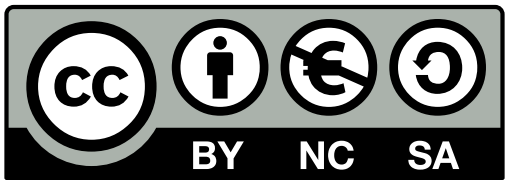
Herramientas de Programación en Bioquímica y Biología Molecular: Introduction to the course

Ramón Díaz-Uriarte

13-September-2016

(Rev: f1fdcb0)

License and copyright



Inglés o español?

- Las diapos y el material docente estarán en inglés.
- Los exámenes estarán en inglés.
- (La gran mayoría de libros, manuales, y docs en programación están en inglés)
- Pero las clases serán en español (con el suficiente spanglish para usar los términos técnicos).

Outline

Objectives

- It is all in the “Guia docente”. Lets summarize.
- Teach computational skills useful to a biochemist/molecular biologist/biologist.

Objectives (II)

- Use programming languages to help you deal with your data (manage, analyze, plot).
- Write programs to automate repetitive tasks.

Objectives (III)

- “Making scientists more productive by teaching them basic computing skills.” (From Software Carpentry)
- (And not just the scientist)

Objectives (IV)

Now, really, how?

- Languages: Python, R, SQL. Why?
- Dealing with files and directories, and quickly getting info out of/from them. Implicit in a lot of what we'll do.
- All of these tools are free software (which is not mainly about money).

Contents and why

- Intro
- Programming: Python (and regexp).
- R (programming, data handling, graphics).
- Data bases and SQL.
- Odds and ends.

What will you get out of this?

- Knowledge and attitudes (conocimientos y actitudes) to solve problems using computers.
- The key is “solve problems”; the computer is the tool.
- Knowledge and attitudes that will keep you learning your way through new problems.

- This course ought to be empowering and should open many doors (things that were impossible will become easy, and things that were tedious will become fun and fast).

But there are no miracles

http://norvig.com/21-days.html

Is this for you?

- We presuppose little.
- Have you seen a computer, used a keyboard, a tablet or a smartphone? If yes to any, then you can deal with this course.

The pace of the course

- This course should go neither too fast (you should not get lost) nor too slow (you should not get bored or feel you are wasting time).
- You can provide us with continuous feedback.
- We'll ask for the time it took you to do the exercises.
- You need to be actively engaged during the classes.

Schedule

- In Moodle: “Calendario HPBBM”
- (Should be mostly right)
- Classes at Seminario IV.
- Classes start at 18:00 and end at 19:20.
- **Beware**: some exams at 11 am!!

Grading

- Again, “Guia docente”
- Final exam: 35%
- First mid-term exam: 15%
- Second mid-term exam: 15%
- Practical programming exercises: 25% (“Programming exercises”).
- Comments to practical programming exercises: 10%

Exams

- Will be in class.

Office hours

- By appointment.
- Email us.

Bibliography

- See “Guia Docente” for general resources.
- We will make specific recommendations when needed.
- Google/web searches are your friends.

Moodle

- We will be using moodle. Make sure you can access it from here. Let's do it now
- Make sure you can access from home. Try it tonight

Some of the stuff in Moodle

- A questionnaire.
- These slides.
- Some papers.
- Guia docente.
- Exercises.
- Etc.

About the class notes in Moodle

- Provided ahead of time.
- Try not to print them, as they might suffer minor changes right before class.
- ($\Rightarrow$ if you are reading these notes before class, these notes themselves can suffer changes :-))

Do you receive Moodle emails?

- The email
- Note from the “Foro”

“Software for your own computer”

- As it says: some notes about software you can install in your home computers/laptops/whatever.

Miscell. papers

- A few papers about software development in computational biology and bioinformatics.
- Might be interesting, but a lot of stuff might not be familiar now (it should at the end of the course).
- This **is not** required reading.

What you should do before, during, after class

- Material available before class.
- Material: slides and/or notes, and possibly other links or documents.
- Please, please, take a look at the material before class.
- Work with it after class.
- Regular programming exercises (▸ “Programming exercises”).
- Start with the exercises as soon as you can:
  - Avoid running out of time
  - You will learn more

How classes will be

- For each major topic, one to three (median: two) days of “lecture” and one day of “programming exercises” (▸ “Programming exercises”).
- During “lectures”: type the code to follow along if we can. (I.e., during lectures we will use the computers).
- Ask questions if something does not make sense. It is crucial you do not get lost! Adjust our speed.

“Programming exercises”

- You will have to complete a programming exercise for each major block.
- You will work in groups of three.
- Your programming exercise will be reviewed by your peers (by another group).
- We will grade:
  - Your programming exercise.
  - Your review of another groups’ exercise.

