**1. Intra-Class Similarity**

**Compare sequences within the same class** to see how genetically similar members of a protein family are.

* ✅ **Why it’s interesting**: Shows **evolutionary or functional similarity** within a protein type.
* ✅ **What to do**: Pick pairs from the same class (e.g., class 0) and compute **LCS length**.
* 🔁 Aggregate the LCS lengths: average, min, max.

📌 **Conclusion**: Classes with higher intra-similarity likely have more conserved regions — useful for classification or drug targeting.

**🧬 2. Inter-Class Dissimilarity**

**Compare sequences across different classes** (e.g., class 0 vs. class 1).

* ✅ **Why it’s interesting**: Helps verify how distinct one class is from another at the sequence level.
* ✅ **What to do**: Take multiple pairs across two classes and compute **average LCS**.

📌 **Conclusion**: Classes with **low average LCS** are more **functionally or structurally distinct**.

**🧠 3. Find the Most Similar/Dissimilar Class Pairs**

* Compare **each class with every other class**, calculate average LCS.
* Build a **similarity matrix**: 7x7 table with average LCS scores.

📌 **Use this to:**

* Visualize it as a heatmap.
* Detect which classes are evolutionarily closer.

**🧭 4. Discover Conserved Subsequence Patterns**

* Extract **actual LCS strings** within a class.
* Analyze if there's a **common conserved motif** that appears across sequences.

📌 Example:

If class 0 sequences all share a "TGAAC" subsequence — that could be a **conserved domain**.

**🔗 5. Use LCS as a Feature for Clustering or Classification**

* For each DNA sequence, generate a **vector of LCS lengths with representatives from all classes**.
* Use these vectors for:
  + **Clustering** (unsupervised learning)
  + **Nearest-neighbor classification**

📌 Could lead to a **simple classifier** based purely on LCS distances.

**📊 6. Plotting LCS Length Distributions**

* Make boxplots/histograms of LCS lengths for:
  + Intra-class comparisons
  + Inter-class comparisons

📌 Useful to **visualize variation** and **compare stability** of sequences in each class