Functional microbiome research – bioinformatics section

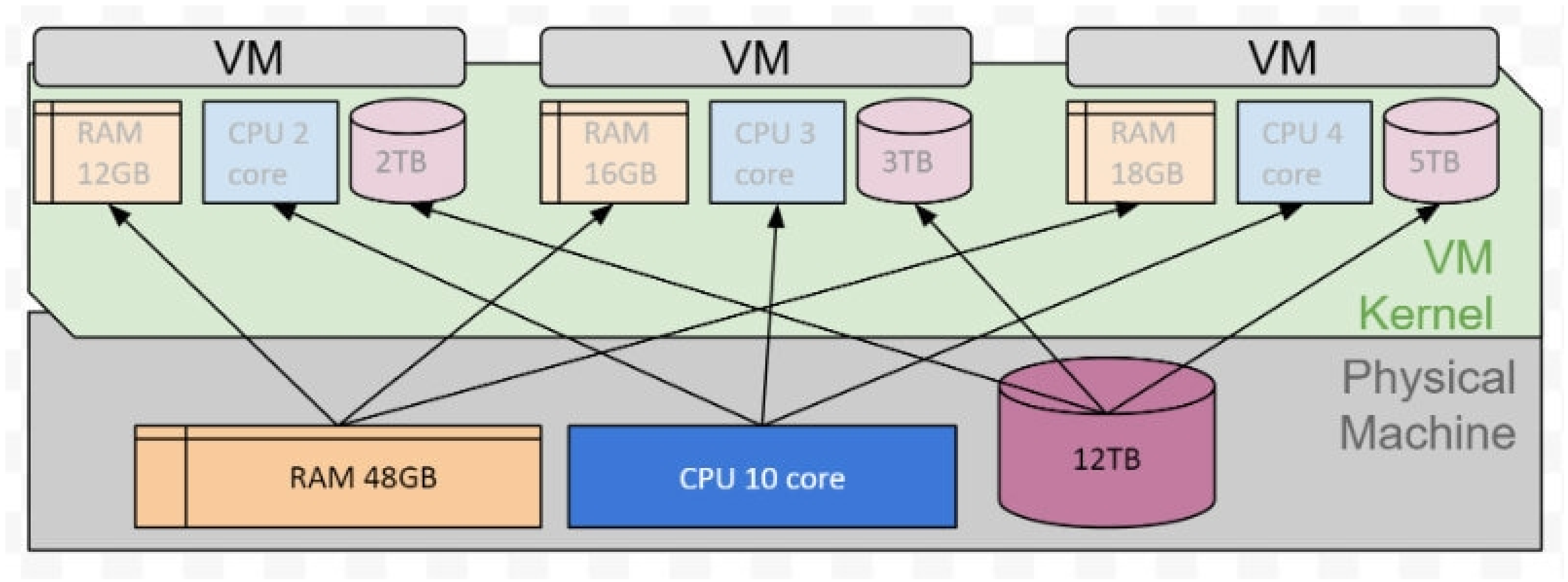
Day 1 – intro to linux

de.NBI

GERMAN NETWORK FOR BIOINFORMATICS INFRASTRUCTURE

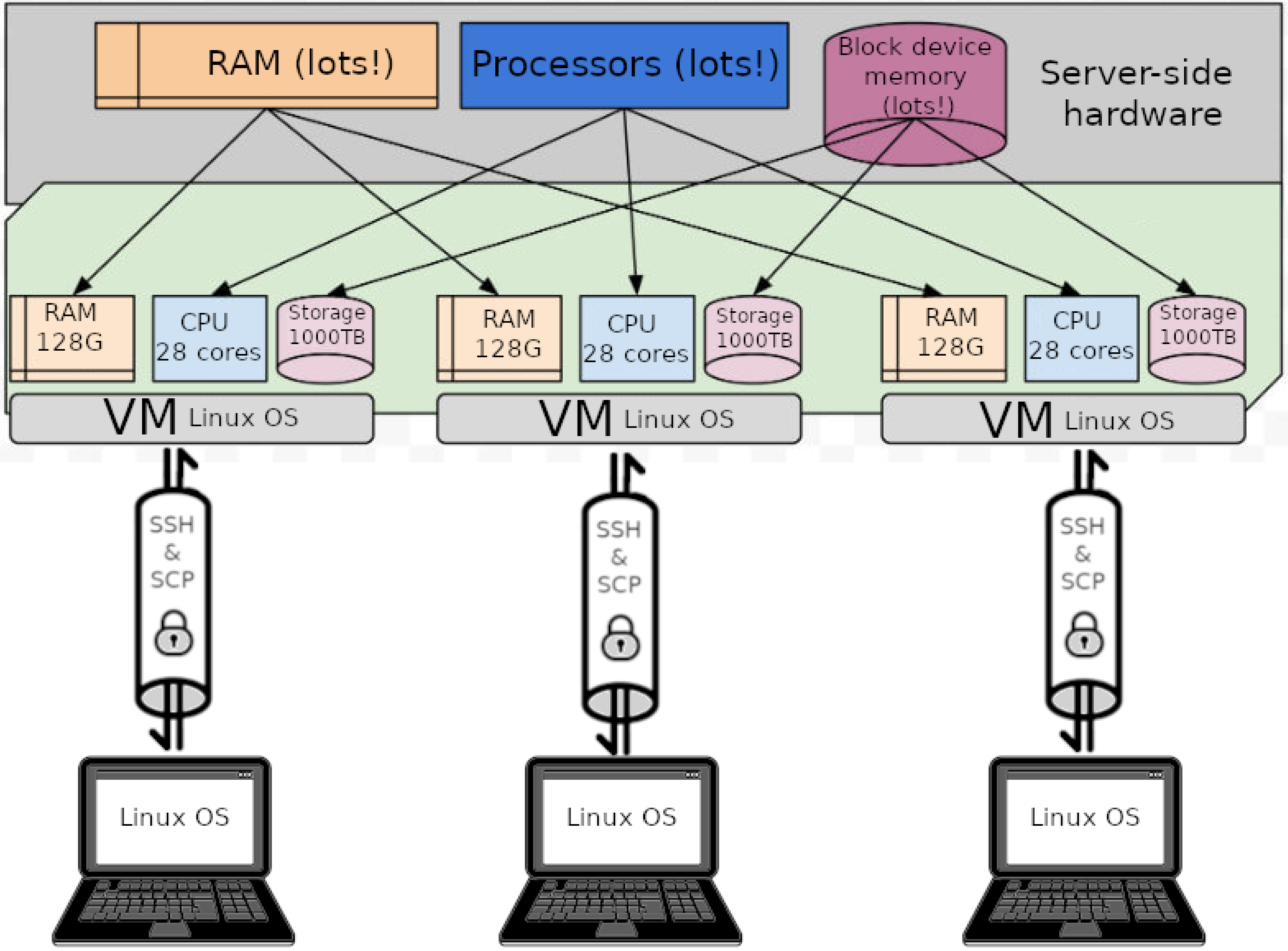
First, thanks to de.NBI. They are amazing.

Distributed computing and compute clusters are essential tools for scientists.



https://actusdigital.com/2018/01/08/virtual-machine-or-physical-server/

We use our local computers to talk to shared remote servers that have lots of high-powered processors, and storage and RAM.