“Programming exercises”: details about mechanics (I)

BEFORE CLASS

- Each group works on the exercises on their own.
- One person from each group uploads it to Moodle before 23:30 of the day before the class (e.g., if Exercise class is on Thursday, upload to Moodle by Wednesday at 23:30).
- **Make sure you confirm that it is the final, definitive, version**.
- Teachers prepare a compressed file with ALL the exercises.

“Programming exercises”: details about mechanics (II)

DURING CLASS

- First 20’ (or 30’): peer review of work.
  - Download the compressed file with exercise.
  - Open the one your are assigned to review.
  - Review
  - Upload your reviews to Moodle. Use the standard rúbrica (“Rúbrica”).
- If needed, at most 20’: go over hardest exercises.
- (We will not necessarily devote the full 90 min to the programming exercises.)

“Rúbrica”

- Often, comments/grading generally based on a few key elements. For example:
  - Is the code correct (4 pts)
  - Ease of reading: indentation, names of variables, line length, etc (2 pts)
  - Logic: is it straightforward or convoluted (1 pt)
  - Comments: used when needed and as needed (1 pt)
  - Modular (1 pt)
- By section, you score it and you explain why.

An example rúbrica

Reading all files. Correct and clear?	No ***0 puntos***	Partly ***1 puntos***	Yes but ... ***2 puntos***	Yes ***3 puntos***	Great ***4 puntos***
Function to return length of characters of sequences. Correct and clear? (Consider also how the function is used)	No ***0 puntos***	Partly ***1 puntos***	Yes but ***2 puntos***	Yes ***3 puntos***	Great ***4 puntos***
Function to return the frequency of Ts. Correct and clear? (Consider also how the function is used)	No ***0 puntos***	Partly ***1 puntos***	Yes but ***2 puntos***	Yes ***3 puntos***	Great ***4 puntos***
Obtaining the summary results for sequences of each king (i.e., frequency of Ts and lenght of sequences for the coding and non-coding). Correct and clear?	No ***0 puntos***	Partly ***1 puntos***	Yes but ***2 puntos***	Yes ***3 puntos***	Great ***4 puntos***
Overall code organization and logic, and results	Wrong and a mess ***0 puntos***	Wrong but makes sense ***1 puntos***	Works but is a mess ***2 puntos***	Works and OK ***3 puntos***	Works and is very clear and well organized ***4 puntos***

An example exercise. THIS IS TEXT!

- This code is all executable.
- So we use comments for things that are not code.

```
### My_group:  g1  (Yo Misma, Otra Persona, Alguien Mas)

## 1.Reading all files.

x1 <- read.csv(''some.csv'')


### Printing length of chars. Define two functions first
f1 <- function(x, y) {balbalblaba}

f2 <- function(x, y, w, z) {
    some complicated code here
}
print(''hello world'')
```

An example commented exercise. THIS IS TEXT!

(This is not code that needs to be executed.)

```
My_group:  g1  (Yo Misma, Otra Persona, Alguien Mas)
Group_I_am_reviewing: g2 (Nombre1 Apellido1, ...)

@######

@1.Reading all files. Correct and clear.

2: does it correctly, but it is a mess to understand
   in lines 2 to 9. And lines 10 to 12 could be
   simplified to:

x <- readline("fichero.txt")

@######

@.Function to return length of characters of sequences.
  Correct and clear?

3: Yes, but could be improved in line 34 doing
   blablabla.
```

Rúbrica for first block?

- There is none:
- Right or wrong and why.
- There could be other exercises without rúbrica.

Groups

- Probably groups for first and second half of course will be different.
- Lets create the groups now. You choose your group mates.
- Remember: 3 per group.

The questionnaire in Moodle

- Please, answer the Moodle questionnaire.
- It really helps us to know what we know.
- There are no right or wrong answers!
- Do not feel overwhelmed by the questions: you are supposed to have no idea about many of those things!!!
- Oh, but you should have a pretty good idea about all of them by the end of the course.
- Please finish it by tonight.

Major changes compared to previous years

- No linux, grep, sed, awk, etc.
  - The mess of pendrives/linux/etc.
  - This gives us more time for R.
- So no linux/pendrive. Just use the Windows computers here or whatever you want (Win, Mac, Linux).
- Changes in Python and R (more of both, changes in order of material), reduced emphasis in editors.

Backups of your copies

- Version control will be covered later (Block III).
- For now: at least use Dropbox, Wuala, Google Drive, or similar, that allows you to roll back.
- (Why don't we have them in these machines?)

Assignments (for today!!!!)

- Answer questionnaire.
- Install software.
- Access Moodle from home.
- Upload a picture to Moodle. **Really, do it!**
- Report any problems (during the next class, in the “HPBBM Questions” Foro before next class).