Python Lab

Functions

Proteomics Informatics, Spring 2014

Week 6

04th Mar, 2014

Himanshu.Grover@nyumc.org

From last week…

- **Compute Peptide Fragment Ions**

- **Problem:** Given a peptide sequence and charge state, compute m/z of various b- & y-ions

Functions: Concept

- Named group of a set of statements
- **May** receive inputs that differ each time the function is executed
- Perform some computation
- **May** return some result back

Examples

- **Built-in** functions

  x = 10; range**(**x**)**

- Functions from **other libraries/modules**:

  import math; **math.pow(**3,2**)**

- Functions associated with **other objects** (actually called *methods*):

  x = [1, 3, 8, 7, 5]; **x.sort()**

- User-defined

General Syntax

- When **defining** it specify *name* and *statements*

def **funcName (**args/params/inputs**)**:

statement$_1$

statement$_2$

…

return <something>

- Use indentation to demarcate a function from the rest of the code

Flow of Execution

- Once python interpreter has executed the “def” statement, you can **“call”** the services of the function anywhere in your script, and as many times as needed
  - **‘call’** it using the name
  - followed by round braces with comma-separated inputs, just like in math, f(x):
    - **funcName(**args**)**
- **Must be defined before calling it**
- Detour in the program’s flow of execution (**Check using ipdb**)

Why write functions?

- Modularize and organize your code, by meaningfully dividing a long program into smaller pieces (steps of a process/procedure)
  - Easier to:
    - Build
    - Debug
    - Read (if using meaningful and self-documenting labels/function-names)
- Facilitates code reuse
  - remove redundant code
    - run more than once, at different time points
    - share across the same or different projects
  - easier to modify

Function Design Guidelines

- **Size**
  - keep them short
- **Cohesion**
  - each function should have a single, unified purpose
- **Coupling**
  - each function should be independent of things outside of it. Use ***arguments*** for inputs and ***return*** for outputs

HW Assignment

- Add remaining neutral losses
  - b-nh3
  - y-h2o
  - y-nh3
- Add remaining multiply-charged fragments
  - y+
- Write functions:
  - to compute neutral losses
  - to compute higher charged fragments

Next Class

- More about functions
- Reading/Writing files
- Examples