電子計算機入門

池田, 大輔
九州大学情報基盤センター

電子計算機入門
第12回

池田 大輔

daisuke@cc.kyushu-u.ac.jp

情報基盤センター

目次

- 前回課題の回答例
  - 置換（復習）
- 正規表現
  - 正規表現による検索
  - 正規表現による置換

前回課題の解答例

- FASTA 形式の各遺伝子を 1 本の配列として出力

```
if __name__ == '__main__':
    import sys
    f = open(sys.argv[1], 'r')
    seq = []
    line = f.readline()
    while line:
        if line.find('>') < 0:
            seq.append(line)
        else:
            print "(", ''.join(seq), ")"
            seq=[]
        line = f.readline()
```

出力例

```
(    )
(  ATGA
)
(  ATGCGA
TGGAAA
GGTGGC
)
```

- 各行は連結されているが、改行文字が残っているため、１本の配列に見えない

置換（復習）

- 文字列 `old` を `new` に置きかえる場合

  `replace(old, new)`

  - `replace` は置きかえた後の文字列を返す

置換（復習）

- 文字列 old を new に置きかえる場合

```
replace(old, new)
```

  - replace は置きかえた後の文字列を返す

- 例

```
var="abcd"
var=var.replace("ab", "AB")
```

置換による改行文字の削除

- 改行文字は \n で表わされる
  - タブは \t
  - 日本語環境では \ は円記号 (¥) である

置換による改行文字の削除

- 改行文字は \n で表わされる
  - タブは \t
  - 日本語環境では \ は円記号 (¥) である

```
str = '''
abc# '(シングルクオート)3 つで囲めば途中改行も o.k.
def
ghi'''
print str
# 改行こみで出力される
```

置換による改行文字の削除

- 改行文字は \n で表わされる
  - タブは \t
  - 日本語環境では \ は円記号 (¥) である

```
str = '''
abc# '(シングルクオート)3 つで囲めば途中改行も o.k.
def
ghi'''
print str
# 改行こみで出力される
print str.replace('\n', "")
```

実習

- さきほどの例をプログラムに書き、`replace` を使って、改行文字を削除しなさい

正規表現

- 正規表現 (Regular Expression) とは
  - 複数 (無限個でも可) の語や文字列を簡単に表わす手段
  - 例：“ACCT で始まり T で終わる配列”
- 多くのプログラミング言語 (Perl や Ruby など) で利用可能
  - ただし、記法が異なる場合もある
- 文字列処理に必須の機能
- リファレンスマニュアル 4.2 節

正規表現の記法

- “.”(ピリオド): 任意の一文字
  - ただし、改行文字とは一致しない
- “*”: 直前の正規表現の 0 回以上の繰り返し
  - "ACCT.*T"は"ACCT"で始まり"T"で終わる文字列全部
    注意："ACCTT"も含まれる
- “+”: 直前の正規表現の 1 回以上の繰り返し
  - "ACCT.+T"は"ACCT"で始まり１つ以上文字があり"T"で終わる文字列全部
    "ACCTT"は含まれない

正規表現の使い方

```
import re# re モジュール
pat = re.compile("正規表現") # パターン作成
m = pat.search("文字列") # パターンから検索
# pat は変数名なので好きにつけてよい
```

- search はマッチオブジェクトを返す

  与えた文字列に正規表現に一致する部分があれば真、
  そうでなければ偽となる

正規表現の使い方

```
import re# re モジュール
pat = re.compile("正規表現") # パターン作成
m = pat.search("文字列") # パターンから検索
# pat は変数名なので好きにつけてよい
```

- search はマッチオブジェクトを返す

  与えた文字列に正規表現に一致する部分があれば真、そうでなければ偽となる

```
p = re.compile("ACCT.+T")
for line in lines:
    m = p.search(line)
    if m: # 一致したら出力
        print line
```

