

Aktionssequenz-Alignment

Automatisiertes Sequenz-Alignment zur Analyse von Aktionssequenzen.

Daten

Die Daten stammen aus der CMU MMAC Databse sowie dem TUM Kitchen Data Set und sind unter folgenden Links zu finden:

- <http://kitchen.cs.cmu.edu/labels.php>
- <http://ias.in.tum.de/software/kitchen-activity-data>

Haupt-Klassen und Methoden

- **LabelsCSVFileReader**

- `getSequence(String)`

Zum Einlesen der Label-Dateien des TUM Kitchen Data Sets

```
LabelsCSVFileReader reader = new LabelsCSVFileReader();  
ArrayList<ActionElement> seq = reader.getSequence("Pfad zur CSV-Datei");
```

- **LabelsDATFileReader**

- `getSequence(String)`

Zum Einlesen der Label-Dateien der CMU MMAC Database

```
LabelsDATFileReader reader = new LabelsDATFileReader();  
ArrayList<ActionElement> seq = reader.getSequence("Pfad zur DAT-Datei");
```

- **Ontology**

- `getWupSimilarity(String, String)`

Zum Laden der Ontologie und Berechnen der WUP-Ähnlichkeit

```
Ontology ontology = new Ontology("URL der Ontologie");  
ontology.getWupSimilarity("entity1", "entity2");
```

- **HierarchicStructure und Transformer**

Zur Berücksichtigung hierarchischer Strukturen

```
HierarchicStructure hierarchy = new HierarchicStructure("Pfad zur TXT-Datei");  
Transformer transformer = new Transformer(hierarchy);
```

TXT-Datei im Format:

```
action1: subaction- ... -subaction  
action2: subaction- ... -subaction  
.  
.  
.
```

- **NeedlemanWunsch**

– printAlignment()

Zur Berechnung und Darstellung eines globalen Alignments zweier Aktionssequenzen

```
NeedlemanWunsch nw = new NeedlemanWunsch(seq1, seq2);  
nw.printAlignment();  
//mit WUP-Ähnlichkeit:  
NeedlemanWunsch nw = new NeedlemanWunsch(seq1, seq2, ontology);
```

- **SmithWaterman**

– printAlignment()

Zur Berechnung und Darstellung der lokalen Alignments zweier Aktionssequenzen

```
SmithWaterman sw = new SmithWaterman(seq1, seq2);  
sw.printAlignment();  
//mit WUP-Ähnlichkeit:  
SmithWaterman sw = new SmithWaterman(seq1, seq2, ontology);
```

- **ConfusionMatrix**

- printConfusionMatrix()

Zur Berechnung und Darstellung einer Konfusions-Matrix von mehreren Aktionssequenzen

| |
|---|
| <pre>ArrayList<ActionSequence> list; ConfusionMatrix cm = new ConfusionMatrix(list); cm.printConfusionMatrix();</pre> |
| <pre>//mit WUP-Ähnlichkeit: ConfusionMatrix cm = new ConfusionMatrix(list, ontology);</pre> |