Battle of Dimension Zero Bottleneck Distance Computations: Aliguyon vs. Lumáwig

Jay-Anne Baga Bulauan
jbbulauan@up.edu.ph

Bottleneck Distance

Given two persistence diagrams X and Y,the bottleneck distance between them is

$$d_B(X, Y) = \inf \sup_{x \in X} \|x - \phi(x)\|_\infty$$

Bottleneck Distance

Given two persistence diagrams X and Y,the bottleneck distance between them is

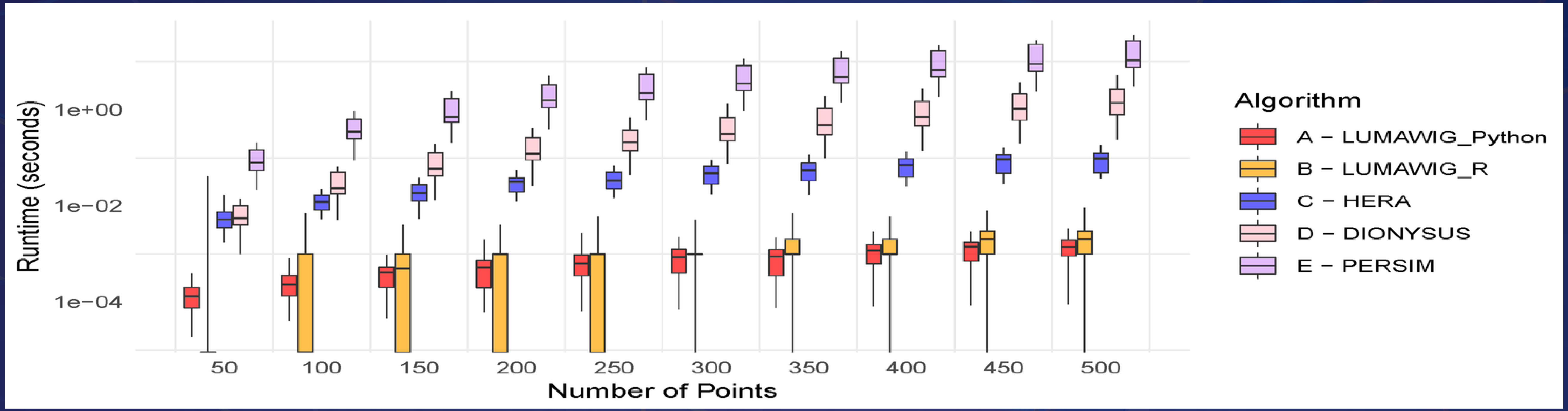
$$d_B(X,Y) = \inf \sup_{x \in X} \|x - \phi(x)\|_\infty$$

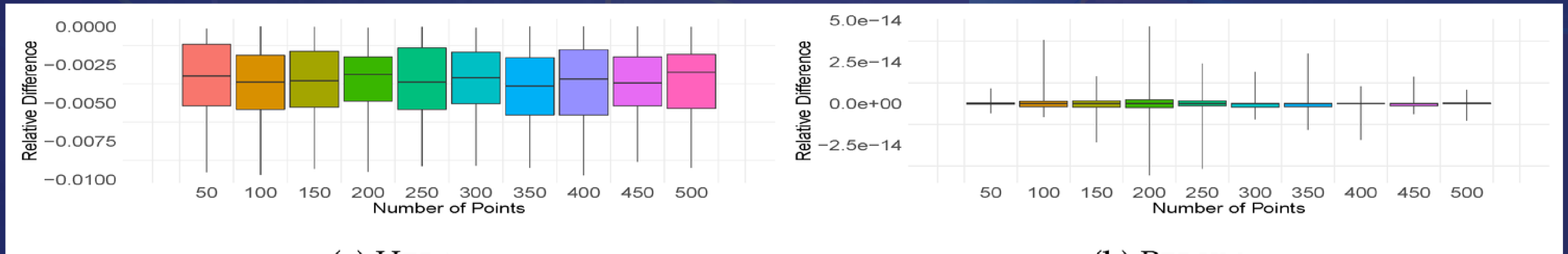
Previous implementations:

- Dionysus implementation (in R)
- Hera (implemented in C++ and wrapped in Python)
- Persim (Python)

Lumáwig (Ignacio et al., 2020)

- a novel algorithm for computing the bottleneck distance between dimension 0 persistence diagrams induced by Rips filtration.





