A Warm Welcome

"Practical python programming for Big Data and the scientist"

Lucas Sinclair
Alexander Eiler
Ludovic Dutoit

The Teachers

Lucas
Sinclair
PhD

Alexander
Eiler
Docent

Ludovic
Dutoit
PhD student

The Students

1. Your name
2. Your affiliation
3. What you wanna use python for

Lucas Sinclair

I work as an external for Uppsala University

I want to recompose microbial genomes from metagenomic data.

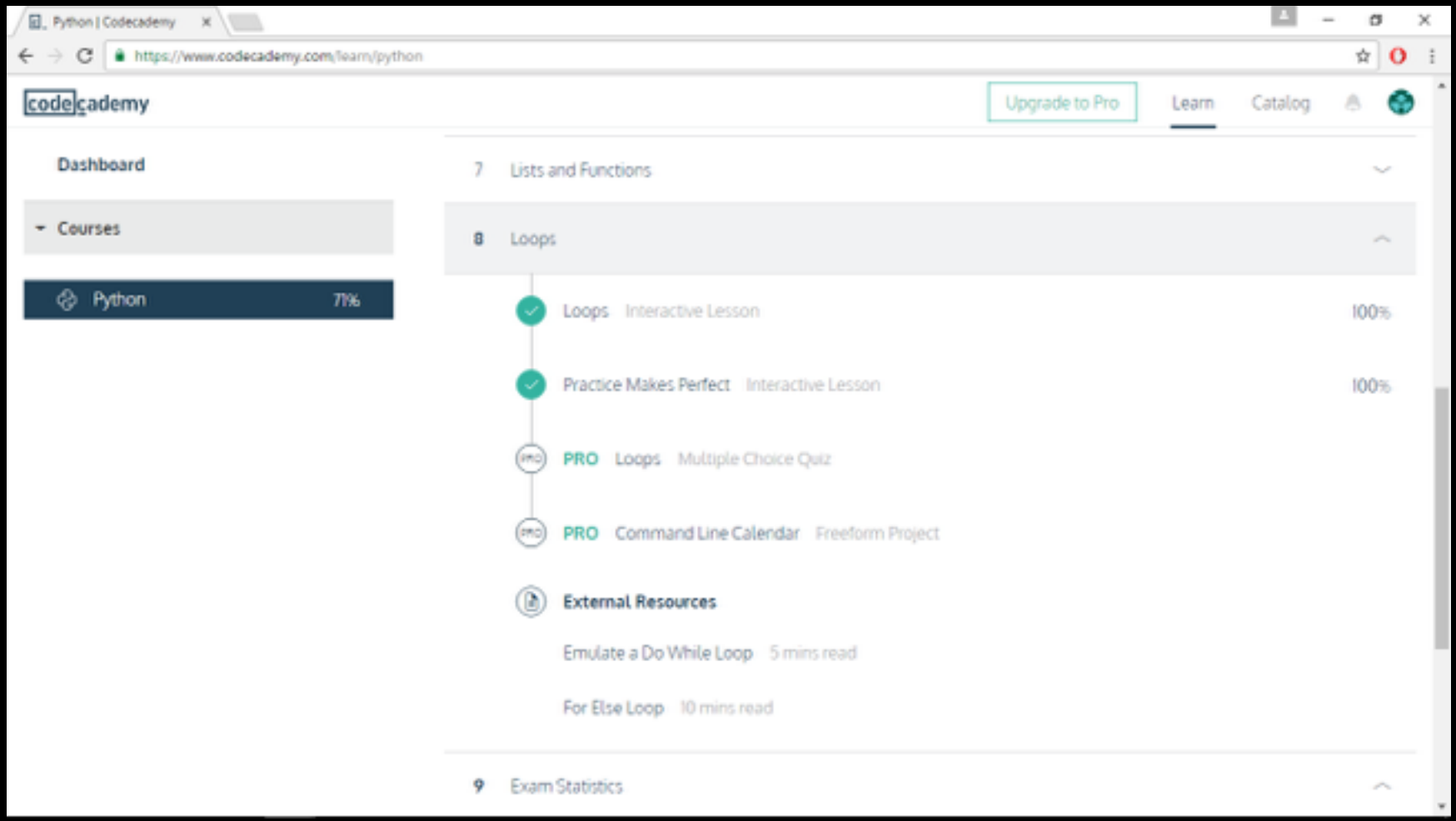
Diversity

Department of Cell and Molecular Biology	6
Department of Earth Sciences	1
Department of Ecology and Genetics	10
Department of Engineering Sciences	1
Department of Information Technology	1
Department of Organismal Biology	4
Department of Physics and Astronomy	7

