_j \in C_i$. We can say that

$$A' \ : \ \begin{array}{cccc} x_1^* & x_2^* & \dots & x_{L-1}^* \\ y_1^* & y_2^* & \dots & y_{L-1}^* \end{array}$$

is an alignment with a minimum cost of $\hat{x}_{i-1}$ and $\hat{y}_{j-1}$ with cost $\sum_{k=1}^{L-1} w(x_k^*, y_k^*)$. To see this, suppose by contradiction, that there exists another alignment A'' of $\hat{x}_{i-1}$ and $\hat{y}_{j-1}$ with a cost lower than that of A'. Then, the alignment A'' plus the last element

$$\begin{array}{rcl} \hat{x}_L^* & = & x_i \\ \hat{y}_L^* & = & y_j \end{array}$$

would be an alignment of $\hat{x}_i$ and $\hat{y}_j$ with a cost lower than that of A, which, by hypothesis, was a minimum cost.

Thus, in this case, we have that $x_i \in C_j$, $y_j \in C_i$ and

$$\begin{array}{rcl} W_{i,j} & \triangleq & W(\hat{x}_i, \hat{y}_j) = \sum_{k=1}^{L-1} w(x_k^*, y_k^*) + w(x_i, y_j) = W(\hat{x}_{i-1}, \hat{y}_{j-1}) + w(x_i, y_j) \\ & \triangleq & W_{i-1,j-1} + w(x_i, y_j) \end{array} \tag{2}$$

(Case 3)

$$\begin{array}{rcl} \hat{x}_L^* & = & - \\ \hat{y}_L^* & = & y_j \end{array}$$

In this case there is an insertion of y_j. Again, by the fact that the chosen alignment is correct, we also know that $y_j \in C_i$. Following the same reasoning of the other two cases, we conclude that $y_j \in C_i$ and

$$\begin{array}{rcl} W_{i,j} & \triangleq & W(\hat{x}_i, \hat{y}_j) = \sum_{k=1}^{L-1} w(x_k^*, y_k^*) + w(-, y_j) = W(\hat{x}_i, \hat{y}_{j-1}) + w(-, y_j) \\ & \triangleq & W_{i,j-1} + w(-, y_j) \end{array} \tag{3}$$

To conclude we observe that Equations (1), (2) and (3), corresponding to cases 1, 2 and 3, hold whenever the considered initial alignment A is optimal. Moreover, once the initial optimal alignment A is fixed, we have that, at each step, only one of the three cases is used.

If, on the contrary, we want to *construct* an optimal alignment from scratch, then we have to reason without the assumption that we already know one optimal alignment. Thus, suppose we want to determine the value of $W_{i,j}$ with $i > 0$ and $j > 0$. We can apply Equation (1) if $x_i \in C_j$, Equation (2) if $x_i \in C_j$ and $y_j \in C_i$, or Equation (3) if $y_j \in C_i$. Proposition 2 of the main text ensures that at least one of these cases is always true. If more than one constraint is satisfied, then we have to select the equation that will lead to the optimal alignment between $\hat{x}_i$ and $\hat{y}_j$. This will be the case corresponding to the minimum value among:

- the value of an optimal alignment of $\hat{x}_{i-1}$ and $\hat{y}_j$ plus the cost of cancelling x_i,
- the value of an optimal alignment of $\hat{x}_{i-1}$ and $\hat{y}_{j-1}$ plus the cost of matching/mismatching x_i and y_j,
- the value of an optimal alignment of $\hat{x}_i$ and $\hat{y}_{j-1}$ plus the cost of adding y_j.

Using the function W of Definition of SERNA distance, this value corresponds exactly to the minimum of the set

$$\{W(\hat{x}_{i-1}, \hat{y}_j) + w(x_i, -), W(\hat{x}_{i-1}, \hat{y}_{j-1}) + w(x_i, y_j), W(\hat{x}_i, \hat{y}_{j-1}) + w(-, y_j)\}$$

By Equation (3) in Theorem 2 of the main text this is equal to

$$\min\{W_{i-1,j} + w(x_i, -), W_{i-1,j-1} + w(x_i, y_j), W_{i,j-1} + w(-, y_j)\}$$

Thus, we obtain exactly the procedure expressed by the recurrence of Equation (4) in Theorem 2 of the main text.

□

Proof of Theorem 3

For the sake of readability, we recall Theorem 3 of the main text.

Theorem. *Let x and y be two structural sequences and let $W(x, y)$ be their SERNA distance. Then:*

1. *$W(x, y) \geq 0$ and $W(x, y) = 0$ iff $x = y$ (**non-negativity, identity of indiscernibles**);*
2. *$W(x, y) = W(y, x)$ (**symmetry**);*
3. *for any structural sequence z, $W(x, y) \leq W(x, z) + W(z, y)$ (**triangle inequality**).*

Proof. Let x and y be two structural sequences. By Definition of SERNA distance, $W(x, y) = \sum_{i=1}^{L} w(x_i^*, y_i^*)$ where the sum corresponds to a *correct* alignment A of x and y with minimum cost:

$$A \quad : \quad \begin{array}{ccllll} \hat{x}^* & = & x_1^* & x_2^* & \dots & x_L^* \\ \hat{y}^* & = & y_1^* & y_2^* & \dots & y_L^* \end{array}$$

Moreover we know that it holds $chk(n, m, L)$.

Recall that w is assumed to be a metric. Non-negativity holds trivially because the cost of each edit operation $w(x_i^*, y_i^*)$ is greater than or equal to 0, then the sum is greater than or equal to 0. Also the identity of indiscernibles holds trivially because the sum is equal to 0 only if all the addends (which are greater than or equal to 0) are equal to 0, i.e., for all i, $x_i^* = y_i^*$, $x_i^* \neq -$ and $y_i^* \neq -$. This means that x and y are the same structural sequence.

To see the symmetry, we observe that every edit operation can be inverted, provided that the constraints are satisfied. We can obtain a correct alignment with minimum cost of y and x by inverting A into A^{r}:

$$A^{\mathrm{r}} \quad : \quad \begin{array}{ccllll} \hat{z}^* & = & z_1^* & z_2^* & \dots & z_L^* \\ \hat{u}^* & = & u_1^* & u_2^* & \dots & u_L^* \end{array}$$

where $z_i^* = y_i^*$ and $u_i^* = x_i^*$ for all $i = 1, 2, \dots, L$. This is an alignment of y and x having the same cost of $W(x, y)$, which must be minimum because otherwise there would exist an alignment of x and y which is better than A.

It remains to show that A^{r} is correct. We proceed by induction on ℓ to prove that $chk(i, j, \ell) = chk(j, i, \ell)$ for all $i \in \{0, 1, \dots, n\}$, $j \in \{0, 1, \dots, m\}$ and $\ell \in \{0, 1, \dots L\}$. Therefore, $chk(n, m, L) = true = chk(m, n, L)$.

(base case) $\ell = 0$, we have $\mathit{chk}(i, j, 0) = \mathit{true} = \mathit{chk}(j, i, 0)$ by definition of *chk*.

(inductive case) $\ell > 0$, we assume that for all $i \in \{0, 1, \dots, n\}$, $j \in \{0, 1, \dots, m\}$ it holds $\mathit{chk}(i, j, \ell - 1) = \mathit{chk}(j, i, \ell - 1)$. We have four cases:

1. $\mathit{chk}(i, 0, \ell) = \mathit{true} = \mathit{chk}(0, i, \ell)$;

2. $(x^*_\ell, y^*_\ell) = (x_i, -) \wedge \ell > 0 \wedge i > 0$, i.e., x_i is cancelled from $\hat{x}$. By the correctness of A, we know that $x_i \in C_j$ and that $\mathit{chk}(i-1, j, \ell-1)$. On A^{r} this position becomes $(z^*_\ell, u^*_\ell,) = (-, u_i) \wedge \ell > 0 \wedge i > 0$ where $u_i = x_i$. We have to show that $u_i \in C_j$, but this is trivially true because we know that $u_i = x_i$ and that $x_i \in C_j$. Moreover, $\mathit{chk}(j, i-1, \ell-1) = \mathit{chk}(i-1, j, \ell-1)$, which holds by induction hypothesis. Thus, in this case, $\mathit{chk}(i, j, \ell) = \mathit{chk}(j, i, \ell)$.

3. $(x^*_\ell, y^*_\ell) = (x_i, y_j) \wedge \ell > 0 \wedge i > 0 \wedge j > 0$, i.e., x_i is matched/mismatched with y_j. By the correctness of A, we know that $x_i \in C_j$, $y_j \in C_i$ and that $\mathit{chk}(i-1, j-1, \ell-1)$. On A^{r} this position becomes $(z^*_\ell, u^*_\ell,) = (z_j, u_i) \wedge \ell > 0 \wedge i > 0 \wedge j > 0$ where $u_i = x_i$ and $z_j = y_j$. We have to show that $u_i \in C_j$ and that $z_j \in C_i$, but this is trivially true because we know that $u_i = x_i$, $z_j = y_j$, $x_i \in C_j$ and that $y_j \in C_i$. Moreover, $\mathit{chk}(j-1, i-1, \ell-1) = \mathit{chk}(i-1, j-1, \ell-1)$, which holds by induction hypothesis. Thus, also in this case, $\mathit{chk}(i, j, \ell) = \mathit{chk}(j, i, \ell)$.