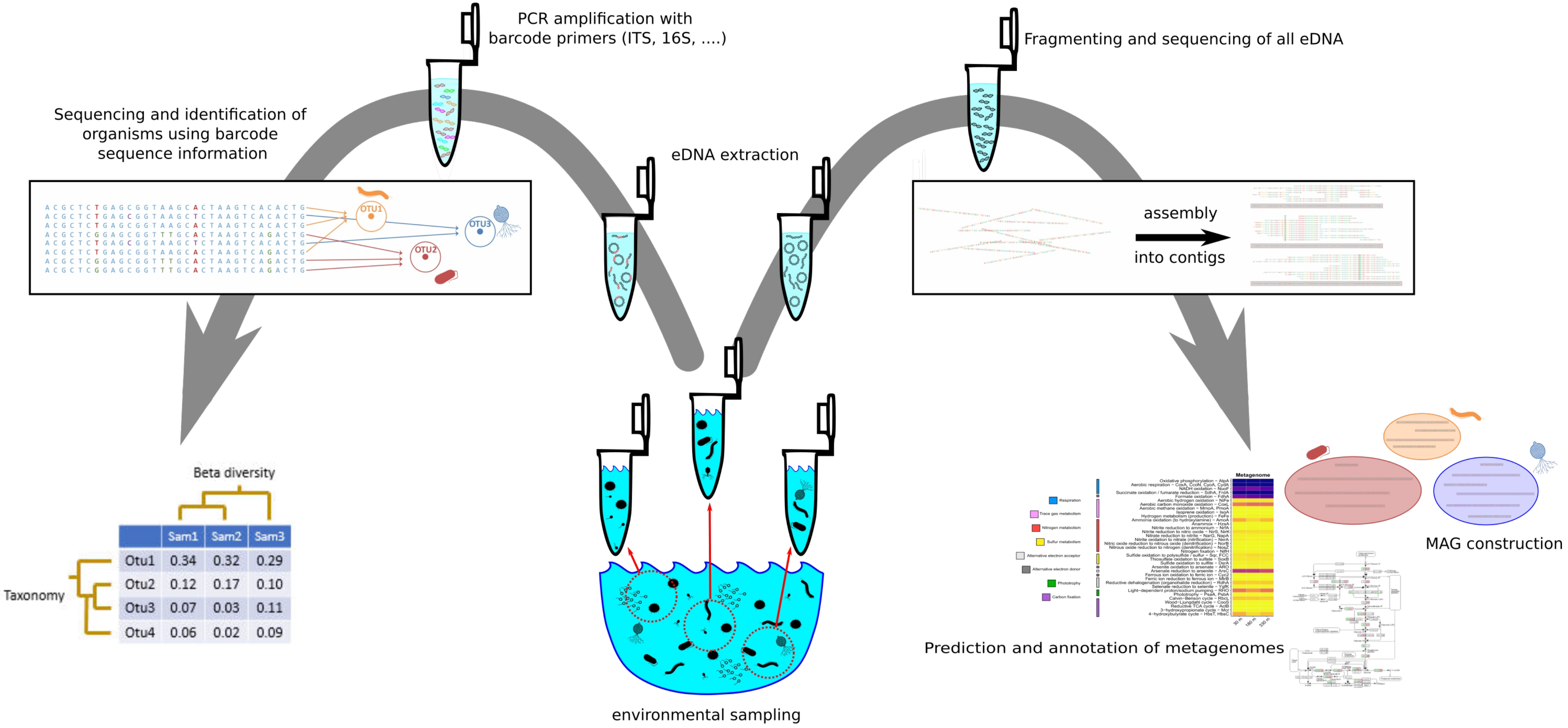


https://actusdigital.com/2018/01/08/virtual-machine-or