Course goals

1. Process, filter, clean, analyze, and visualize scientific data.
2. Automate the above.
3. Learn object-oriented programming and other best practices.
4. Learn a few extra "ecosystem" tools.

Pre work attendance



Attendance sheet

1. We will pass it around.
2. Your responsibility to get your name on it once per day.
3. Required to get the credits.

Your own computer

1. For better reproducibility and subsequent skill transfer.

Schedule

Week 12	Monday	Tuesday	Wednesday	Thursday	Friday
	Afternoon	Afternoon	Afternoon	no course	no course

Week 13	Monday	Tuesday	Wednesday	Thursday	Friday
	Afternoon	Afternoon	Afternoon	no course	no course

Repo

http://tinyurl.com/jcat3ms

Any Questions ?

Why programming

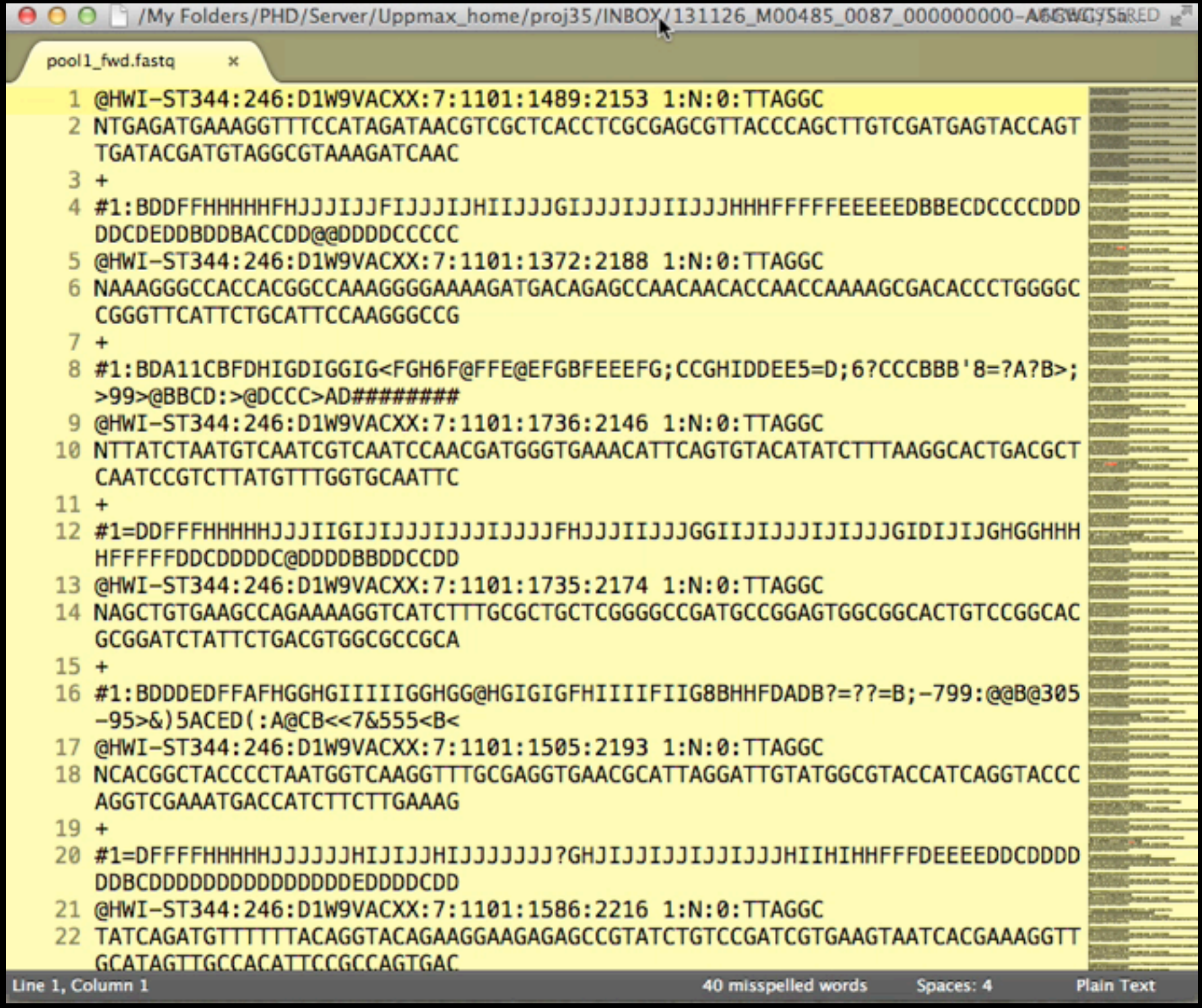
1. I can just do that in Excel !

