Genome 559: Introduction to Statistical and Computational Genomics

Winter 2009

Lecture 20b

Exceptions

Larry Ruzzo

1 Minute responses

Today's freeform exercise was helpful in that it generated more questions to explore than it answered - things to explore on my own time. I'll need to spend time thinking about LD because as a concept it isn't clicking just yet - again probably just requires individual investigation.

Great class today. Was finally able to complete HW6, thanks for walking me through the loops. Look forward to playing with Biopython once I get it up and running.

LD seems like a useful method and pretty simple too.

Today's lecture was very understandable. I'm just worried about the homework at this point. It's still very confusing to me.

Python “Exceptions”

A moderately “advanced” feature, but worth seeing because you’ll see it in other people’s programs and/or packages you use

Computers are frustratingly literal; odd cases that wouldn’t bother you cause errors

In your program, you can test for/handle all sorts of oddities. But more difficult if A calls B calls C calls D, which detects some error; what to do? Print a message? Return a special value? Just quit?

Exceptions: (semi) uniform error handling

If you detect an error, you “raise” or “throw” and “exception”.

Caller may (or may not) choose to write special code to handle it – e.g., print a message or quit or maybe go on via some default or fall-back.

Try This, with argv = 1 and 0

Which “Exiting” messages do/don’t appear?

```
import sys
def A(x):
    print "Entering A with", x
    try:
        result = B(2*x)
    except ZeroDivisionError:
        print ”..ZDiv Error."
        print "..Returning  default"
        result = 42
    print "Exiting A with", result
    return result
```

```
def B(x):
    print "  Entering B with", x
    result = C(2*x)
    print "  Exiting B with", result
    return result
def C(x):
    print "    Entering C with", x
    result = 1/x
    print "    Exiting C with", result
    return result

print ”the answer is..."
answer = A(int(sys.argv[1]))
print  answer
```

More on exceptions

Exceptions are named

Exception handler can, probably should, say which ones it wants to handle.

If you don’t handle it, it’s passed “up” to your caller

A “default” handler is provided at the top level to handle all the rest (prints the error messages you know and love).