4. $(x^*_\ell, y^*_\ell) = (-, y_j) \wedge \ell > 0 \wedge j > 0$, i.e., y_j is inserted into $\hat{y}$. By the correctness of A, we know that $y_j \in C_i$ and that $\mathit{chk}(i, j-1, \ell-1)$. On A^{r} this position becomes $(z^*_\ell, u^*_\ell,) = (z_j, -) \wedge \ell > 0 \wedge j > 0$ where $z_j = y_j$. We have to show that $z_j \in C_i$, but this is trivially true because we know that $z_j = y_j$ and that $y_j \in C_i$. Moreover, $\mathit{chk}(j-1, i, \ell-1) = \mathit{chk}(i, j-1, \ell-1)$, which holds by induction hypothesis. Thus, also in this case, $\mathit{chk}(i, j, \ell) = \mathit{chk}(j, i, \ell)$.

Finally, to prove the triangle inequality we proceed by contradiction. First recall that we assumed the distance w to be a metric, so triangle inequality holds among the costs of the edit operations. Then, assume that there is a structural sequence z such that $W(x, y) > W(x, z) + W(z, y)$, i.e., there are two optimal alignments A_0 and A_1 such that A_0 transforms x into z and A_1 transforms z into y. Therefore there is an alignment A_2 corresponding to the composition of A_0 with A_1 whose cost is less than $W(x, y)$. But such composition transforms x into y, so that would be an optimal alignment with cost less than $W(x, y)$, which is supposed to be minimum, a contradiction. Notice that the condition of w being a metric does not allow degenerate cases in which triangle inequality is violated by using costs of the edit operations with different magnitudes. □

Supplementary Note S3: Experimental Results

This Supplementary Note reports experimental results described in the main text using heatmaps. Each heatmap summarizes the clustering performance obtained using hierarchical clustering under single, complete, and average linkage, evaluated through the same metrics described in the main text.

The results are organized by ribosomal RNA type (5S, 16S, and 23S). For each RNA type, heatmaps are reported separately for Archaea, Bacteria, and Eukaryota, and within each group results are shown at three taxonomic ranks (Phylum, Class, and Order). This organization allows a comprehensive comparison across domains, taxonomic granularity levels, and RNA types, while preserving a consistent visualization layout throughout the Supplementary Material.

We also provide the numerical results as tables in the Zenodo repository . Tables report the values of the metrics for SERNA, SERNA-NC, ASPRA, PSMAlign, RAG-2D, Genus, and PskOrder. For the sake of completeness, they also include the metrics for base-pair number (BP), number of GC nucleotides in the sequence (GC-Seq), number of G-C base pairs (GC-Pairs), and number of nucleotides (Length).

5S Ribosomal RNA

Archaea

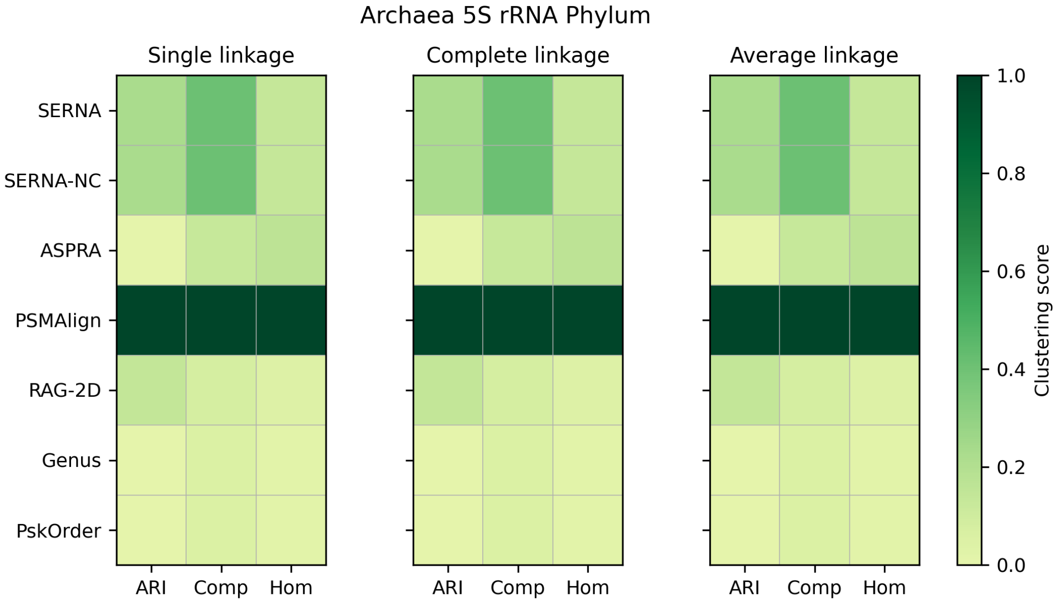

Figure 1: Hierarchical clustering results for Archaea 5S rRNA phylogenetic reconstruction at the Phylum level.

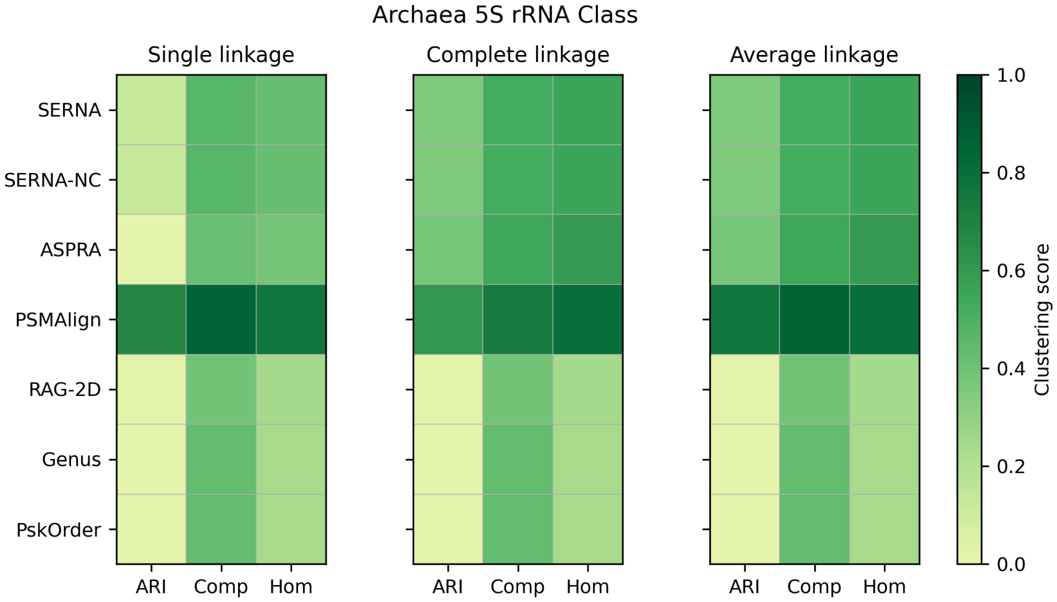

Figure 2: Hierarchical clustering results for Archaea 5S rRNA phylogenetic reconstruction at the Class level.

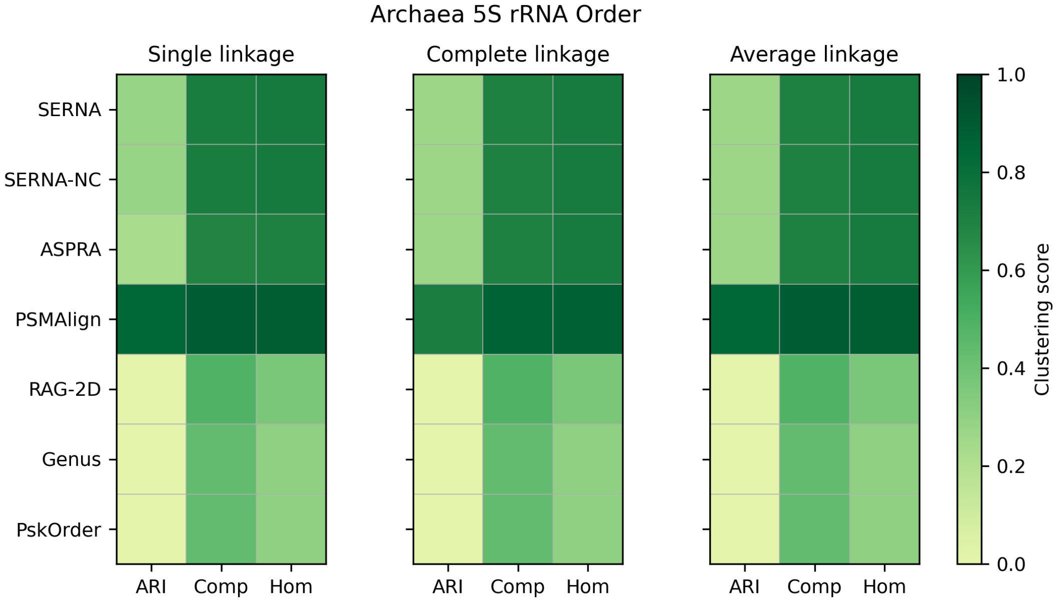

Figure 3: Hierarchical clustering results for Archaea 5S rRNA phylogenetic reconstruction at the Order level.