Big Data

1. Big data is a term for data sets that are so large or complex that traditional data processing applications are inadequate to deal with them.

Small data

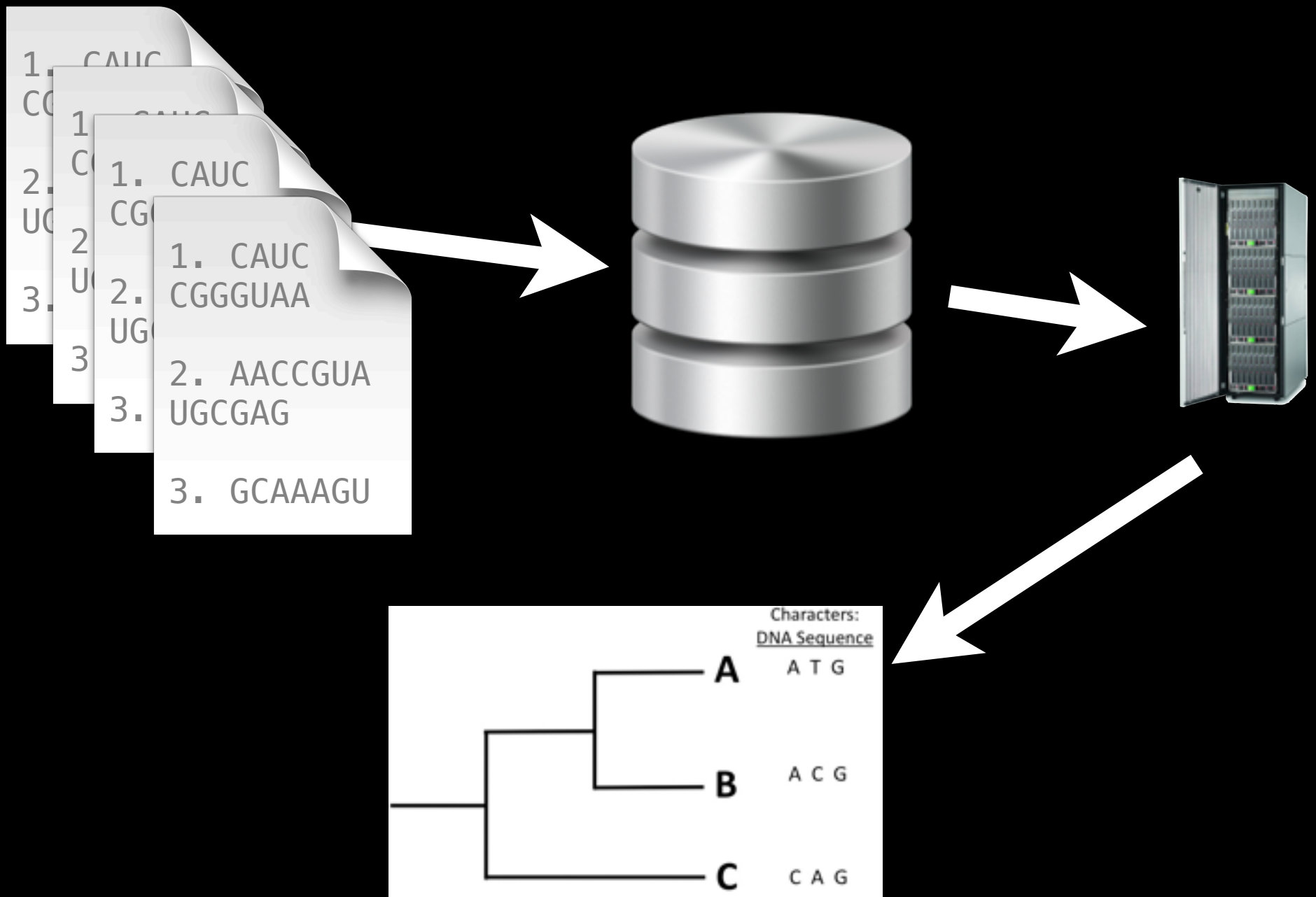
	A	B	C	D
1	Density	Parasitized larave	Total number of larvae	Habitat Type
2	3	7	14	A
3	5	10	22	A
4	11	9	21	A
5	12	8	17	A
6	22	6	10	A
7	57	5	11	A
8	2	3	10	B
9	7	2	5	B
10	17	15	31	B
11	23	17	20	B
12	29	9	11	B
13	33	20	22	B
14	4	8	17	C
15	7	10	22	C
16	7	7	15	C
17	10	6	9	C
18	12	22	43	C
19	14	5	11	C