(a) HERA.

(b) PERSIM.

(c) LUMÁWIG$_{PY}$.

(d) LUMÁWIG$_R$.

Ignacio, et al. (2020)
Lumáwig: An efficient algorithm for dimension zero bottleneck distance computation in topological data analysis.
Algorithms, 13(11), 291. https://doi.org/10.3390/a13110291

Lumáwig (Ignacio et al., 2020)

- ϕ - matches points between X and Y according to the relative ranking of death times from largest to smallest, and where unmatched points in Y are matched to the diagonal.
- N = length(X) $\leq$ length(Y)
- $Z = [z_i]_1^{length(Y)}, \quad \text{where} \quad z_i = \begin{cases} |x_i - y_i| & \text{if} \quad i \leq N \\ y_i/2 & otherwise \end{cases}$
- l=argmax(Z)

Lumáwig (Ignacio et al., 2020)

Lemma 1. If $N < length(Y)$ and $\max(Z) \leq y_{N+1}/2$, then $d_B(X,Y) = y_{N+1}/2$, where y_{N+1} is the largest death time of a point in Y matched to the diagonal.

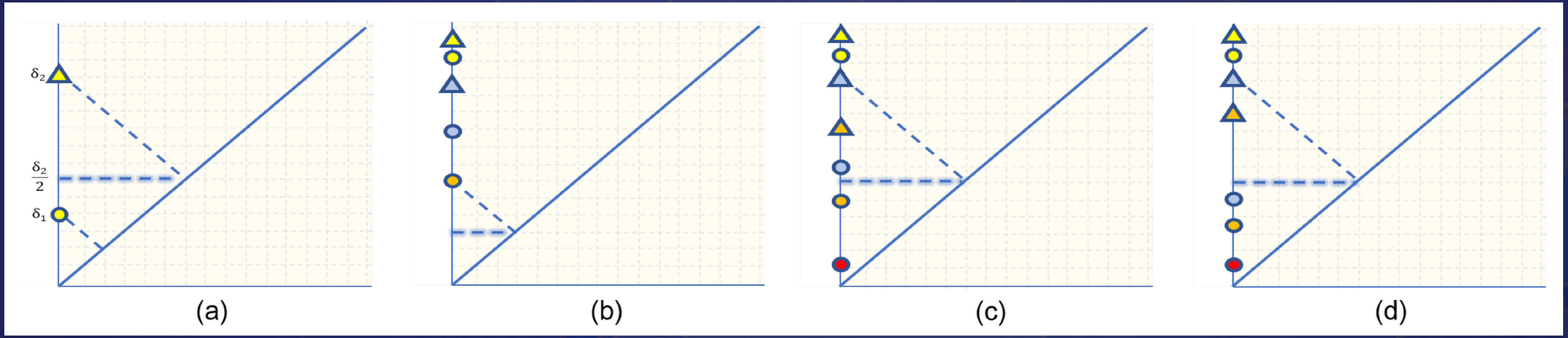
Lemma 2. Let ζ be the second largest entry of Z.

i. If $\max(Z) \leq \max(x_l, y_l)/2$, then $d_B(X,Y) = \max(Z)$.

ii. If $\zeta < \max(x_l, y_l)/2 < \max(Z)$, then $d_B(X,Y) = \max(x_l, y_l)/2$.

iii. If $\zeta \geq \max(x_l, y_l)/2$ and $m \geq l$ for every m such that $z_m \geq \max(x_l, y_l)/2$, then $d_B(X,Y) = \max(x_l, y_l)/2$.

iv. If $\zeta \geq \max(x_l, y_l)/2$ and there exists $m < l$ such that $z_m \geq \max(x_l, y_l)/2$, then there exists a bijection τ between X and Y such that one of the three preceding cases holds and where $\max ||x - \tau(x)||_\infty < \max ||x - \phi(x)||_\infty$.



