Recipe 2.10. Generate modified peptides from a peptide and a set of rules

Table of contents

1. Problem

You want to generate all possible modified peptides given an initial peptide and a set of rules defining the site(s) of modification and the nature of applied modifs.

2. Solution

The `PeptideEditorFactory` creates new instances of modified `Peptide`s given the set of `EditionRule`s and the initial `Peptide`.

A `EditionRule` is composed of 2 parts:

1. a `AAMotifMatcher` specify site of editions
2. a `EditionAction` specify the modification to add in the specific mode (fixed or variable)

```
import org.expasy.jpl.core.mol.polymer.pept.rule.*;
import import org.apache.commons.collections15.Transformer;

// the initial peptide
Peptide peptide = new Peptide.Builder("MQRSTATGCFKL").build();

// the rules to apply
EditionRule ruleFixed =
        new EditionRule("1", AAMotifMatcher.newInstance("[KNQR]"),
EditionAction.newFixedModifAction(LossModification.AMMONIUM_LOSS));

EditionRule ruleVariable =
        new EditionRule("2", AAMotifMatcher.newInstance("[ST]"),
EditionAction.newVariableModifAction(LossModification.WATER_LOSS));

List<EditionRule> rules = new ArrayList<EditionRule>();
rules.add(ruleFixed);
rules.add(ruleVariable);

// the factory makes the process
PeptideEditorFactory factory =
        PeptideEditorFactory.newInstance(rules);

List<Peptide> peptides = factory.transform(peptide);

// display:
// H_MQ(H-3N-1)R(H-3N-1)STAT(H-2O-1)GCFK(H-3N-1)L_HO
// H_MQ(H-3N-1)R(H-3N-1)ST(H-2O-1)ATGCFK(H-3N-1)L_HO
// H_MQ(H-3N-1)R(H-3N-1)ST(H-2O-1)AT(H-2O-1)GCFK(H-3N-1)L_HO
// H_MQ(H-3N-1)R(H-3N-1)S(H-2O-1)TATGCFK(H-3N-1)L_HO
// H_MQ(H-3N-1)R(H-3N-1)S(H-2O-1)TAT(H-2O-1)GCFK(H-3N-1)L_HO
// H_MQ(H-3N-1)R(H-3N-1)S(H-2O-1)T(H-2O-1)ATGCFK(H-3N-1)L_HO
// H_MQ(H-3N-1)R(H-3N-1)S(H-2O-1)T(H-2O-1)AT(H-2O-1)GCFK(H-3N-1)L_HO
for (Peptide peptide : peptides) {
        System.out.println(peptide);
}
Assert.assertEquals(7, peptides.size());
```

3. Discussion

By now `EditionAction` is only able to handle modifications adds. In the future, I would like to integrate protein/peptide digestion here as a cut is an edition.

4. See Also

See also how to generate tuples and how to create modifs or to add them in peptides.