/My Folders/PHD/Server/Uppmax_home/proj35/INBOX/131126_M00485_0087_000000000-A6GWG/Sa
pool1_fwd.fastq
@HWI-ST344:246:D1W9VACXX:7:1101:1489:2153 1:N:0:TTAGGC
NTGAGATGAAAGGTTTCCATAGATAACGTCGCTCACCTCGCGAGCGTTACCCAGCTTGTCGATGAGTACCAGT
TGATACGATGTAGGCGTAAAGATCAAC
+
#1:BDDFFHHHHHFHJJJIJJFIJJJIJHIIJJJGIJJJIJJIIJJJHHHFFFFFEEEEEDBBECDCCCCDDD
DDCDEDDBDDBACCDD@@DDDDCCCCC
@HWI-ST344:246:D1W9VACXX:7:1101:1372:2188 1:N:0:TTAGGC
NAAAGGGCCACCACGGCCAAAGGGGAAAAGATGACAGAGCCAACAACACCAACCAAAAGCGACACCCTGGGGC
CGGGTTCATTCTGCATTCCAAGGGCCG
+
#1:BDA11CBFDHIGDIGGIG<FGH6F@FFE@EFGBFEEEFG;CCGHIDDEE5=D;6?CCCBBB'8=?A?B>;
>99>@BBCD:>@DCCC>AD########
@HWI-ST344:246:D1W9VACXX:7:1101:1736:2146 1:N:0:TTAGGC
NTTATCTAATGTCAATCGTCAATCCAACGATGGGTGAAACATTCAGTGTACATATCTTTAAGGCACTGACGCT
CAATCCGTCTTATGTTTGGTGCAATTC
+
#1=DDFFFHHHHHJJJIIGIJIJJJIJJJIJJJJFHJJJIIJJJGGIIJIJJJIJIJJJGIDIJIJGHGGHHH
HFFFFFDDCDDDDC@DDDDBBDDCCDD
@HWI-ST344:246:D1W9VACXX:7:1101:1735:2174 1:N:0:TTAGGC
NAGCTGTGAAGCCAGAAAAGGTCATCTTTGCGCTGCTCGGGGCCGATGCCGGAGTGGCGGCACTGTCCGGCAC
GCGGATCTATTCTGACGTGGCGCCGCA
+
#1:BDDDEDFFAFHGGHGIIIIIGGHGG@HGIGIGFHIIIIFIIG8BHHFDADB?=??=B;-799:@@B@305
-95>&)5ACED(:A@CB<<7&555<B<
@HWI-ST344:246:D1W9VACXX:7:1101:1505:2193 1:N:0:TTAGGC
NCACGGCTACCCCTAATGGTCAAGGTTTGCGAGGTGAACGCATTAGGATTGTATGGCGTACCATCAGGTACCC
AGGTCGAAATGACCATCTTCTTGAAAG
+
#1=DFFFFHHHHHJJJJJJHIJIJJHIJJJJJJJ?GHJIJJIJJIJJIJJJHIIHIHHFFFDEEEEDDCDDDD
DDBCDDDDDDDDDDDDDDEDDDDCDD
@HWI-ST344:246:D1W9VACXX:7:1101:1586:2216 1:N:0:TTAGGC
TATCAGATGTTTTTTACAGGTACAGAAGGAAGAGAGCCGTATCTGTCCGATCGTGAAGTAATCACGAAAGGTT
GCATAGTTGCCACATTCCGCCAGTGAC
Line 1, Column 1
40 misspelled words
Spaces: 4
Plain Text