-physical-server/

de.NBI / Universität Bielefeld

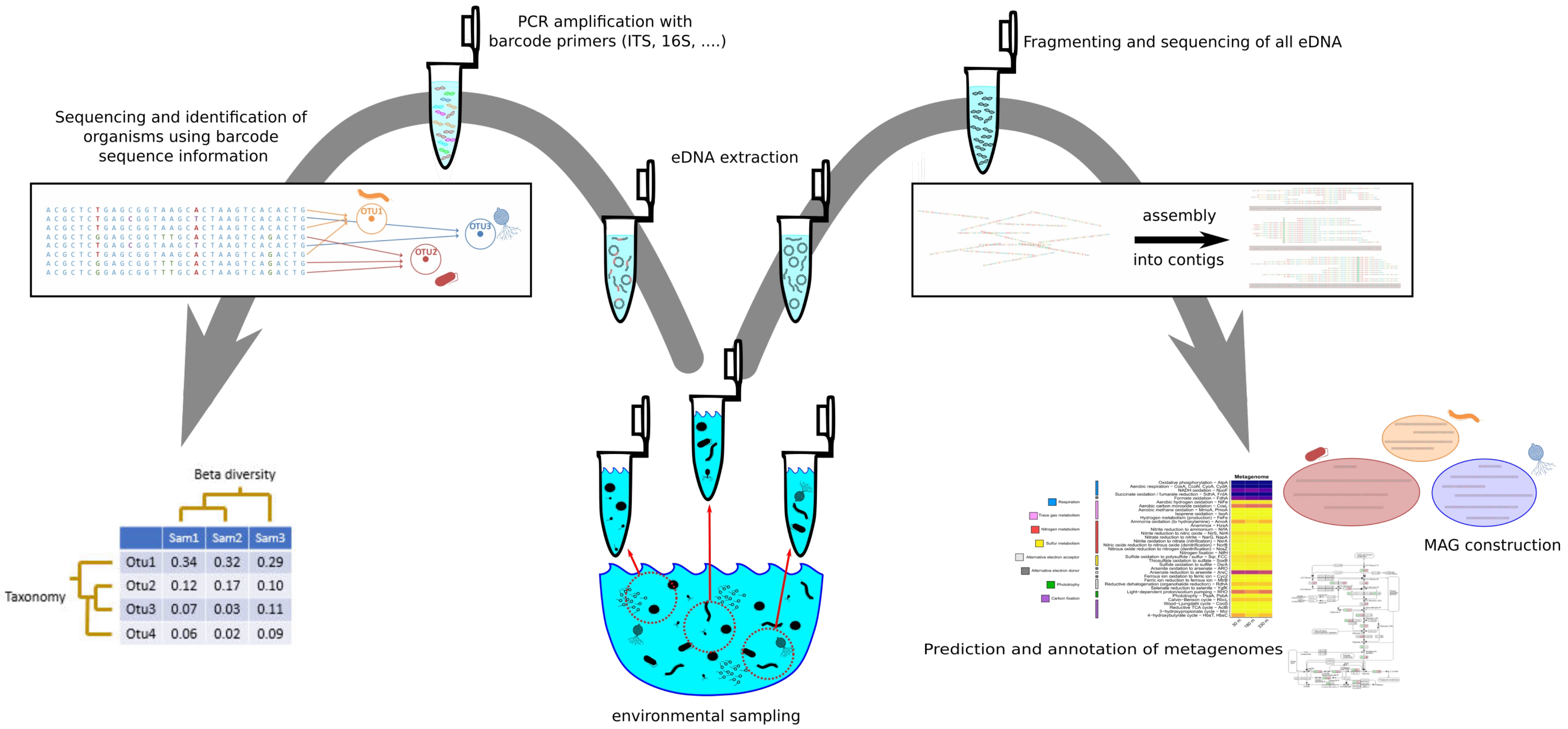
Us.

