
# BIO214 Lecture 12

## Bioinformatics-II

## Coding guide for bioinformatician

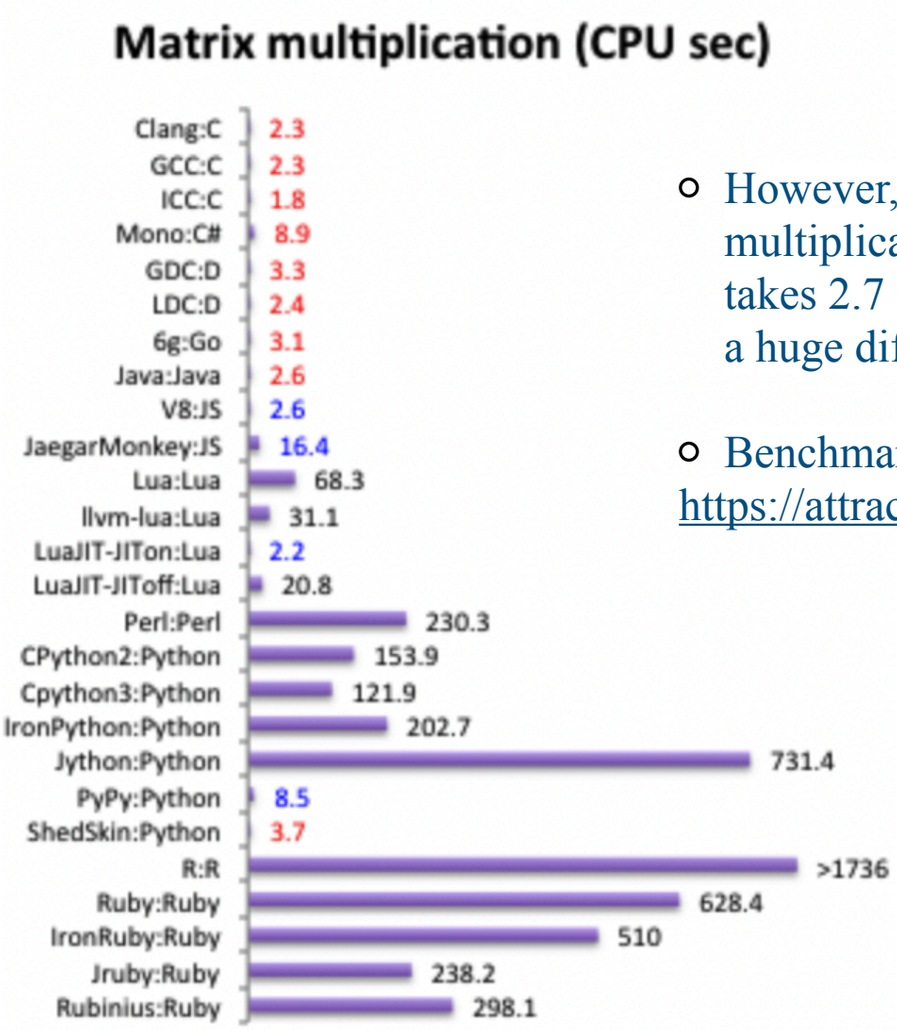
**Zhen Wei; 2023-Feb-20**

# Features of good code

- **Efficient** — Vectorization
- **Accurate** — Debug
- **Modular** — Functional programming

# Part 1. Fast code

# Is R slow?




- However, using the built-in matrix multiplication operator (%*%), R takes 2.7 sec in 57.0 MB memory, a huge difference.

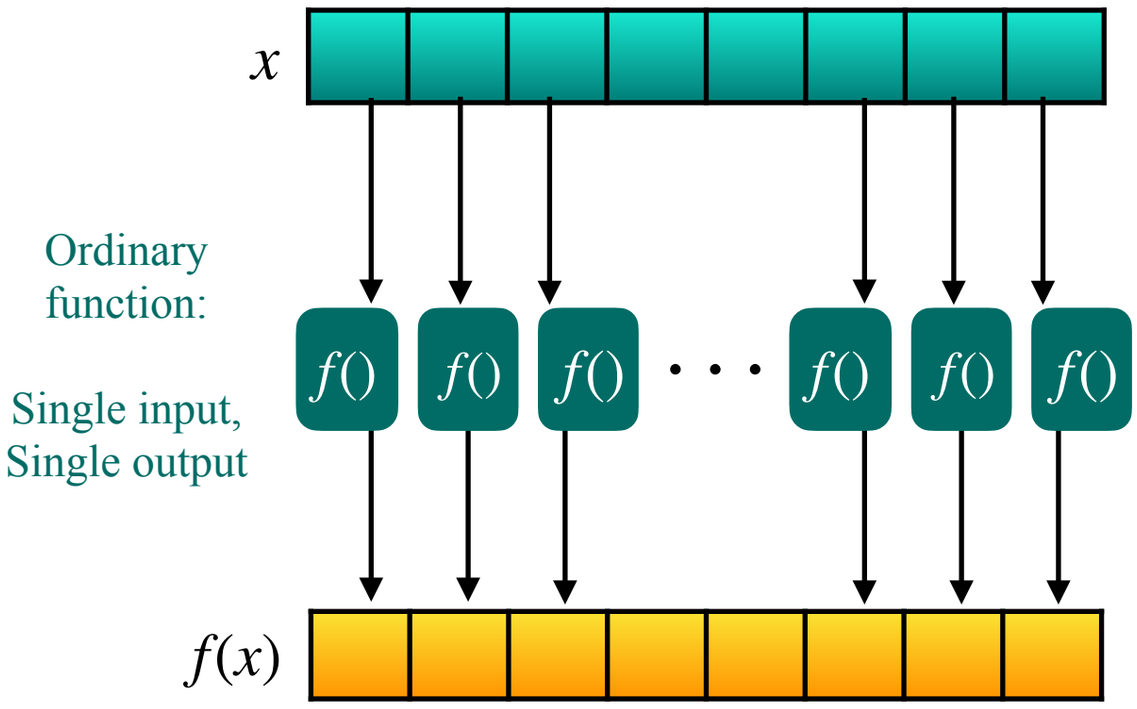
- Benchmark website: https://attractivechaos.github.io/plb/

