**Biopython Sequence Annotation Objects**

**1. Core Concepts**

* **Sequence objects**:
* **Seq** and **MutableSeq** for storing biological sequences.
* Handles DNA, RNA, and protein sequences.
* **AnnotatedSequence**:
* Combines sequence with metadata (annotations).
* Found in **SeqRecord** class.

**2. The SeqRecord Object**

* **Definition**:

The central class for sequence annotation in Biopython, encapsulating a sequence via **Seq** Object with annotations and features.

* **Key Attributes**:
* **seq**: The biological sequence (e.g., DNA, RNA, protein).
* **id**: Identifier for the sequence.
* **name**: A short human-readable name.
* **description**: Additional details about the sequence.
* **annotations**: Dictionary of general information (e.g., organism, topology).
* **features**: List of SeqFeature objects for detailed annotation.
* **dbxrefs**: List of database cross-references.

**3. The SeqFeature Object**

* **Definition**:

Represents specific annotations for features (e.g., genes, CDS, exons).

* **Key Attributes**:
* **type**: Type of the feature (e.g.,"gene","CDS").
* **location**: The position of the feature (using **FeatureLocation** class).
* **qualifiers**: Dictionary of additional metadata (e.g.,"gene" → "BRCA1").
* **Location Attributes**:
* Start/End positions (.start, .end), just like with string objects.
* Strand information (+1, -1, or None).

**4. Common Operations**

* **Create a SeqRecord**:

***from Bio.Seq import Seq***

***from Bio.SeqRecord import SeqRecord   
seq = Seq("ATGCGTACGTA")   
record = SeqRecord(seq, id="ABC123", name="Example", description="Sample DNA sequence")***

***from Bio.SeqFeature import SeqFeature, FeatureLocation***

***feature = SeqFeature(FeatureLocation(0, 6), type="gene", qualifiers={"gene": "example"}) record.features.append(feature)***

***print(record.id) # Output: ABC123***

***print(record.features[0]) # Output: gene at [0:6****]*

**5. Key Use Cases**

* Genome annotation.
* Parsing data from GenBank files.
* Creating annotated records for analysis and visualization.

**6. Some Methods**

* **know from str objects:**

***print(len(record.seq))*** *# print length of sequence stored in record*

***new = record + 'ATG'*** *# add str to sequence, position defindes insert position (left, right)*

***print(record.seq.upper())*** *# format sequence to upper case letters*

***edited = record[:20]*** *# slice sequence, careful about annotations when slicing SeqRecord object!*