Review – what do we do with DNA sequence data?


PCR amplification with barcode primers (ITS, 16S, ...)
Fragmenting and sequencing of all eDNA
Sequencing and identification of organisms using barcode sequence information
eDNA extraction
OTU1
OTU2
OTU3
assembly
into contigs
Beta diversity
Taxonomy
Metagenome
MAG construction
Prediction and annotation of metagenomes
environmental sampling

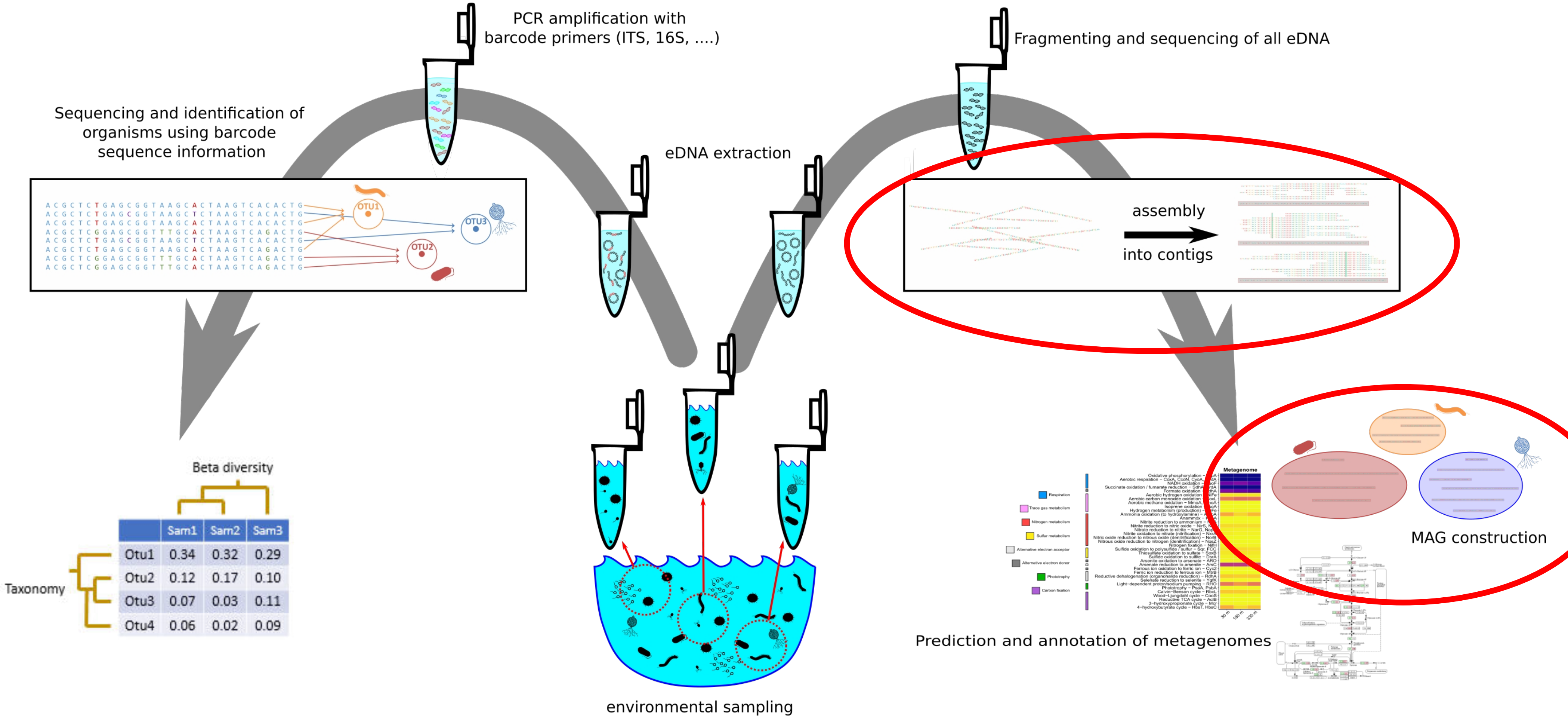
	Sam1	Sam2	Sam3
Otu1	0.34	0.32	0.29
Otu2	0.12	0.17	0.10
Otu3	0.07	0.03	0.11
Otu4	0.06	0.02	0.09