Bacteria

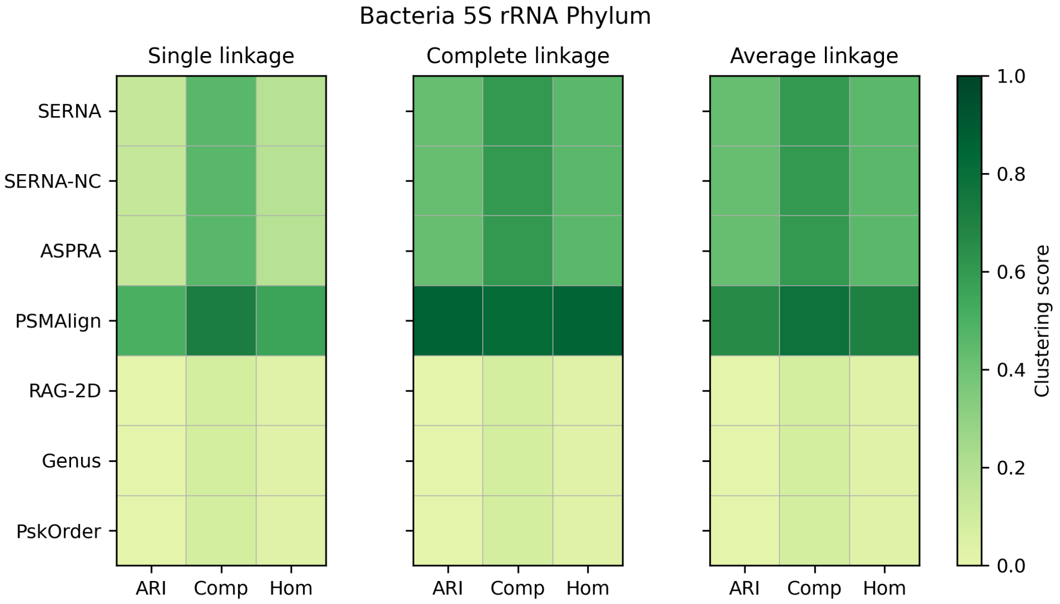

Figure 4: Hierarchical clustering results for Bacteria 5S rRNA phylogenetic reconstruction at the Phylum level.

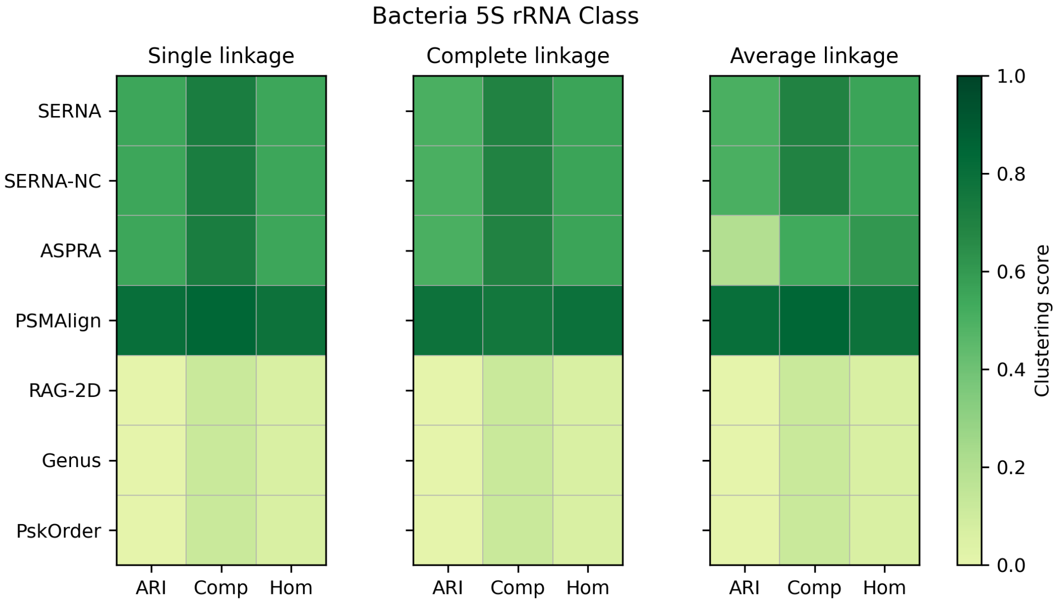

Figure 5: Hierarchical clustering results for Bacteria 5S rRNA phylogenetic reconstruction at the Class level.

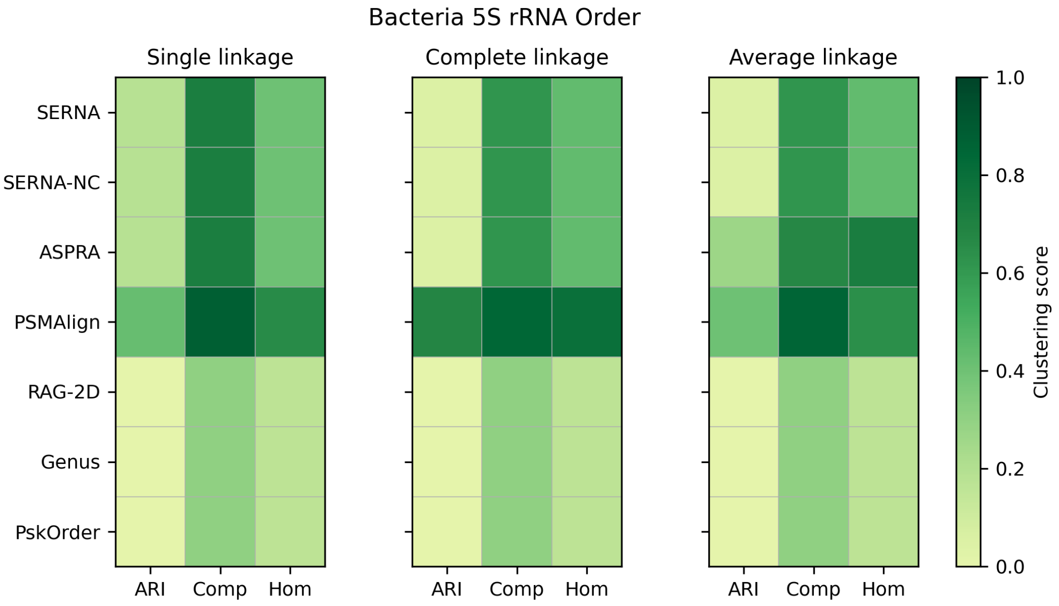

Figure 6: Hierarchical clustering results for Bacteria 5S rRNA phylogenetic reconstruction at the Order level.

Eukaryota

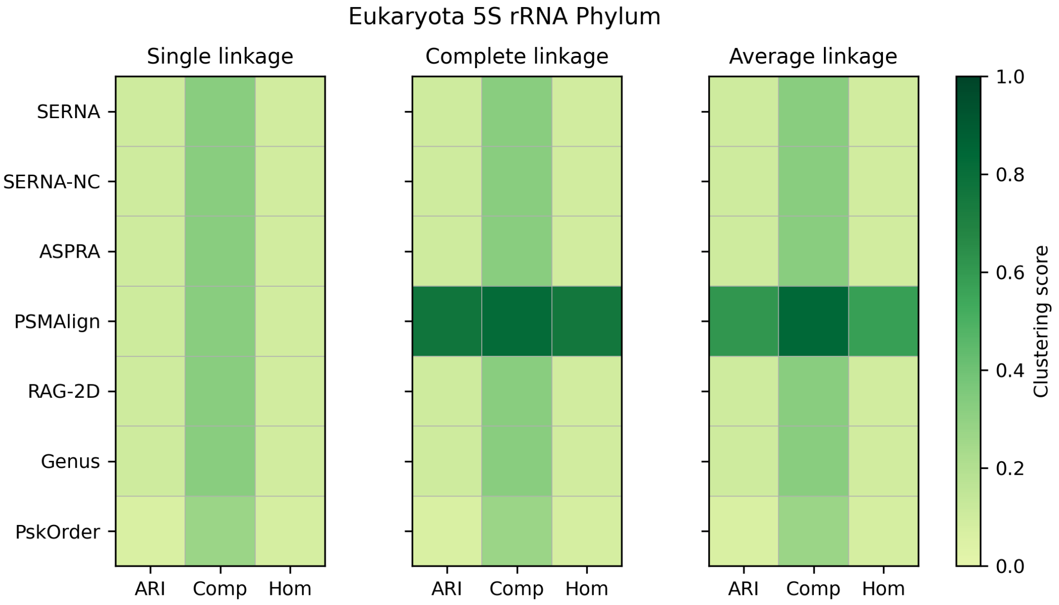

Figure 7: Hierarchical clustering results for Eukaryota 5S rRNA phylogenetic reconstruction at the Phylum level.