実習

- 授業の Web ページから “comment.py” をダウンロードし、プログラムを完成させなさい
  - このプログラムは前回課題と同様、FASTA 形式のファイルを読み込み、“>” で始まる行を無視し、一連の遺伝子部分の文字列を 1 行にまとめて出力する
  - “HERE” とコメントしているところを埋めなさい

正規表現のオプション

- compile にオプションを指定可能
  - re.I →大文字小文字の違いを無視
  - re.S →“.”(ピリオド) が改行にも一致する
- 指定の仕方

```
re.compile('パターン', re.I) # 単数
re.compile('パターン', re.I|re.S)
# 複数は“|”で区切る
```

正規表現：その他の記法

- 正規表現 A と B に対し “A|B” で A または B のどちらかに一致する
- “^” で文字列の先頭
  - “^>gi” で “>gi” が先頭にある時に一致
- “$” で文字列の最後
- “[]” で複数文字からの選択
  - [abc] で abc のうちどれか 1 文字と一致
- “[^]” で複数文字以外からの選択
  - [^abc] で abc 以外の 1 文字と一致

正規表現：特殊記号

- \d(=[0-9]) →数字と一致
  - “\d+” で数字のみからなる文字列と一致
- \D(=[^0-9]) →数字以外と一致
- \s →空白文字 (改行やタブ、スペースなど)
- \S →非空白文字 (改行やタブ、スペースなど)
- \b →単語の直前か直後の空白文字
  - “\b[a-z]+\b” で、単語がすべて小文字アルファベットで書かれたものと一致
- \B →それ以外
- \\→バックスラッシュ

実習

- さきほどの “comment.py” を改良し、コメント部分のパターンを「行の先頭で始まる」ものに限定しなさい

正規表現による置換

- sub メソッドを用いる
  - replace による置換は固定した文字列を固定した文字列で置換
  - sub では正規表現に一致する文字列すべてを固定した文字列で置換

正規表現による置換

- sub メソッドを用いる
  - replace による置換は固定した文字列を固定した文字列で置換
  - sub では正規表現に一致する文字列すべてを固定した文字列で置換

```
p = re.compile("正規表現") # パターン作成
new = p.sub("文字列")
# パターンに一致する部分を文字列で置換
```

- 例：“[Uu]nix” を “UNIX” へ置換

正規表現による部分文字列抽出

- 抜きだしたい部分を"("と")"でくくる

```
pat = compile("<td>(.+):</td>")
```

  - “<td>” と “:</td>” に囲まれた部分を抽出

- マッチオブジェクトの group 関数で抽出

```
m = pat.search(line)

ex = m.group(1)
```

  - "("と")"は複数使ってよくて、group(1) の “1” は１番目の括弧という意味

実習(今日の課題)

- “NC_000913.faa” をダウンロードしなさい

  このファイルのメタ情報部分 (“>gi” で始まる行) は

  `>gi|16127996|ref|....`

  となっている。

- “extract.py” を改良して、この “16127996” 部分を抜きだすプログラムを作りなさい
  - “|” は正規表現では特別な意味を持つので “\|” としないとけない
  - `re.compile('>gi|.*|')` では不十分
    - “.*” は「強欲」で、できるだけ長い文字列と一致しようとするので最初と最後の | に囲まれた `16127996|ref|...` が抽出される

第3回レポート課題

- 締切 9/30
- 問題
  - 入力: FASTA 形式のファイル名と整数 n
  - 出力: “>” で始まる行の次の行から、次の “>” で始まる行の前の行までを 1 つの領域としたとき、すべての領域に表われる長さ n の部分文字列のうちで頻度の高いほう上位 10 個を出力する
- サンプルファイルを “DataSample1” と “DataSample2” という名前で授業の Web ページに掲載します。
  - “DataSample1” は、擬似的な小規模データで、こちらで動作の確認をする
  - “DataSample2” は実際のデータで、これを入力とした場合の出力例を添える
- データファイルに大文字や小文字が混在していても問題なく動くようにしなさい