Why do we need distributed computing to do this?


PCR amplification with barcode primers (ITS, 16S, ...)
Fragmenting and sequencing of all eDNA
Sequencing and identification of organisms using barcode sequence information
eDNA extraction
OTU1
OTU2
OTU3
assembly
into contigs
Beta diversity
Taxonomy
Metagenome
MAG construction
Prediction and annotation of metagenomes
environmental sampling

	Sam1	Sam2	Sam3
Otu1	0.34	0.32	0.29
Otu2	0.12	0.17	0.10
Otu3	0.07	0.03	0.11
Otu4	0.06	0.02	0.09

Because some steps take ~100 gig of RAM (or more!) and lots of processors.

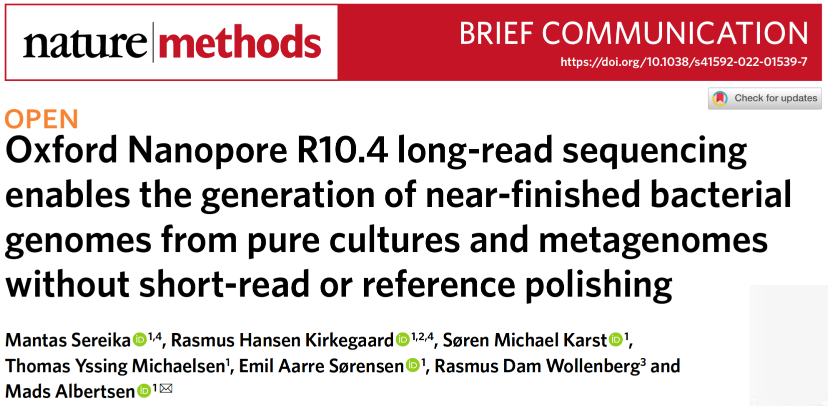

PCR amplification with barcode primers (ITS, 16S, ...)
Fragmenting and sequencing of all eDNA
Sequencing and identification of organisms using barcode sequence information
eDNA extraction
OTU1
OTU2
OTU3
assembly
into contigs
Beta diversity
Taxonomy
MAG construction
Metagenome
Prediction and annotation of metagenomes
environmental sampling

	Sam1	Sam2	Sam3
Otu1	0.34	0.32	0.29
Otu2	0.12	0.17	0.10
Otu3	0.07	0.03	0.11
Otu4	0.06	0.02	0.09

Dataset 1: ZymoBIOMICS HMW DNA Standard.

Species	Theoretical Composition (%)				
	Genomic DNA	16S Only[1]	16S & 18S[1]	Genome Copy[2]	Cell Number[3]
Pseudomonas aeruginosa	14	5.1	4.6	7.8	7.9
Escherichia coli	14	12.4	11.2	10.9	11.0
Salmonella enterica	14	12.7	11.4	11.2	11.2
Enterococcus faecalis	14	12.1	10.9	18.8	18.8
Staphylococcus aureus	14	19	17.1	19.6	19.6
Listeria monocytogenes	14	17.3	15.6	17.8	17.9
Bacillus subtilis	14	21.4	19.2	13.2	13.2
Saccharomyces cerevisiae	2	NA	10	0.63	0.32