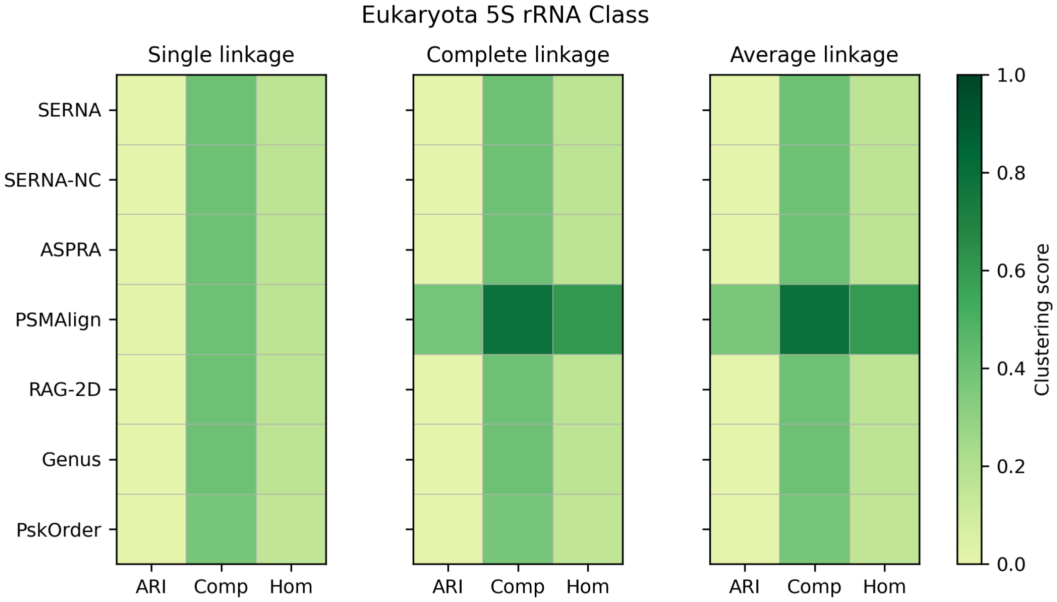

Figure 8: Hierarchical clustering results for Eukaryota 5S rRNA phylogenetic reconstruction at the Class level.

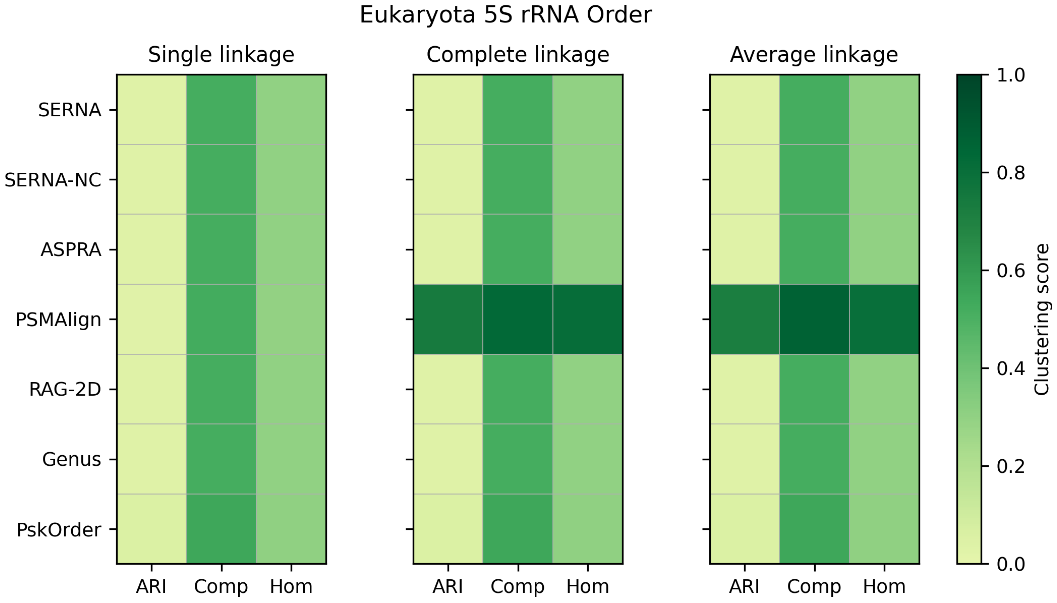

Figure 9: Hierarchical clustering results for Eukaryota 5S rRNA phylogenetic reconstruction at the Order level.

16S Ribosomal RNA

Archaea

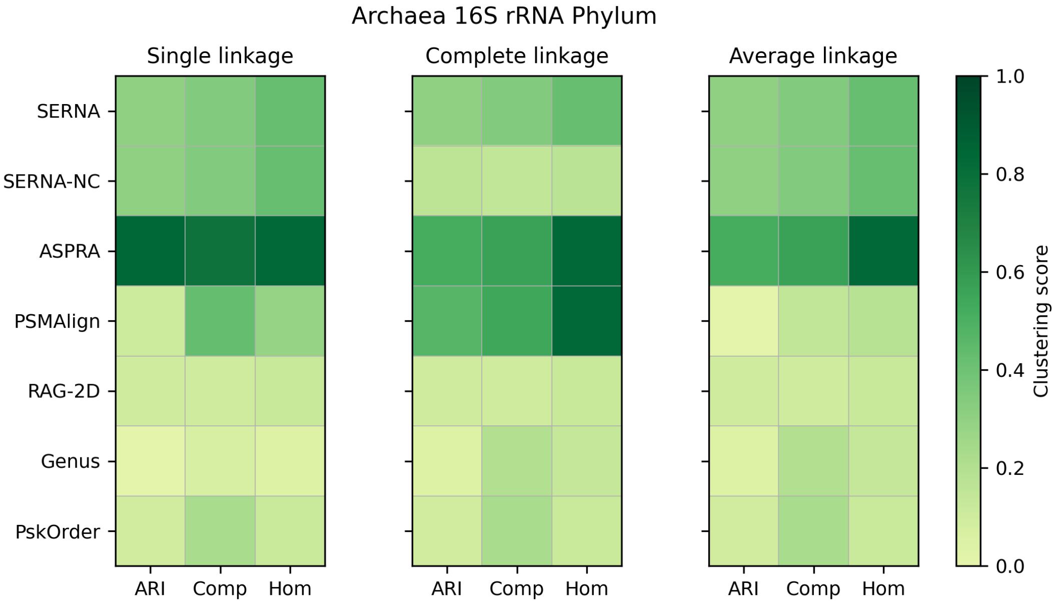

Figure 10: Hierarchical clustering results for Archaea 16S rRNA phylogenetic reconstruction at the Phylum level.

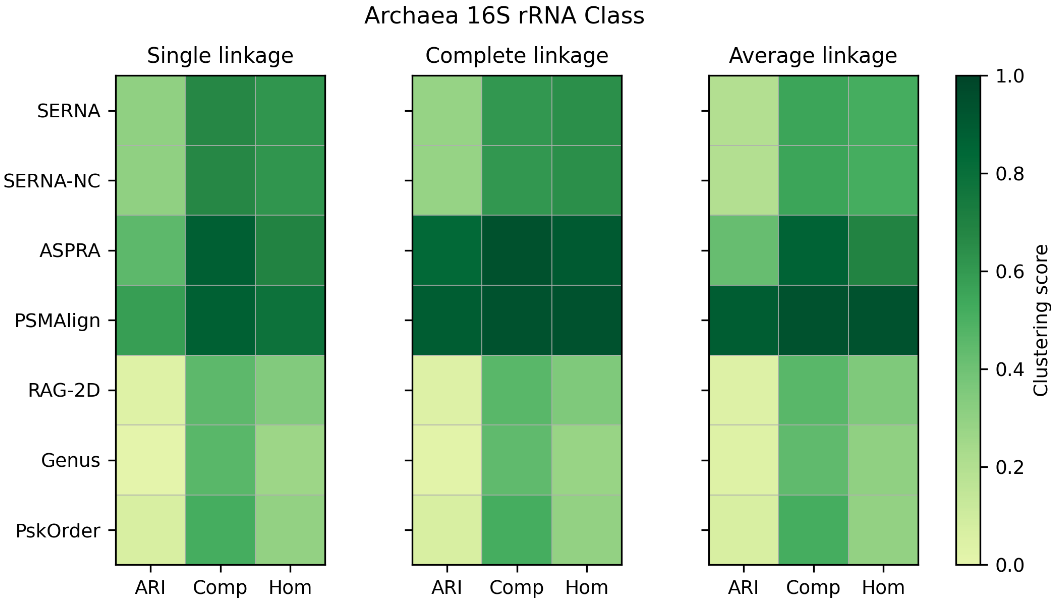

Figure 11: Hierarchical clustering results for Archaea 16S rRNA phylogenetic reconstruction at the Class level.

