#Python code for generating *k-mers* of any defined length form an input biological sequence

*kmerslicer* is a source code, for constructing/generating generate k-mer (either peptides or nucleotide) sequences of any desired length, based on the sliding window method across given FASTA sequences from an input file.

Written by Stephen A. James at School of Data Sciences, Perdana University, Malaysia

Email: [james@perdanauniversity.edu.my](mailto:james@perdanauniversity.edu.my)

**Description of Code**:

This is a Python code(s) designed to generate k-mers and write the output by either appending

k-mer; Accession number and index number e.g.

A)

MSDTGQMEE; AAK77753.1; 0000001

.

.

.

AAAAAAAAA; AAK77753.1; 0000341 0000342 0000343 0000344 0000369 0000370 0000371 0000372 0000397 0000398 0000399

OR

B)

MSDTGQMEE; AAK77753.1; 0000005

.

.

.

AAAAAAAAA; AAK77753.1; 0000345-0000348 0000373-0000376 0000401-0000403

The code incorporates modules such as Biopython, Statistics, and in addition System-specific parameters and functions in the case of kmerslicer2 source code. The source code in “kmerslicer1” generates the output as shown in ‘A’ (i.e. append k-mer; IDs; serial-index number); while the source code in kmerslicer2 generates an output as in ‘B’ but with index number appended based the median residue in the k-mer length. In addition, in kmerslicer2 the occurrence of consecutive index number for a k-mer, it is presented in the form of range value. For the basic implementation of the code see below:

**Input Format:**

A FASTA sequences with a description line (i.e. “> accessions number and sequences detail”) as start of each sequence

**Code Execution:**

#For code with define input and output file variable

*python* sourceCode.py InputFile OutputFile

#For code with define PATH for (input and output) files

*python* sourceCode.py

**Installation:**

*kmerslicer* supports Python 3.x version

For the source code implementation installation of Biopython is required.

**Usage:**

*kmerslicer* is capable of slicing large datasets of biological sequences to any desire length that can be used for and structural/functional characterization.

**Generating Sequences:**

The overlapping example shows below present how *kmerslicer* can generate a dictionary of million sequences as output from small to large FASTA sequences:

Overlapping peptide

1 2 3 4 5 6 7 8 9

GAAGKAAKSSEMQWTATA……

W

GAAGKAAKS

AAGKAAKSS

AGKAAKSSE

*Pn*

*Pn*

W

Construction of a *k-mers* dictionary. For a given window size (W), a complete repertoire of *k-mers* (designated by *Pn*, where n is the number of k-mers) overlapping by W-1 are generated and populate the dictionary