Dataset 2: Seika paper


nature methods BRIEF COMMUNICATION

https://doi.org/10.1038/s41592-022-01539-7

Check for updates

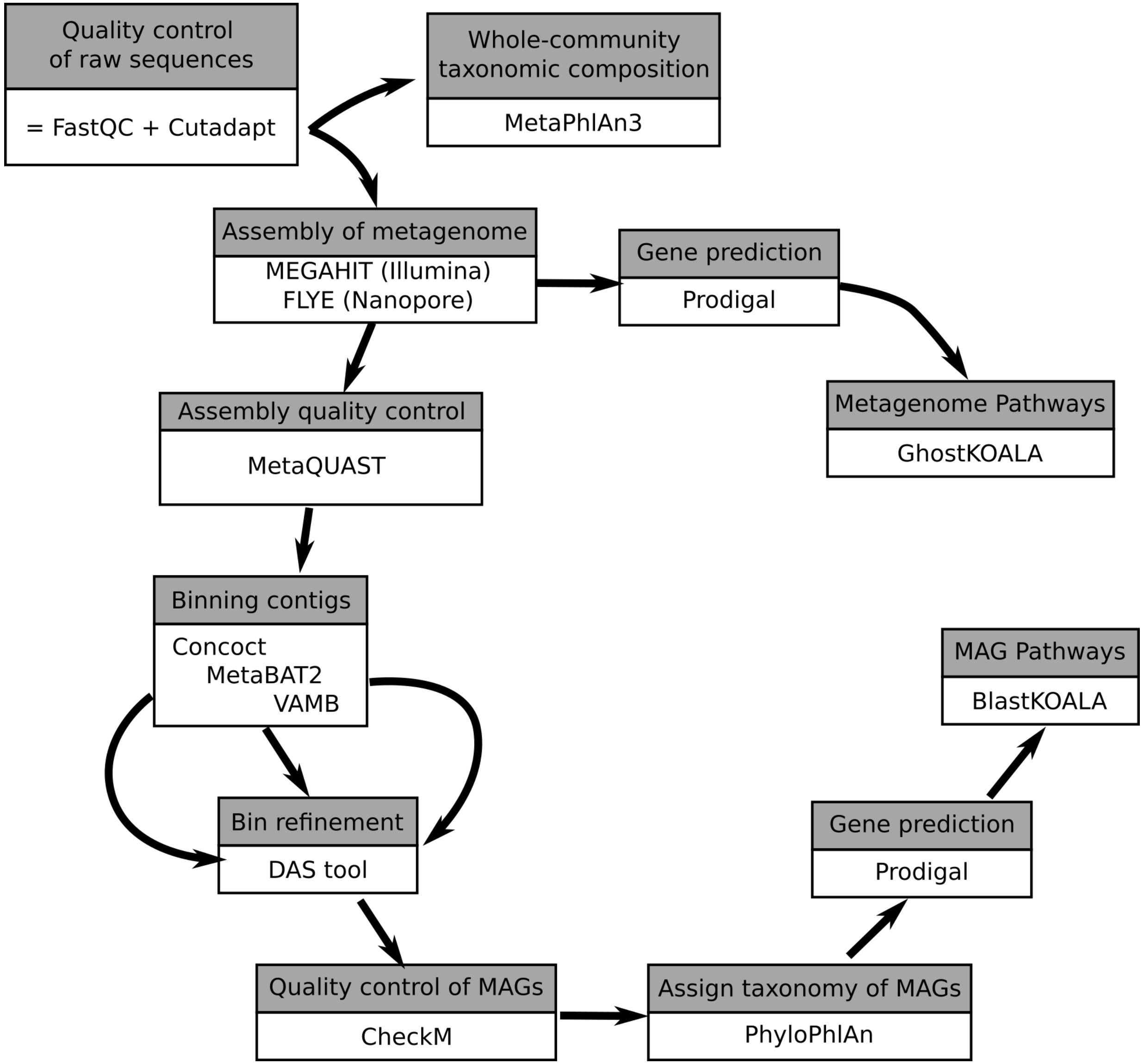
OPEN

Oxford Nanopore R10.4 long-read sequencing enables the generation of near-finished bacterial genomes from pure cultures and metagenomes without short-read or reference polishing