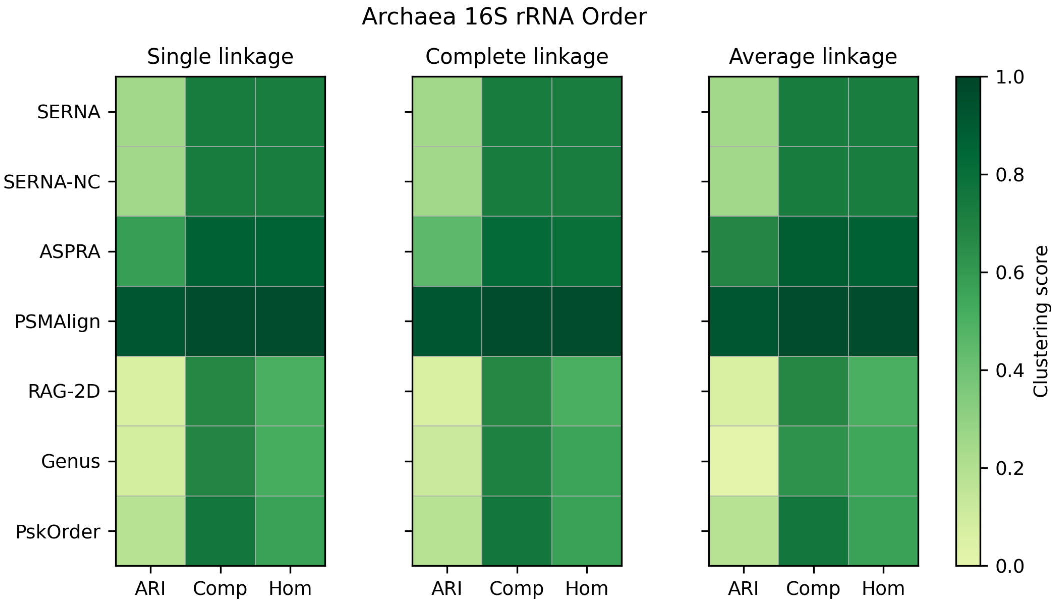

Figure 12: Hierarchical clustering results for Archaea 16S rRNA phylogenetic reconstruction at the Order level.

Bacteria

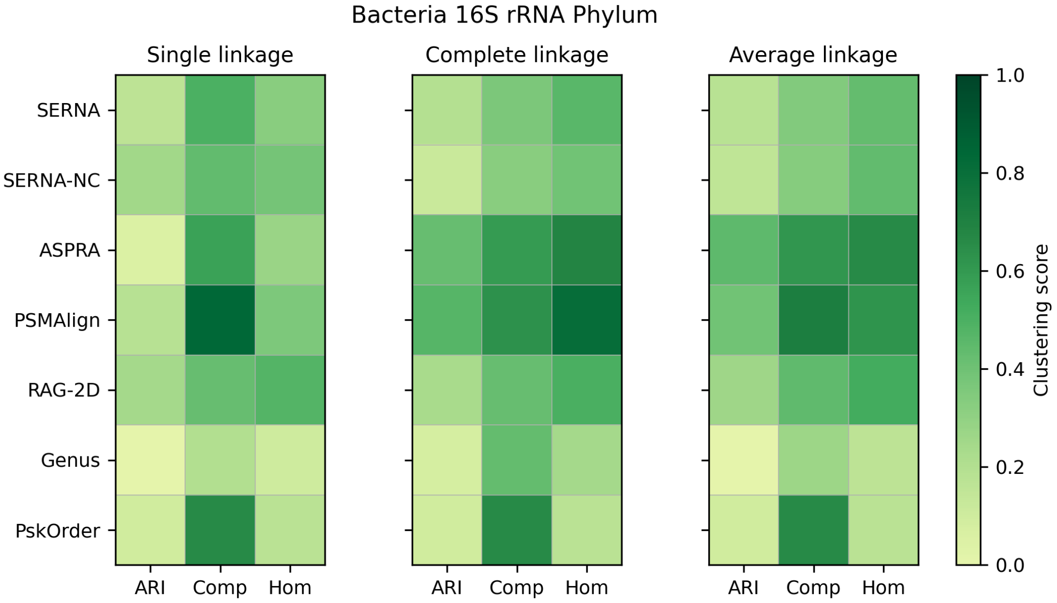

Figure 13: Hierarchical clustering results for Bacteria 16S rRNA phylogenetic reconstruction at the Phylum level.

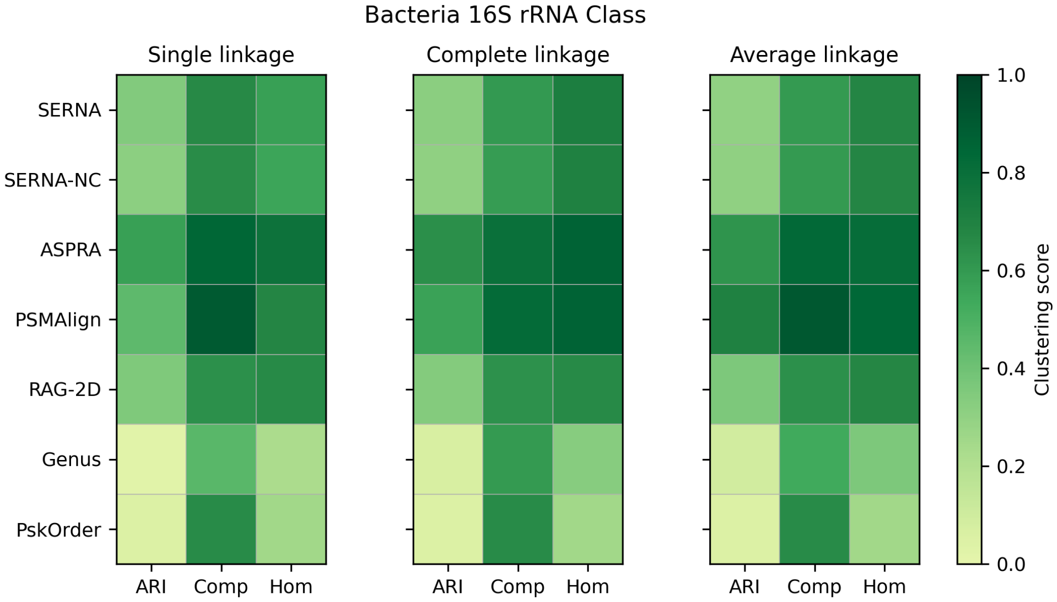

Figure 14: Hierarchical clustering results for Bacteria 16S rRNA phylogenetic reconstruction at the Class level.

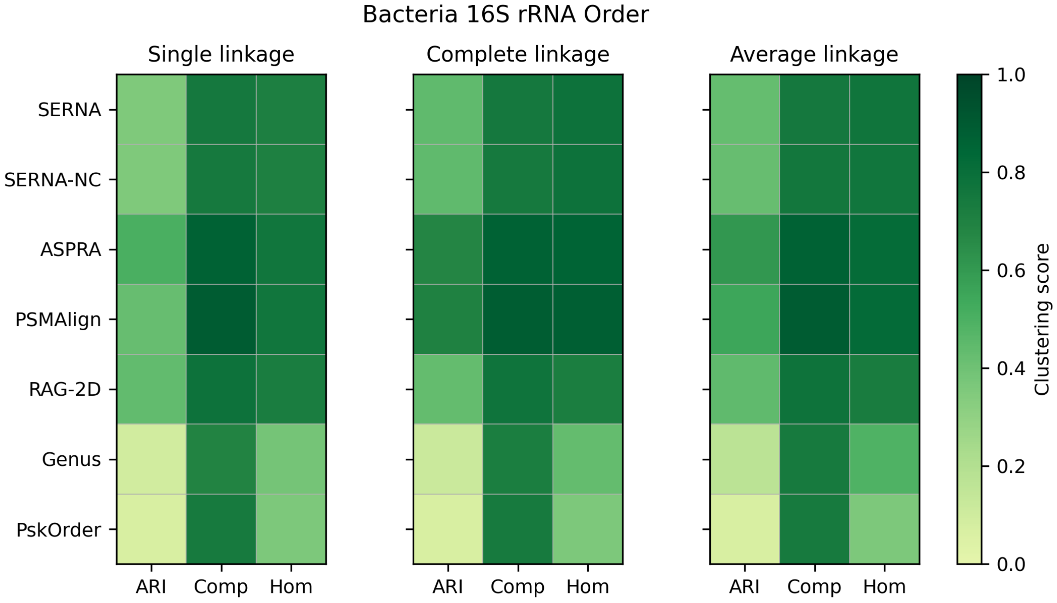

Figure 15: Hierarchical clustering results for Bacteria 16S rRNA phylogenetic reconstruction at the Order level.