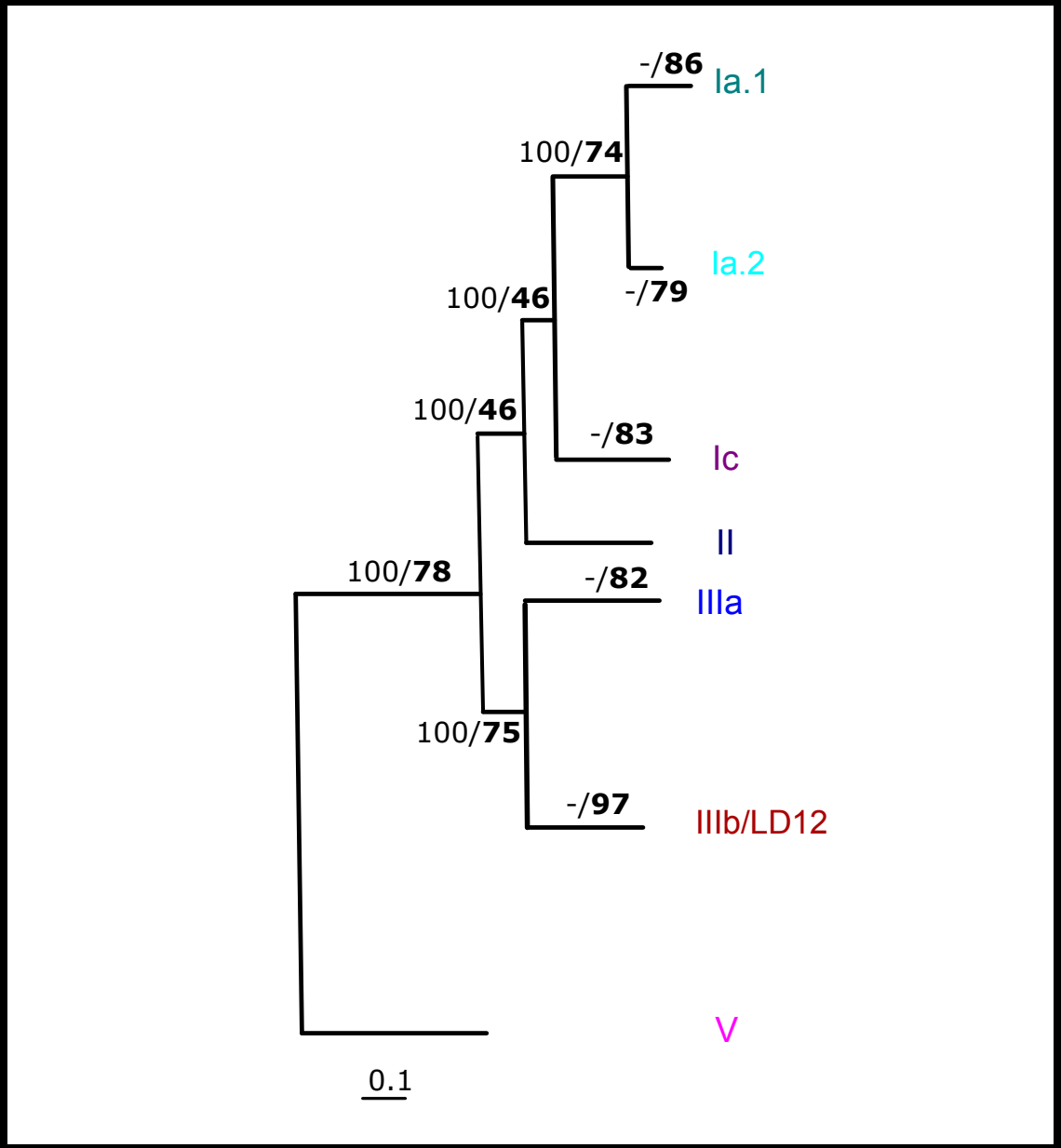
Why programming

1. Too many steps to carry them out by hand with a GUI.

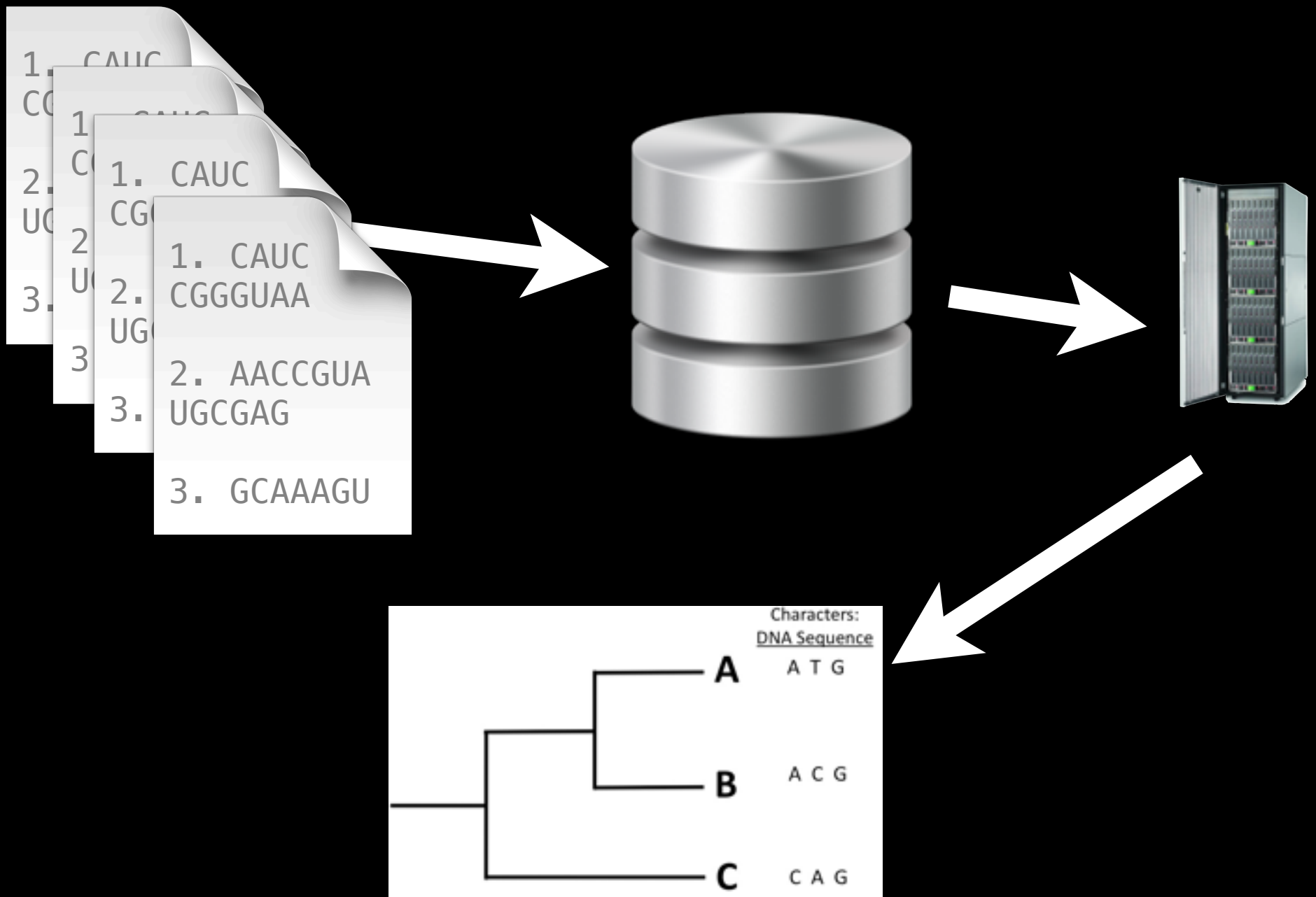
Example project



Result


-/86
Ia.1
100/74
Ia.2
-/79
100/46
100/46
-/83
Ic
II
100/78
-/82
IIIa
100/75
-/97
IIIb/LD12
V
0.1

Example project



Why Python?