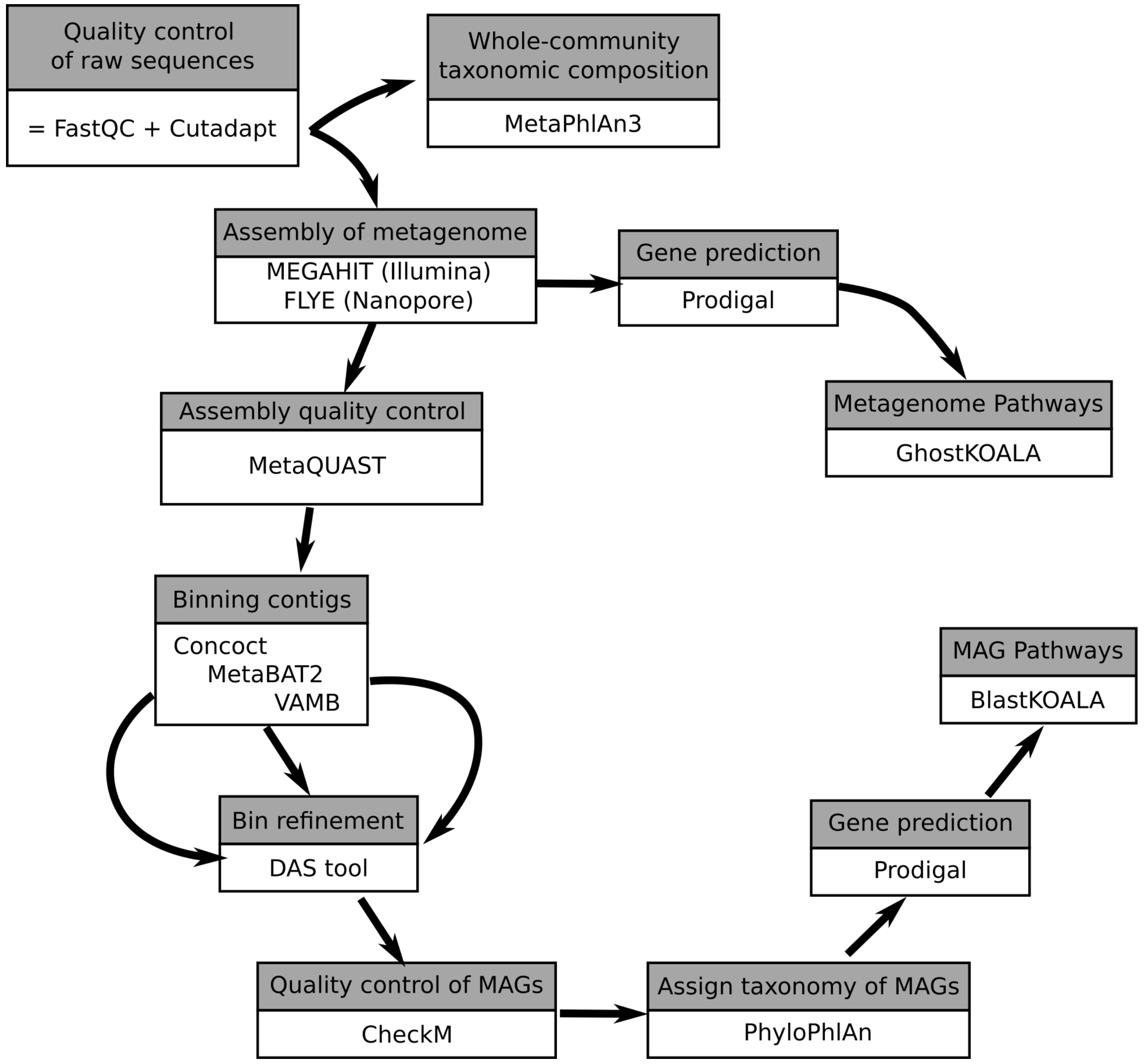
Mantas Sereika[1,4], Rasmus Hansen Kirkegaard[1,2,4], Søren Michael Karst[1], Thomas Yssing Michaelsen[1], Emil Aarre Sørensen[1], Rasmus Dam Wollenberg[3] and Mads Albertsen[1] ✉

Fredericia – Centralrenseanlægget is the second largest wastewater treatment plant in Denmark

https://www.cambi.com/resources/references/europe/denmark/fredericia-centralrenseanlaegget/



We will use various scientific software packages to process our sequence data into a metagenome and Metagenome-Assembled-Genomes (MAGs).



We will do these steps with both illumina and nanopore data, for our learning process with the Zymo Mock Community data.

Intro to command line. Open your terminal and try these:

pwd – print the current working directory

ls – list the contents of the current directory

cd – change directory

cp – copy a file

mv – move or rename a file

mkdir – make a new directory

echo – print something out to the display

cat – concatenate two or more files

rm [-r] – remove a file. Essentially deletes a file forever. Be careful!!

sudo – execute another command that requires root privileges

Intro to command line. Open your terminal and try these:

top – opens up a real time display that shows you the “busiest” processes ongoing in your machine. After you start the program, press i to see only active processes, press e to make memory units more readable, and press 1 then t to keep a close eye on your cores.

man – print the reference manual for a utility, command, or program, if available. If not, try the --help flag for any given program without a manual.

head – show top several lines from a text file

tail – show last several lines from a text file

less – interactively open a text file for reading

Intro to command line. Open your terminal and try these:

Special symbols are very important to the linux command line.