Eukaryota

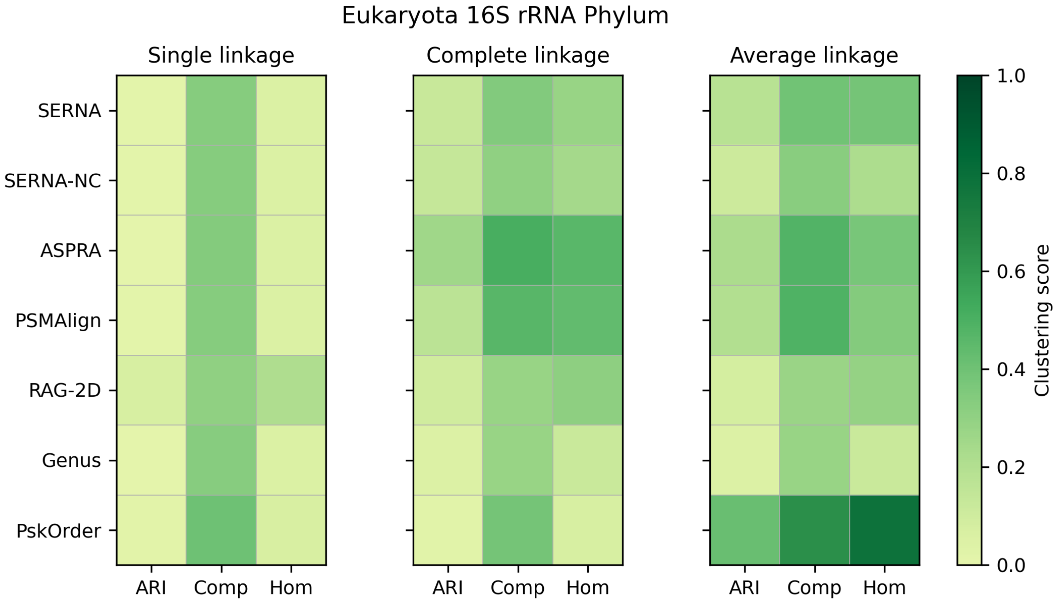

Figure 16: Hierarchical clustering results for Eukaryota 16S rRNA phylogenetic reconstruction at the Phylum level.

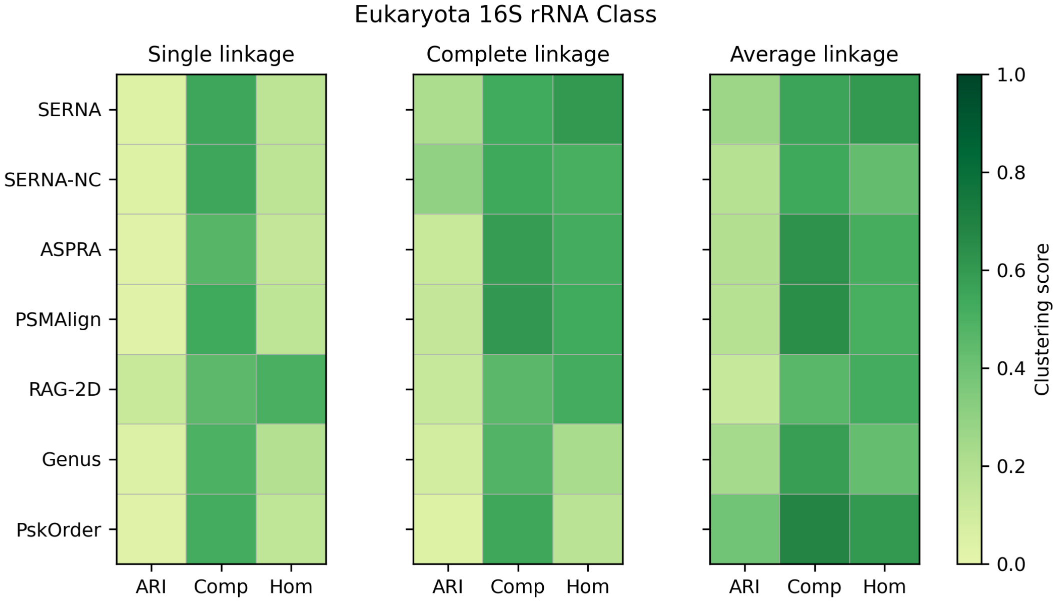

Figure 17: Hierarchical clustering results for Eukaryota 16S rRNA phylogenetic reconstruction at the Class level.

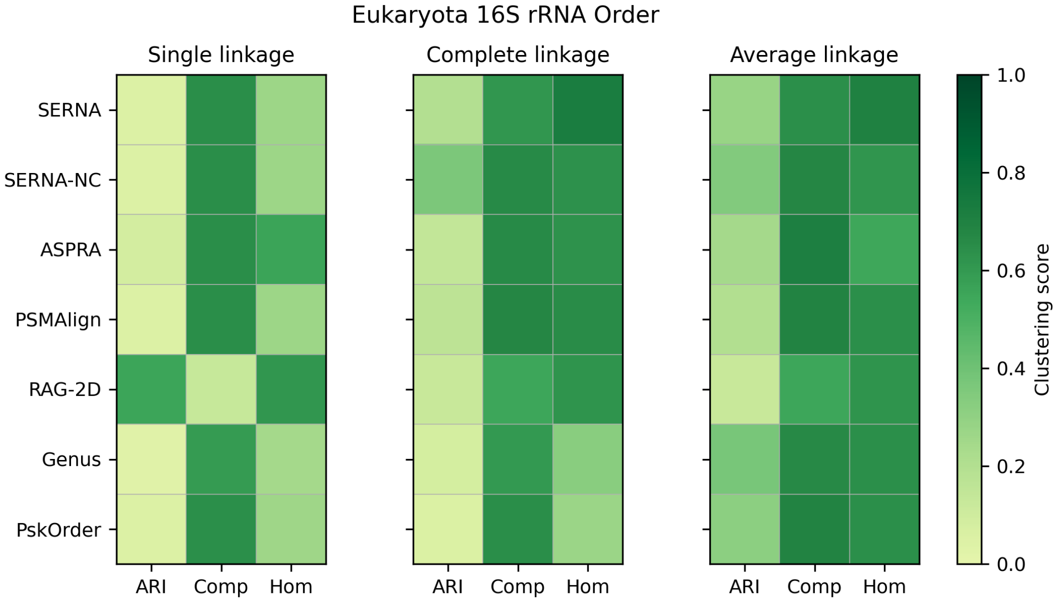

Figure 18: Hierarchical clustering results for Eukaryota 16S rRNA phylogenetic reconstruction at the Order level.

23S Ribosomal RNA

Archaea

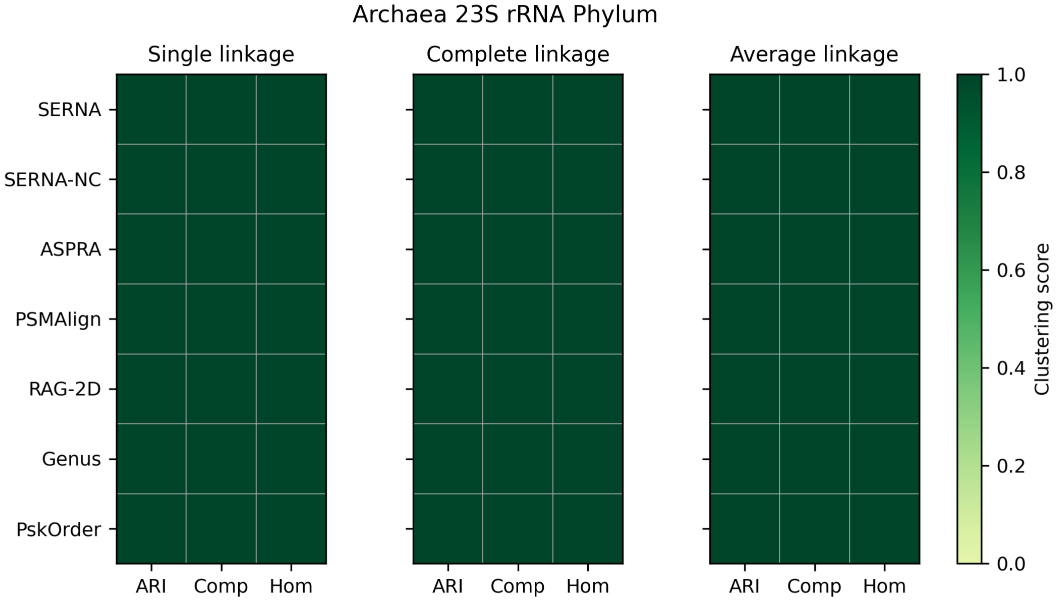

Figure 19: Hierarchical clustering results for Archaea 23S rRNA phylogenetic reconstruction at the Phylum level.