Aliguyon

$$Z' = [z'_i]_1^{length(Y)} \quad \text{where} \quad z'_i = \begin{cases} \min\{z_i, \max(x_i, y_i)/2\} & \text{if} \quad i \leq N \\ y_N/2 & \text{otherwise} \end{cases}$$

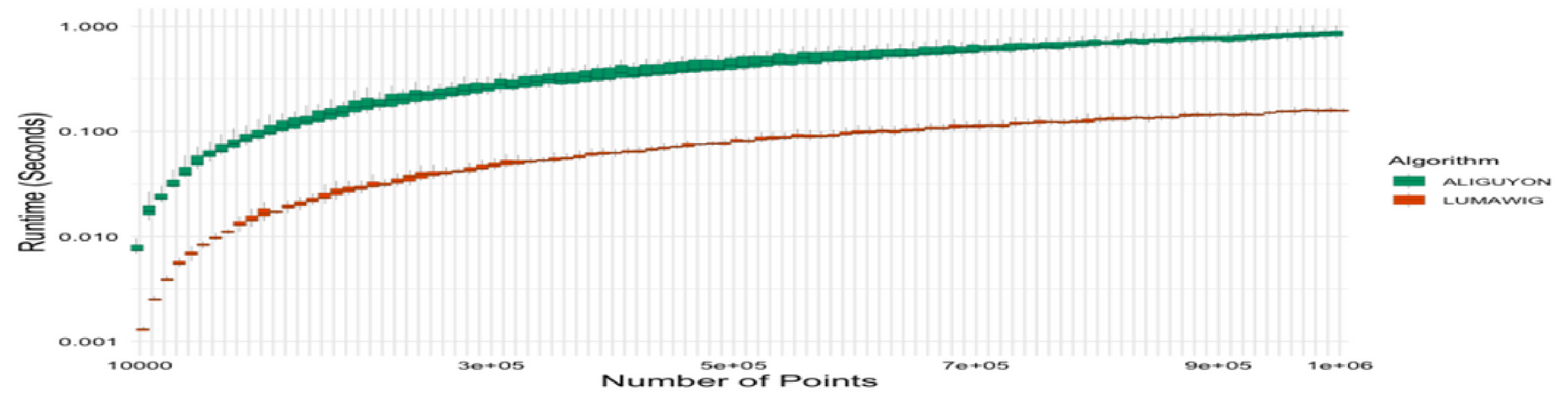
Aliguyon

$$Z' = [z'_i]_1^{length(Y)} \quad \text{where} \quad z'_i = \begin{cases} \min\{z_i, \max(x_i, y_i)/2\} & \text{if} \quad i \leq N \\ y_N/2 & \text{otherwise} \end{cases}$$

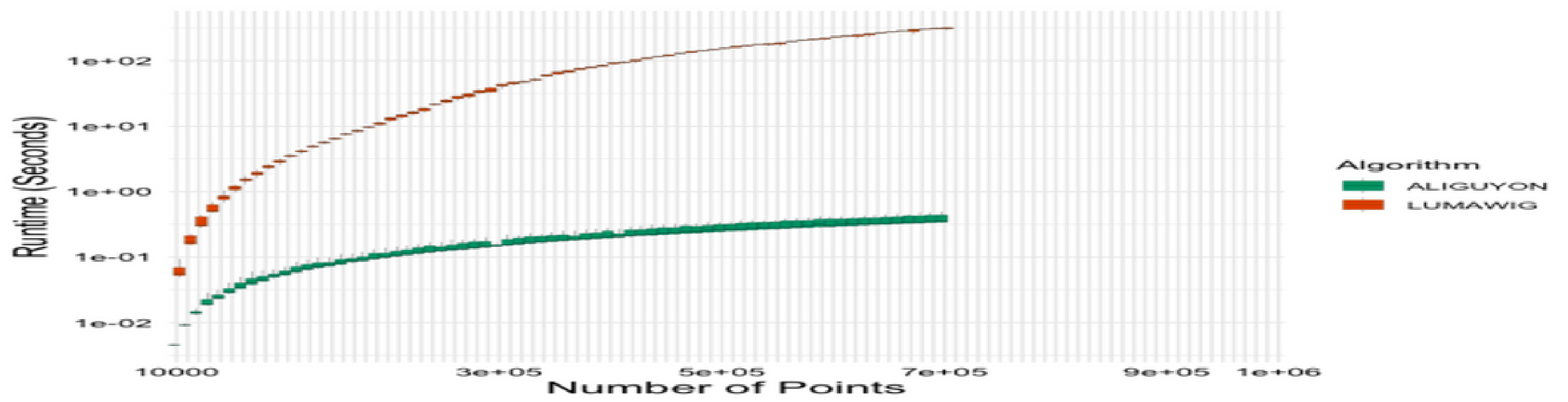
Theorem.