R (for-loop)

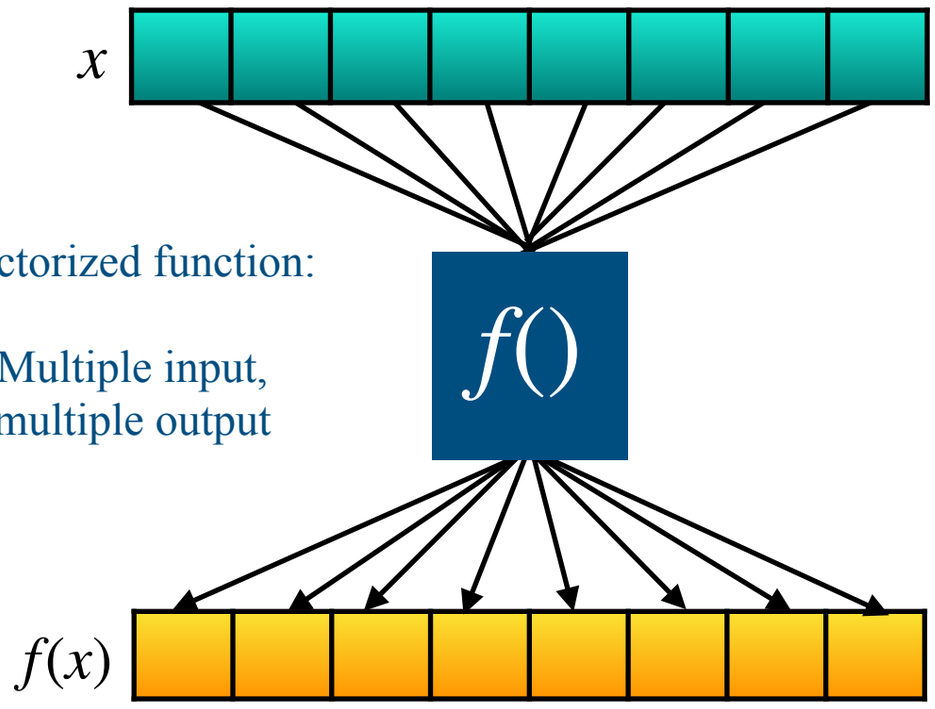
- For-loop in R is indeed very slow:
- ~ 20 times slower than python
- > 200 times slower than C

# Vectorized programming

For-loop:




Vectorization:




- **Vectorization** is the key to speeding up R code.

- It is a trick to replace the explicit for-loop with "internal" for-loop written in a faster language (e.x. C / C++)

# Time for examples

vectorization.rmd

# Part 2. Accurate code

# Overcoming error

Useful general strategies:

- Evaluate outcomes line by line to confirm your beliefs.
- Design mini-experiments to investigate your code.
- Use small input examples.

But what if an error occurred within a function having many lines of code?

# Debug functions

For R, remember these two important functions:

- Debug( )

Debug a function from the beginning with a debug console.

- Browser( )

Call debug console at a specific location we choose.

# Let’s see examples

debug.rmd

# Part 3. Reproducible code

# Motivation of modular programming

Many bad human habits leave the code hard to read and reuse:

- Magic numbers* in the code.
- A similar code structure is often re-emerged.
- Lack of clear documentation on the meanings of variables.

* See through **this link** for more coding style guides.

# Functions

Functions are summarized code structures in which

- Magic numbers are organized into function **parameters**.

- Repeated structures are reduced into a single code **module**.

# Many functions

How to call many functions at once?

- Put all the related functions into a **script** and use source( ).

- When your script is robust and complete, you can make it into a package.

- Packages provide better **documentation** than scripts.

See **this book** for writing R package.

# Reproducibility

## RMD / package vignette

- It is important to let others repeat the key findings in your report/analysis.

- Your package and its user manual are excellent for such a purpose.

- The package vignette is often written in Rmarkdown.

- With some (intermediate) inputs, people should be able to see the paper outcomes by evaluating the RMD file.

# Thank you!