Java

C

Perl

Bash

Matlab

R

Ruby

Julia

Why Python

1. Modern
2. High-level
3. Widely used
4. Considered good for learning.

Oct 2016	Oct 2015	Change	Programming Language	Ratings
1	1		Java	18.799%
2	2		C	9.835%
3	3		C++	5.797%
4	4		C#	4.367%
5	5		Python	3.775%
6	8	⌃	JavaScript	2.751%
7	6	⌄	PHP	2.741%
8	7	⌄	Visual Basic .NET	2.660%
9	9		Perl	2.495%
10	14	⌃⌃	Objective-C	2.263%
11	12	⌃	Assembly language	2.232%
12	15	⌃	Swift	2.004%
13	10	⌄	Ruby	2.001%
14	13	⌄	Visual Basic	1.987%
15	11	⌄⌄	Delphi/Object Pascal	1.875%
16	65	⌃⌃	Go	1.809%

Why not Python

1. Time to learn new language is huge.
2. Pick one (or two) that are more or less well suited.

Python 2 and Python 3

A. We will teach Python 2 only here.

B. Python 3 is backwards incompatible with 2.

C. Python 3 is not as widely adopted in science.

Backwards incompatible

In python 2:

```
1 print "works in python 2"
2 print("also works in python 2")
3
4
```

In python 3:

```
1 print("only way to print in python 3")
2
3
4
```

Origin of Python

1. Made by a guy called "Guido Van Rossum".
2. The name was chosen in reference to the Monty Python movies.
3. Conceived in the late 1980s

Imperative

```
1 first instruction
2 second instruction
3 third instruction
4
5
6
7
8
9
```

Duck typed

1. If it walks like a duck, if it quacks like a duck, let's just assume it is a duck.
2. No type checking, but stop execution at runtime if we have a problem.

Duck typed

In C:

```
1 int x;
2 x = 4;
3
4
```

In python:

```
1 x = 4
2
3
4
```

High level

1. Deal with abstract concepts directly.
2. Ignore hardware and operating system.

Basic built-in types

1. int
2. float
3. str
4. list

Any Questions ?

Exercise

1. Output the string:
   "Hello World".

Solution

```
1  print "Hello World"
2
3
4
5
6
7
8
9
10
11
12
13
```

Execute a file

```
execfile("exercice1_day1.py")



execfile("exercices/01_day1.py")



execfile("../exercices/01_day1.py")
```

Exercise

1. Define a list with four items.
2. Print this list on the screen.
3. Change the second item of the list to "bob".
4. Print this list on the screen again.

Solution

```
1  my_list = [1, 6, 23, "alice", 4]
2
3  print my_list
4
5  my_list[1] = "bob"
6
7  print my_list
8
9
10
11
12
13
```

Exercise

1. Define a function.
2. The function is named "add".
3. It takes two input arguments: "x" and "y".
4. It returns the summation of these two numbers.

Solution

```
1  def add(x, y):
2      print x+y
3
4
5  add(3, 5)
6
7
8
9
10
11
12
13
```

Exercise

1. Define a function.
2. The function is named "check_int".
3. It takes one argument "x".
4. It returns True if x is an integer (type is int). Otherwise, False.

Solution

```
1  def check_int(x):
2      return isinstance(x, int)
3
4
5  check_int(3)
6  check_int("Hello")
7  check_int(False)
8  check_int([1,2,3])
9
10
11
12
13
```

15 minutes break

Exercise

1. Define a function.
2. The function is named "get_time".
3. It takes no input arguments.
4. It returns the time of the day. For instance "15:23:32" as a string.

Solution

```
1  import time
2
3  def get_time():
4      return time.strftime("%H:%M:%S")
5
6
7  get_time()
8
9
10
11
12
13
```