電子計算機入門

池田, 大輔
九州大学附属図書館

電子計算機入門
第10回(07/14)

池田 大輔

z4id01in@cse.ec.kyushu-u.ac.jp

附属図書館

目次

- 前回課題の回答例
- 正規表現
  - 正規表現による置換
  - 正規表現による部分文字列の抽出
- 付録：第２回課題解答例

前回課題の解答例

- **入力：FASTA 形式のファイル**
- **出力：各遺伝子 (^>gi の次の行から次の^>gi の前の行まで) を 1 本の文字列として出力するプログラムを作成せよ。**

```
seq = [] # 1 つの遺伝子を保存
genes = [] # 全遺伝子を保存
------------------.
'>gi|161...'        # seq=[]
ATGAAACGCT          # seq.append()
'>gi|161...'        # join&genes.append()&seq=[]
AACGCATTAG          # seq.append()
AACGCATTAG          # seq.append()
ATAACATTAG          # seq.append()
```

前回課題の解答例(Cont.)

```
import sys, re
f = open(sys.argv[1], 'r')
all = f.readlines()
pat = re.compile("^>gi") # パタン生成
seq = []
genes = []
for line in all: # 各行に対し
    m = pat.search(line) # 検索
    if m:
        gene = ''.join(seq)
        genes.append(gene)
        seq=[]
    else:
        seq.append(line)
for line in all:
    print "(", line, ")"
```

出力例

入力

```
>gi
ATGA
>gi
ATGCGA
TGGAAA
GGTGGC
>gi
```

出力

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

正規表現による置換

- `sub()` を用いる
  - 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() を用いる
  - 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

```
pat = re.compile("正規表現") # パターン作成
new = pat.sub("置換文字列", "対象文字列")
# 対象文字列の中のパターンに一致する部分を
# 置換文字列で置換
```

- 例：“[Uu]nix” を “UNIX” へ置換

実習：正規表現による置換

- 以下のプログラムを作成し、実行結果を確認せよ。

```
-----------------.
str = "unixUnixUNIX" # 文字列
pat = re.compile("[Uu]nix") # パターン
new = pat.sub("UNIX", str)
print new
str = “unixUnixUNIX” # 文字列
new = pat.sub("", str) # 削除
print new
-----------------.
```

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：特殊文字の置換

- FASTA 形式のファイルを読み込み、一つの遺伝子を一行にして、各遺伝子を出力せよ。

―――――――――――――――-

```
# ここより上は前回課題と同様
pat2 = re.compile("\s")
for line in all:
    line = pat2.sub("", line) # 削除
    m = pat.search(line)
    if m:
        gene = ''.join(seq)
  # ここより下も前回課題と同様
```

正規表現による部分文字列抽出

- FASTA 形式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。
- このメタ情報のみ抜きだすには？

正規表現による部分文字列抽出

- FASTA 形式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。
- このメタ情報のみ抜きだすには？

  →正規表現で抜きだしたい部分の前後を含めて表現し、抜きだしたい部分を”(“と”)” でくくる
- `re.compile('>gi\|(.*)\|')`
  - “|” は正規表現で特別な意味を持つので、前に “\” が必要
- マッチオブジェクトの `group` 関数で抽出

```
m = pat.search(line)

print= m.group(1)
```

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

実習：メタ情報の抽出

- FASTA 形式のファイルのメタ情報を出力するプログラムをかけ

```
import sys, re
f = open(sys.argv[1], 'r')
all = f.readlines()
pat = re.compile("^>gi\|(.*)\|") # パタン
for line in all:
    m = pat.search(line) # 検索
    if m: # パタンに一致した
        print m.group(1)
```

実習：メタ情報の抽出(今日の課題)

- FASTA 形式のファイルのメタ情報のうち “16127996” 部分のみを出力するプログラムをかけ
- さきほどの

  `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

  → `re.compile('>gi\|([^|]*)\|')`

  `[^abc]` は “abc” 以外の文字と一致

第3回レポート課題

- 締切 7/30(金)
- 問題
  - 入力: FASTA 形式のファイル名と正規表現 (モチーフ)
  - 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、これらの領域における与えられたモチーフの出現回数と、そのモチーフが最初に出現した遺伝子のメタ情報
    - メタ情報とは、以下における “16127996” の部分のことである

      `>gi|16127996|ref|....`
- サンプル “DataSample1”(小) と “DataSample2”(大)

付録：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
```

付録：第2回課題回答例

```
import sys
f = open(sys.argv[1], 'r')
n = int(sys.argv[2])
lines = f.readlines()
count=
for line in lines:
    for i in range(len(line)):
        if i + n >= len(line):
            continue
        substr = line[i:i+n]
        if count.has‘5Fkey(v)substr:
            count[substr] += 1
        else:
            count[substr] = 1
```

付録：第2回課題回答例(Cont.)

```
array = [(count[key], key) for key in count.keys()]
array.sort()
array.reverse()
for i in range(10):
      print array[i]
```