$$d_B(X, Y) = \max(Z').$$

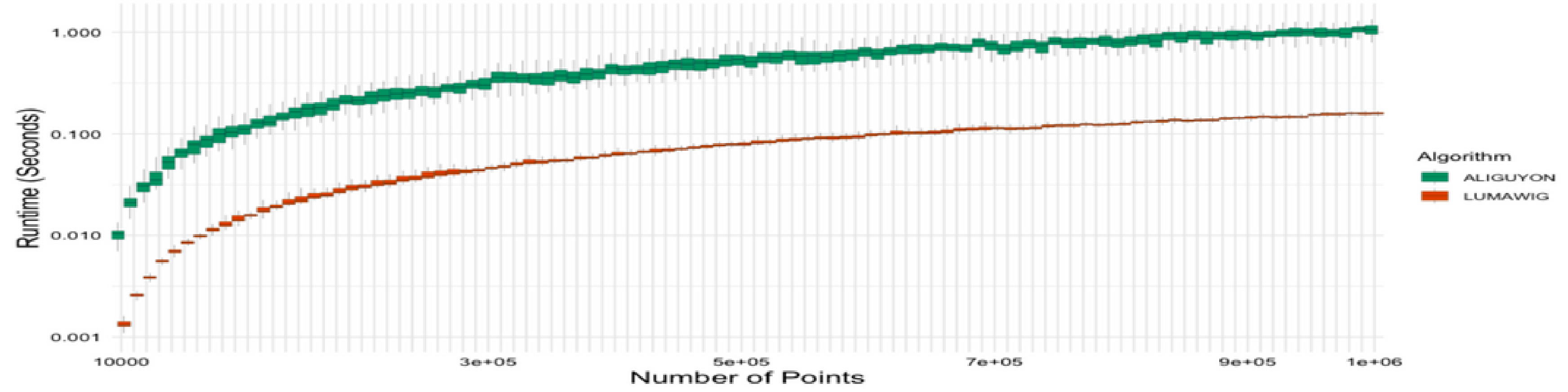
Lumáwig vs. Aliguyon



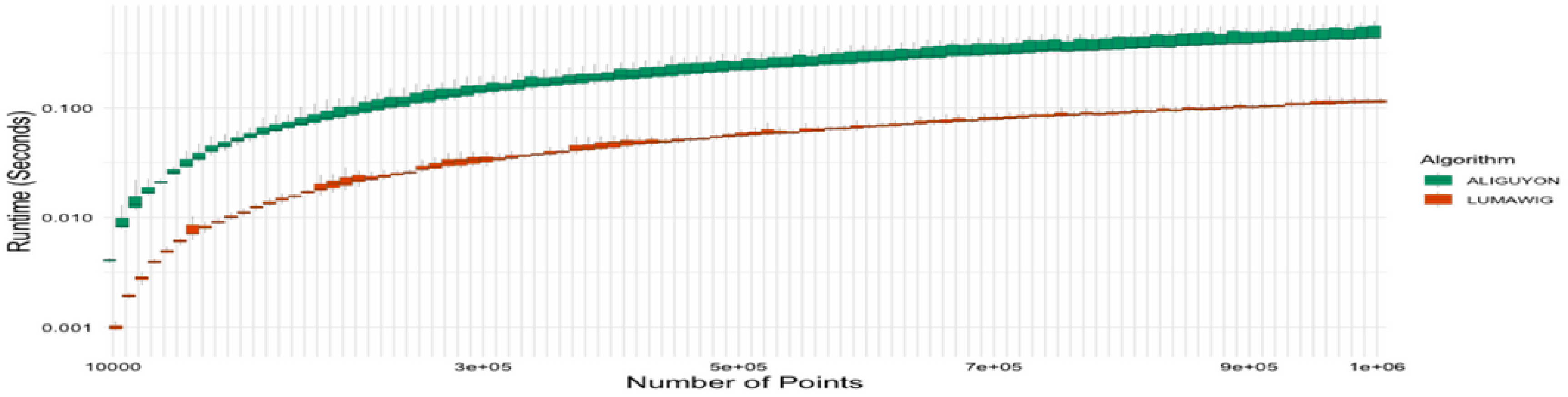
(a) Equal size and range.



