Recitation 2 - Notes

Matt Rasmussen

September 14, 2006

1 Python

Below is a very short introduction to some Python syntax, specifically some things you may need for the homework assignment. Much more information is available at http://www.python.org. Our template is tested to work with the latest stable version is python 2.4. (python 2.5 is available but is still beta).

Python features include:
1. Object oriented programming
2. Exception handling
3. Automatic garbage collecting
4. A large standard library
5. An interpreter
6. Runs on several operating systems (MacOS, Windows, Unix even OS/2)
7. And much more...

Main differences of python from other languages:
1. White space matters! Indentation is used to delineate blocks
2. You do not need to have semicolons but you cannot break up lines unless you use a backslash:
   ```python
   a = 4 + 5 + 2 + \\
   4 + 5
   ```
3. Variables do not need to be declared. Their declaration is implied with the first assignment
4. Types are dynamic. You can just about anything at run time.
5. Nearly everything is an object (Except integers and floats). Functions are first class values. Modules are objects. Classes and types are objects.
6. One thing that you may find handy are the dir() and help() functions. dir() (whose name is inspired from the directory listing command) will list the attributes of any object.
   ```python
   x = [1, 2]
dir(x)
   ```
   The above will show what member functions are available in the list object.
   ```python
   import sys
dir(sys)
   ```
   The above imports the 'sys' (system) module and uses dir() to list all the functions available in this module.
   ```python
   import sys
   help(sys.exit)
   ```

1

1
The above will show help information about the function sys.exit. help() will give help information about any python object.

7. Control structures must have a colon:

```
if not 5 > x:
    x = 5
while x > 5:
    x = x - 1
# prints numbers from 0 to 9
for x in range(10):
    print x
```

8. A zero list (array) of size x can be made with the following command: [0]*x

9. Strings, which are basically lists that cannot be changed, can be indexed like in C (but with ranges) or concatenated like in perl:

```
x = "a"*3  # z = "aaa"
x = x + "bcd"  # z = "aabcd"
x = x[3:5]   # z = "bc"
```

10. Functions have the following syntax:

```
def foo(a,b):
    return a + b + 5
```

11. Tuples are list lists but they cannot be changed once they have been created but they can contain different datatypes.

```
t = ((1,4,2),324)
```

12. Dictionaries allow for a mapping from an immutable (unchangeable) value to some value.

```
a = {'a': 4, 'b': "three", 3:4, (3,7): "three"}
print a[(3,7)]  # "three"
```

Excerpt from python tutorial.

# 4. More Control Flow Tools

* 4.1 if Statements
* 4.2 for Statements
* 4.3 The range() Function
* 4.4 break and continue Statements, and else Clauses on Loops
* 4.5 pass Statements
* 4.6 Defining Functions
* 4.7 More on Defining Functions
  o 4.7.1 Default Argument Values
  o 4.7.2 Keyword Arguments
  o 4.7.3 Arbitrary Argument Lists
  o 4.7.4 Unpacking Argument Lists
  o 4.7.5 Lambda Forms
2 Substitution Matrices

Typically, substitution matrices are developed by manually aligning very similar sequences and then extrapolating to obtain the probability of certain amino acids being replaced by other ones or being maintained.

To develop a score for an alignment we typically use a log odds ratio. We consider two models: the unrelated $U$ and related $R$. For the unrelated model, we assume that the nucleotides in the two sequences are independent of each other. Thus, $P(a, b|U) = P(a|U)P(b|U)$. For the related model we have $P(a, b|R)$. Both these values are derived from frequency counts. If for a pair of sequences $A$ and $B$, $P(A, B|R) \gg P(A, B|U)$, then we have a high confidence that $A$ and $B$ are related.

The log ratio of the single nucleotide probabilities gives us $s(a, b) = \log \frac{P(a, b|R)}{P(a|U)P(b|U)}$. Because we assume that the substitutions are independent, the score of an alignment of $A$ and $B$ leads to:

$$S(A, B) = \sum s(A_i, B_i) = \sum \log \frac{P(A_i, B_i|R)}{P(A_i|U)P(B_i|U)}$$

$$= \log \prod \frac{P(A_i, B_i|R)}{P(A_i|U)P(B_i|U)}$$

$$= \log \frac{P(A, B|R)}{P(A|U)P(B|U)}$$

which should be large when two sequences are related and small when they are unrelated.

Note: this does not account for the score of gaps. The gap penalty is often derived in a heuristic manner.

The value along the diagonal of a substitution matrix measures to what extent the alignment of an amino acid with itself increases our confidence that two sequences are related. Thus, we expect variance amongst these values because certain amino acids are very common and finding them aligned to does not significantly increase how sure we are that two sequences are related.
2.1 BLOSUM (BLOcks SUbstition Matrix) for amino acids

|     | A    | C    | D    | E    | F    | G    | H    | I    | K    | L    | M    | N    | P    | Q    | R    | S    | T    | V    | W    | Y    |
|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| A   | 4    | 0    | -2   | -1   | -2   | 0    | -2   | -1   | -1   | -1   | -2   | -1   | -1   | -3   | -1   | 1    | 0    | 0    | -3   | -2   |
| C   | 9    | -3   | -4   | -2   | -3   | -3   | -1   | -1   | -1   | -3   | -3   | -3   | -3   | -1   | -1   | -1   | -1   | -1   | -1   | -2   | -2   |
| D   | 6    | 2    | -3   | -1   | -3   | -1   | -4   | -3   | 1    | -1   | 0    | -2   | 0    | -1   | -3   | -4   | -3   | -2   | -3   | -2   |
| E   | 5    | -3   | -2   | 0    | -3   | 1    | -3   | -2   | 0    | -1   | 2    | 0    | 0    | -1   | -2   | -3   | -2   | -2   | -3   | -2   |
| F   | 6    | -3   | -1   | 0    | -3   | 0    | 0    | -3   | 0    | -4   | -3   | -3   | -2   | -2   | -1   | 1    | 3    | -2   | -3   |
| G   | 6    | -2   | -4   | -2   | -4   | -3   | 0    | -2   | -2   | -2   | 0    | -2   | -3   | -2   | -3   | -2   | 2    | -2   | -3   | -2   |
| H   | 8    | -3   | -1   | -3   | -3   | -1   | -2   | 0    | 0    | -1   | -2   | -3   | -2   | 2    |
| I   | 4    | -3   | 2    | 1    | -3   | -3   | -3   | -2   | -1   | 3    | -3   | -1   |
| K   | 5    | -2   | -1   | 0    | -1   | 1    | 2    | 0    | -1   | -2   | -3   | -2   |
| L   | 4    | 2    | -3   | -3   | -2   | -2   | -2   | -1   | 1    | -2   | -1   |
| M   | 5    | -2   | -2   | 0    | -1   | -1   | -1   | 1    | -1   | 1    | -1   |
| N   | 6    | -2   | 0    | 0    | 1    | 0    | -3   | -4   | -2   | 5    | 1    | 0    | -1   | -2   | -2   |
| P   | 7    | -1   | -2   | -1   | -1   | -1   | -2   | -4   | -3   | 5    | 1    | -1   | -3   | -3   |
| Q   | 5    | -1   | -1   | -3   | -3   | -2   | 4    | -3   |
| R   | 5    | -1   | -1   | -3   | -3   | -2   | 4    | -3   |
| S   | 4    | 1    | -2   | -3   | -2   | 5    | 0    | -2   |
| T   | 5    | 0    | -2   | -2   |
| V   | 4    | -3   | -1   |
| W   | 11   | 2    |
| Y   | 7    |      |      |

Table 2 - The log odds matrix for BLOSUM 62