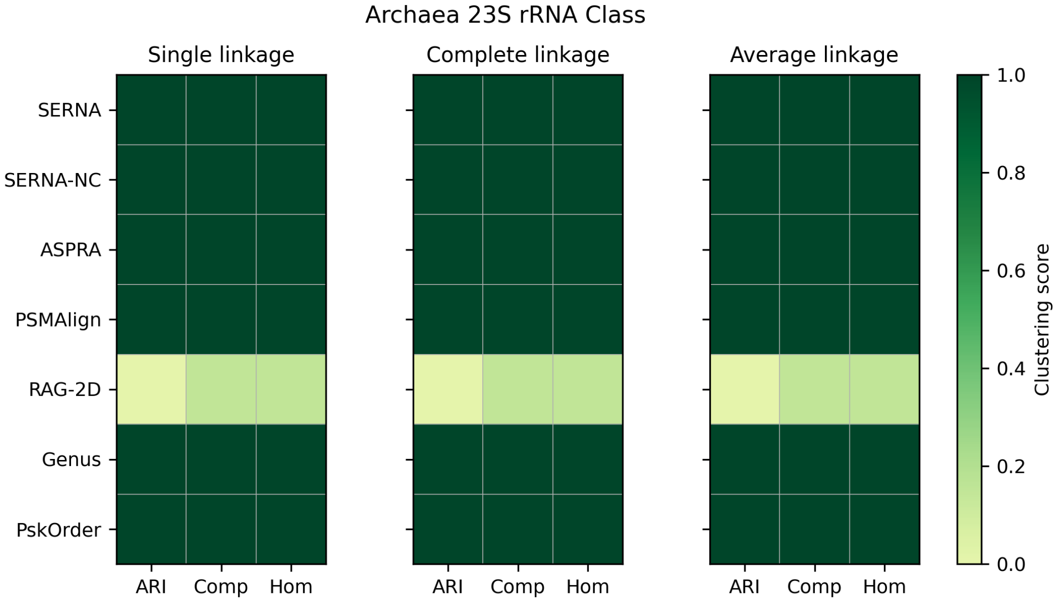

Figure 20: Hierarchical clustering results for Archaea 23S rRNA phylogenetic reconstruction at the Class level.

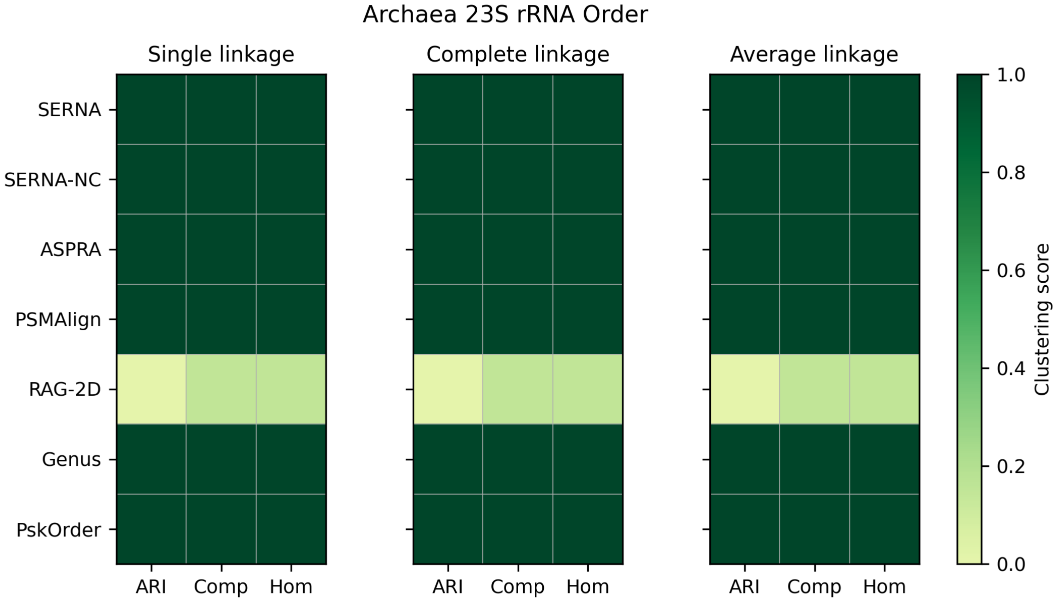

Figure 21: Hierarchical clustering results for Archaea 23S rRNA phylogenetic reconstruction at the Order level.

Bacteria

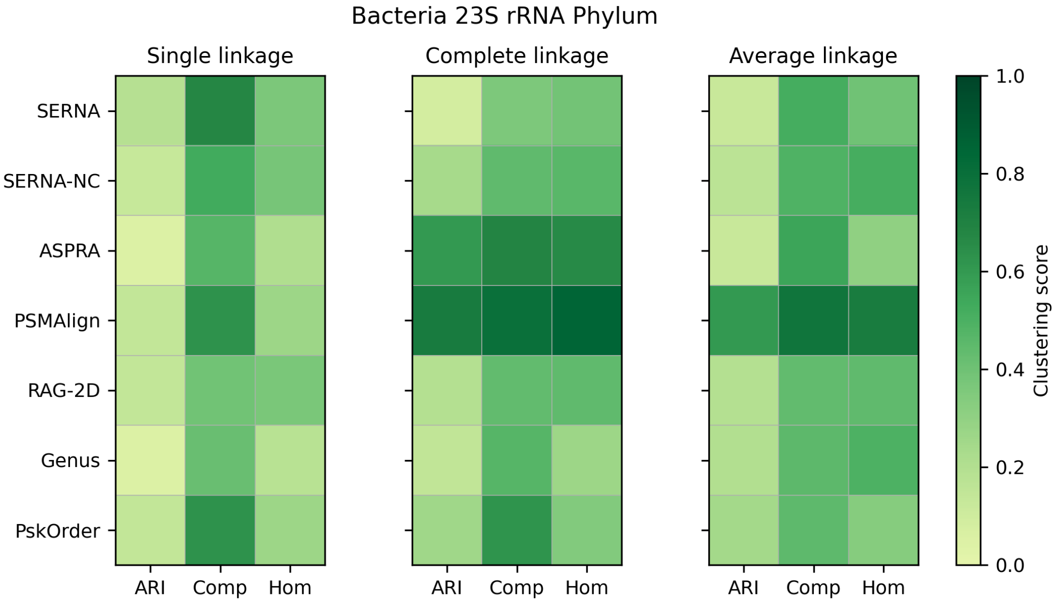

Figure 22: Hierarchical clustering results for Bacteria 23S rRNA phylogenetic reconstruction at the Phylum level.

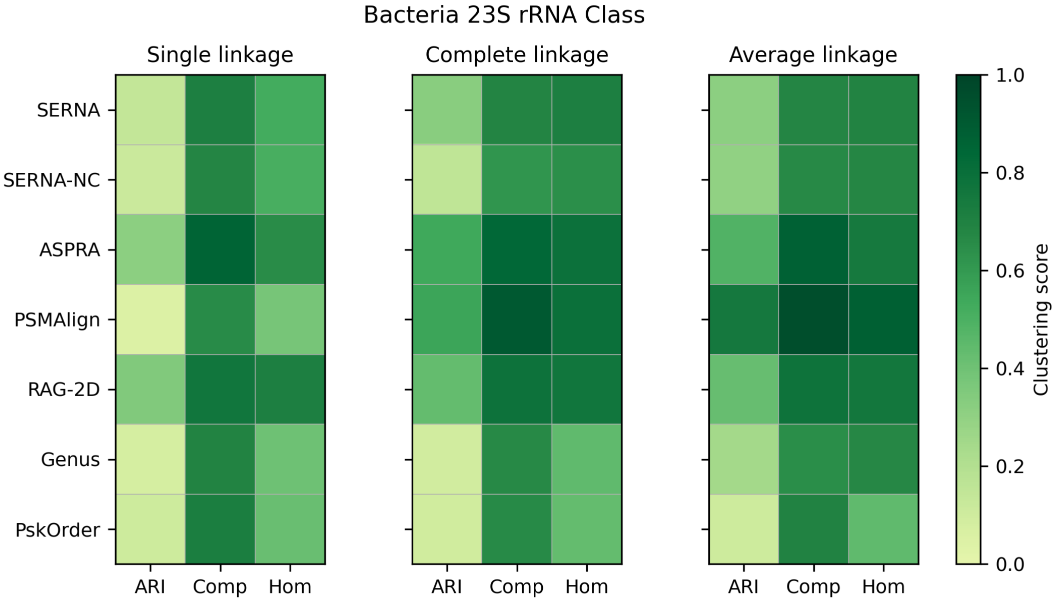

Figure 23: Hierarchical clustering results for Bacteria 23S rRNA phylogenetic reconstruction at the Class level.