(c) Different size but equal range.

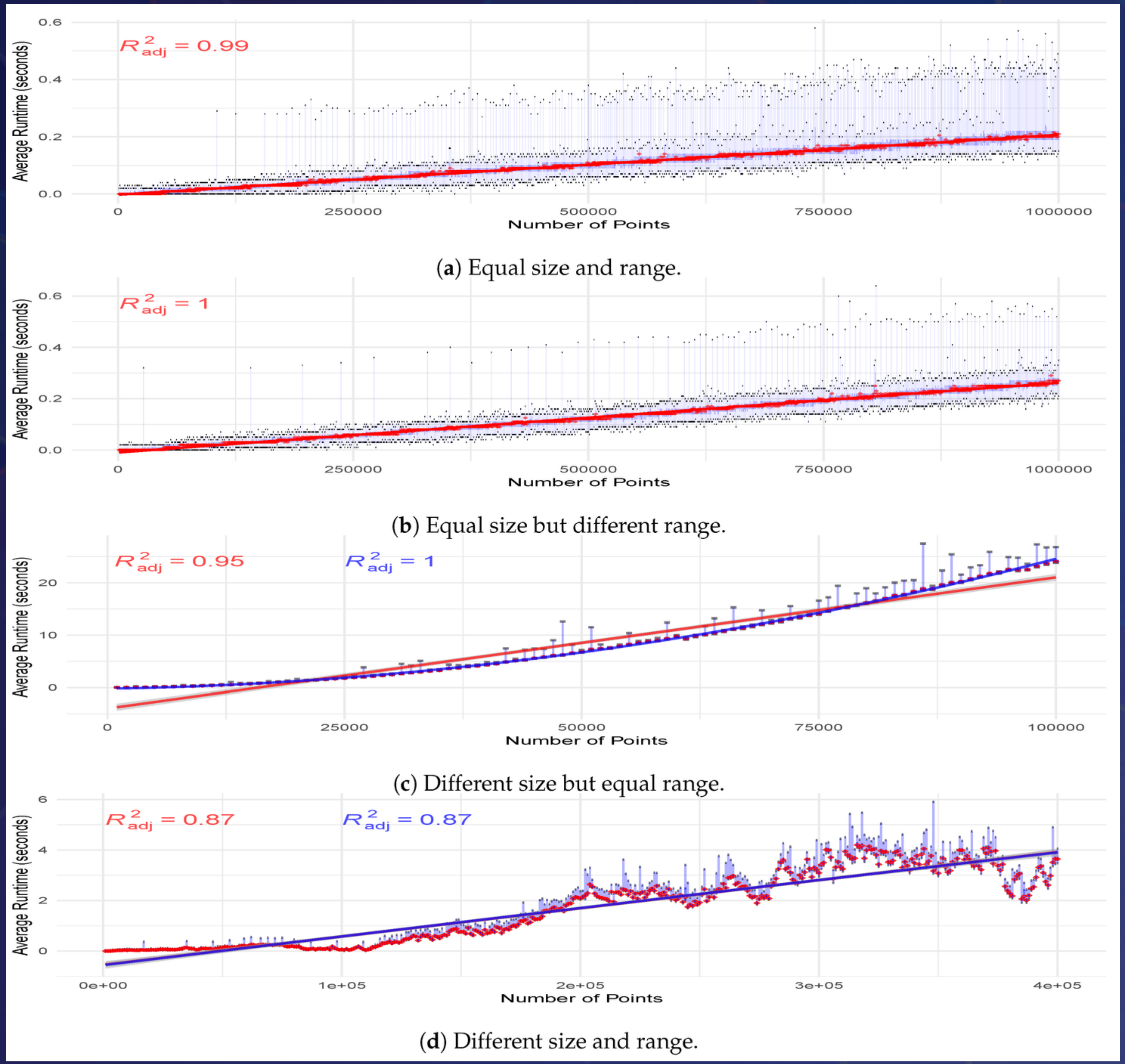


(b) Equal size but different range.



