**Course: Advanced Bioinformatics**

**Module title: Alignment Module**

**Module no. : 198**

This module contains several classes and methods that allow users to perform pairwise and multiple sequence alignment.

Pairwise sequence alignment: For optimal global alignment, BioJava implements the Needleman-Wunsch algorithm and for performing local alignments the Smith and Waterman's algorithm has been implemented. The outputs of both local and global alignments are available in standard formats.

An example on how to use the libraries is shown below.

**protected** void align(String uniProtID\_1, String uniProtID\_2, PairwiseSequenceAlignerType alignmentType) **throws** IOException, Exception {

ProteinSequence proteinSeq1 = FastaReaderHelper.readFastaProteinSequence((**new** URL(String.format

("http://www.uniprot.org/uniprot/%s.fasta", uniProtID\_1))).openStream()).get(uniProtID\_1);

ProteinSequence proteinSeq2 = FastaReaderHelper.readFastaProteinSequence((**new** URL(String.format

("http://www.uniprot.org/uniprot/%s.fasta", uniProtID\_2))).openStream()).get(uniProtID\_2);

SequencePair<ProteinSequence, AminoAcidCompound> result = Alignments.getPairwiseAlignment(proteinSeq1, proteinSeq2,

alignmentType, **new** SimpleGapPenalty(), **new** SimpleSubstitutionMatrix<AminoAcidCompound>());

System.out.println(result.toString());

}

An example call to the above function would look something like this:

**For Global Alignment**

align("Q21691", "Q21495", PairwiseSequenceAlignerType.GLOBAL);

**For Local Alignment**

align("Q21691", "Q21495", PairwiseSequenceAlignerType.LOCAL);

In addition to these two algorithms, there is an implementation of Guan–Uberbacher algorithm[[33]](https://en.wikipedia.org/wiki/BioJava#cite_note-33) which performs global sequence alignment very efficiently since it only uses linear memory.

For **Multiple Sequence Alignment**, any of the methods discussed above can be used to progressively perform a multiple sequence alignment.