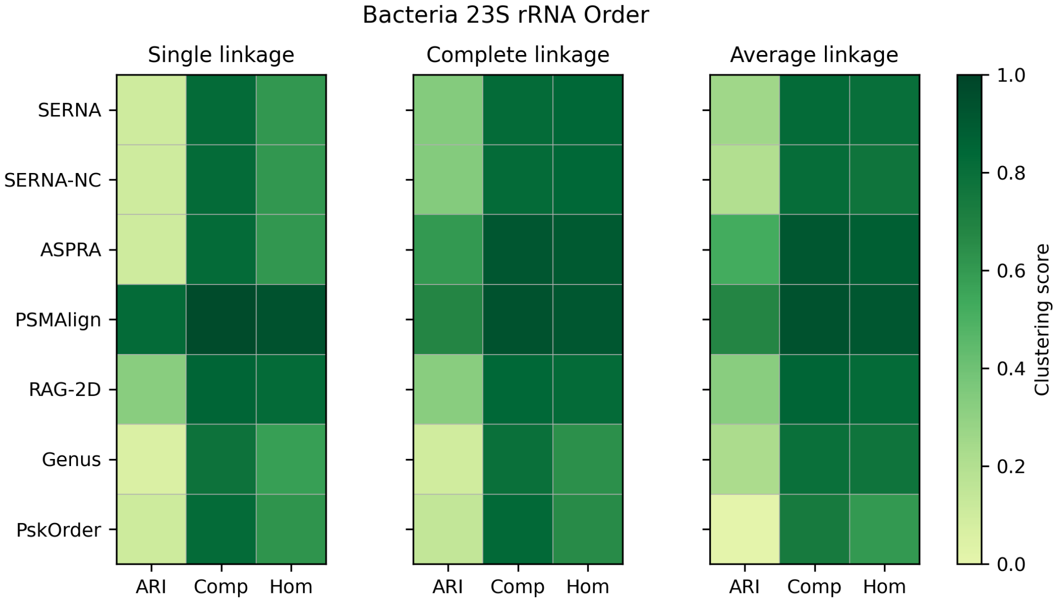

Figure 24: Hierarchical clustering results for Bacteria 23S rRNA phylogenetic reconstruction at the Order level.

Eukaryota

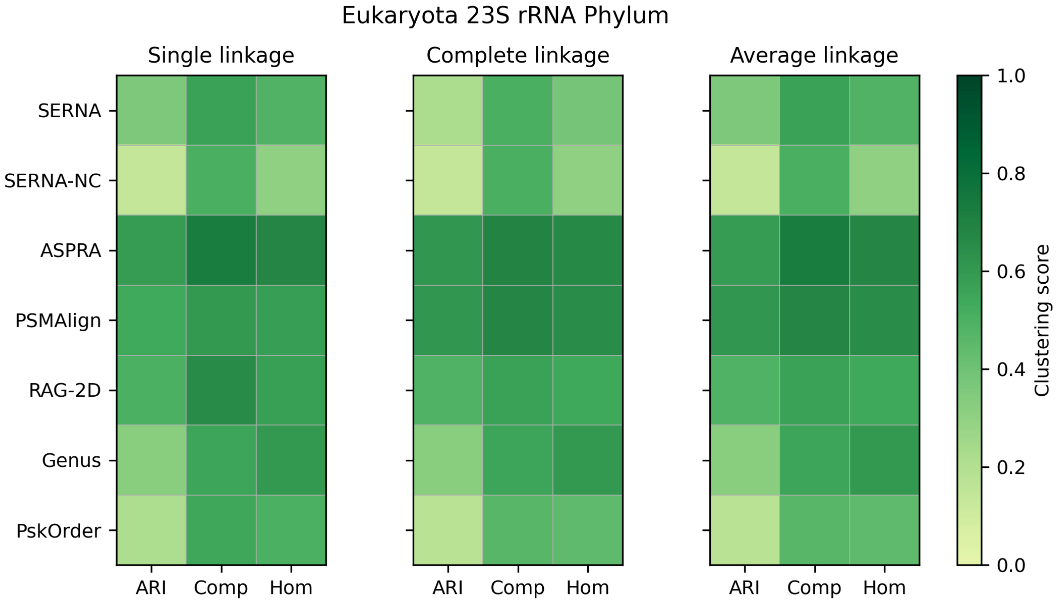

Figure 25: Hierarchical clustering results for Eukaryota 23S rRNA phylogenetic reconstruction at the Phylum level.

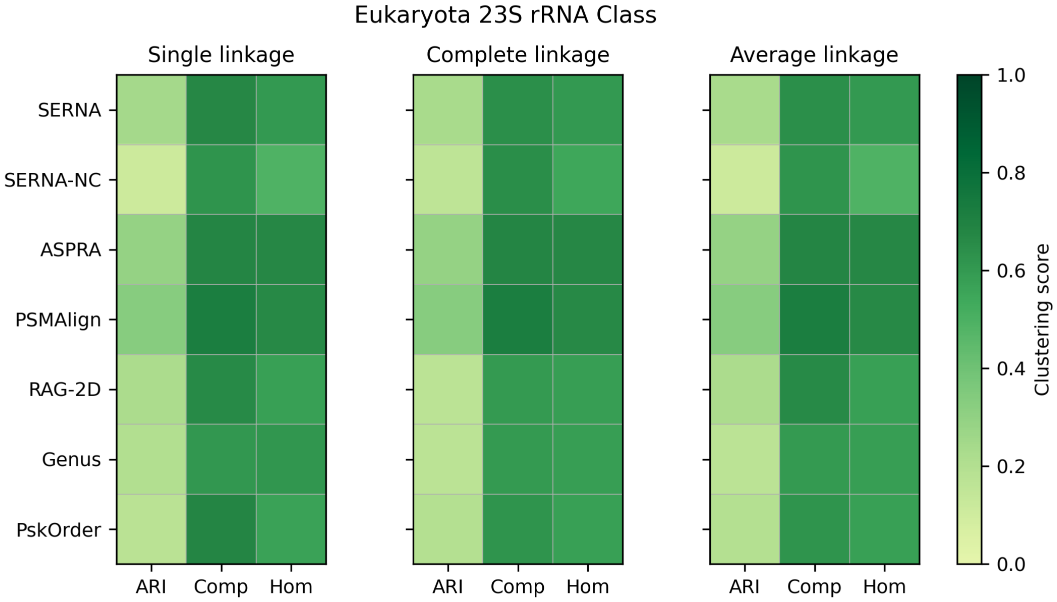

Figure 26: Hierarchical clustering results for Eukaryota 23S rRNA phylogenetic reconstruction at the Class level.

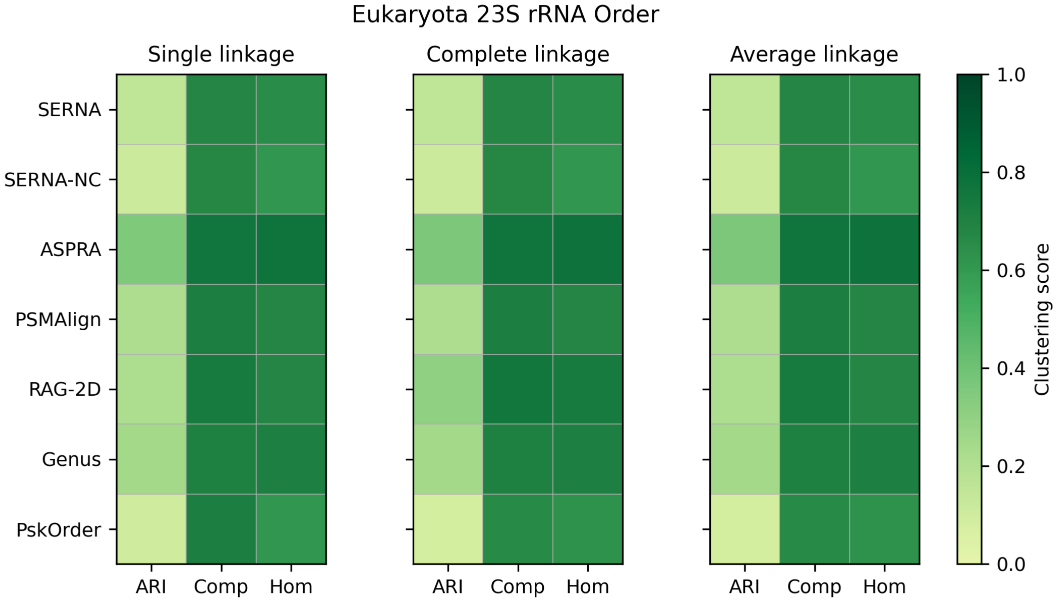

Figure 27: Hierarchical clustering results for Eukaryota 23S rRNA phylogenetic reconstruction at the Order level.