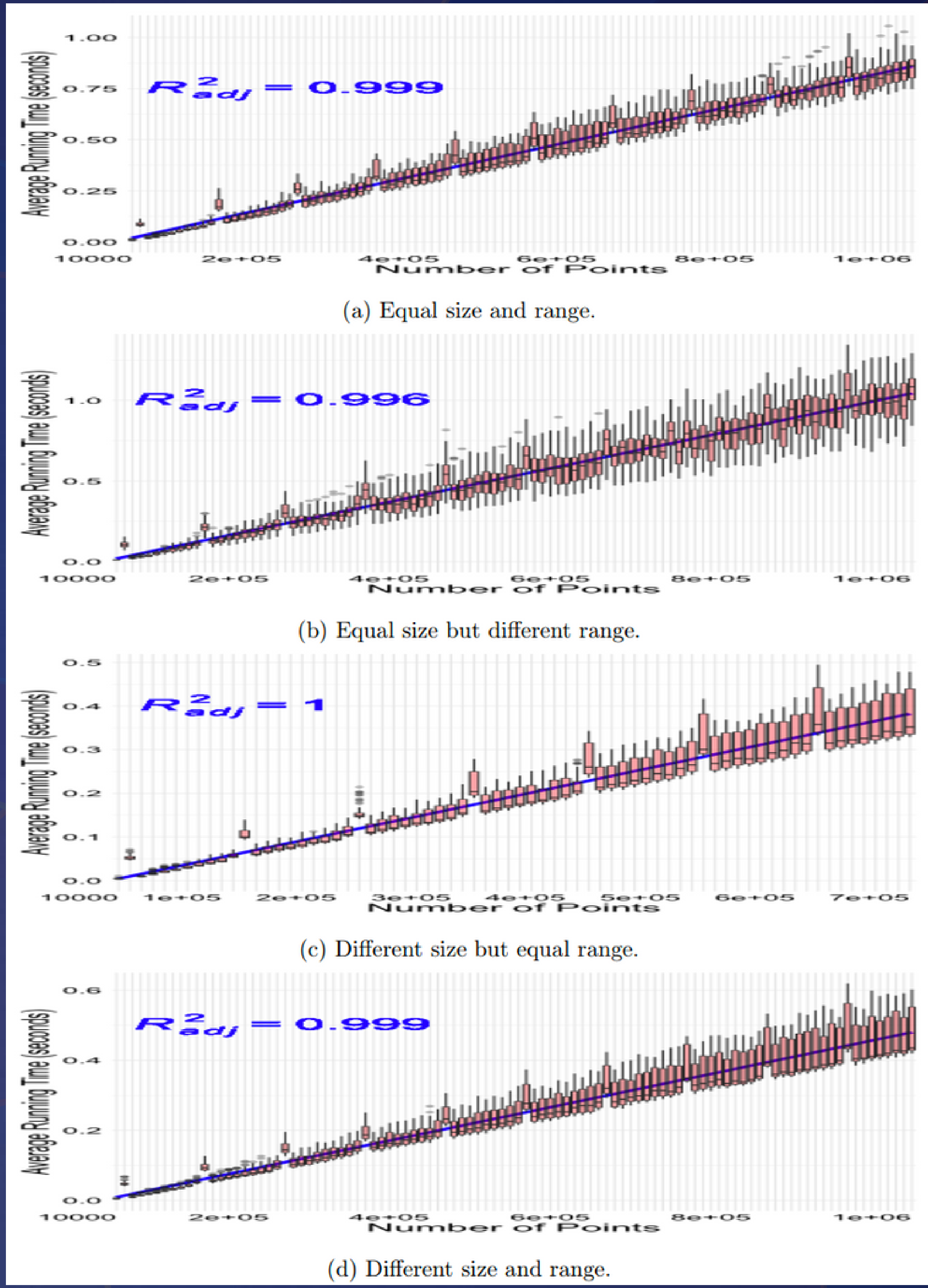
(d) Different size and range.

Lumáwig vs. Aliguyon



(a) Equal size and range.
(b) Equal size but different range.
(c) Different size but equal range.
(d) Different size and range.

Lumáwig



(a) Equal size and range.
(b) Equal size but different range.
(c) Different size but equal range.
(d) Different size and range.

Aliguyon

Ignacio, et al. (2020)
Lumáwig: An efficient algorithm for dimension zero bottleneck distance computation in topological data analysis.
Algorithms, 13(11), 291. https://doi.org/10.3390/a13110291

Battle of Dimension Zero Bottleneck Distance Computations: Aliguyon vs. Lumáwig

Jay-Anne Baga Bulauan
jbbulauan@up.edu.ph