~	home directory (try with cd)
.	current directory (try with cd). Also filenames that start with "." are "hidden".
..	parent directory (try with cd)
>	direct the output of a process to a file
$	variable expansion
|	pipe, connects the output from one command to another
;	command separator
/	directory separator (compare to Windows!)
\	escape a character, so that it is no longer a "special character"
&	send a process to run in the background
=	variable assignment
*	wildcard for multiple characters (try with ls)
?	wildcard for single character (try with ls)
#	comment (place it before a command and see what happens)

Intro to command line. Open your terminal and try these:

top – opens up a real time display that shows you the “busiest” processes ongoing in your machine. After you start the program, press `i` to see only active processes, press `e` to make memory units more readable, and press `1` then `t` to keep a close eye on your cores.

man – print the reference manual for a utility, command, or program, if available. If not, try the `--help` flag for any given program without a manual.

head – show top several lines from a text file

tail – show last several lines from a text file

less – interactively open a text file for reading

Special command: **wget**

We’ll keep track of what we do in scripts that will be kept in an online github repository. Let’s practice grabbing the latest version:

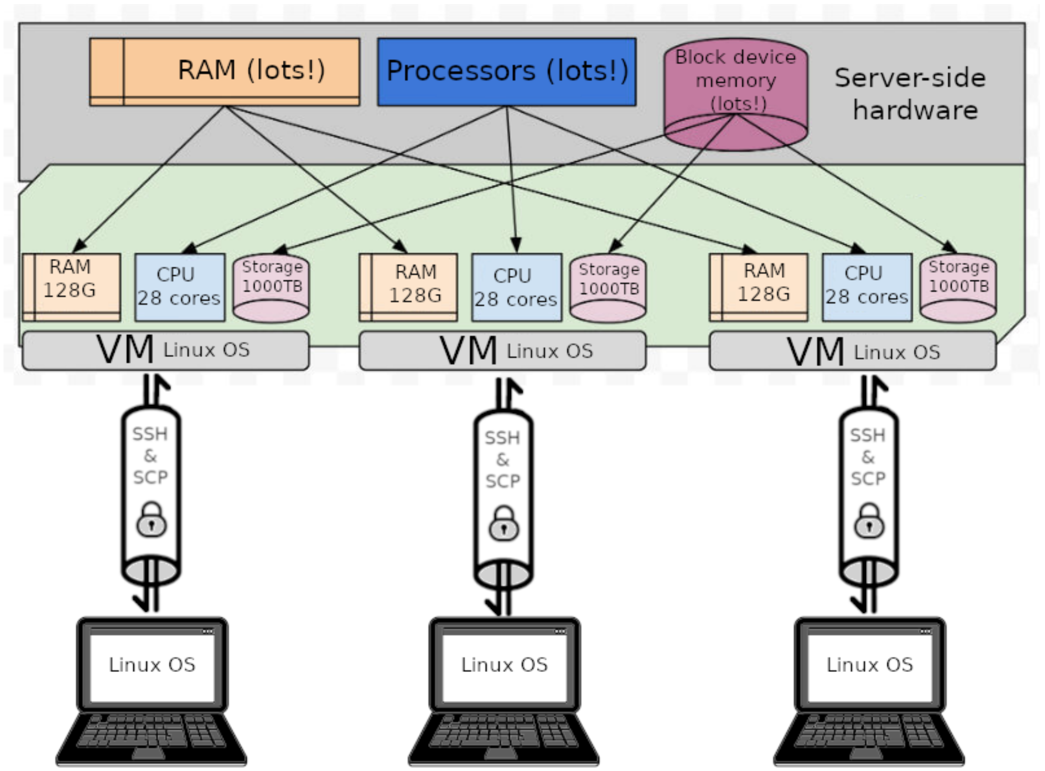
```
wget https://raw.githubusercontent.com/danchurch/FunctionalMicrobiomePractical2022/main/funmic2023/funBASHterminalScript.txt
```

(Maybe better to follow the link in the skript PDF on page 20, or download this presentation, then copy-paste)

First task: connect to VMs

Milan	ssh -p 30284 -i /path/to/your/privateKey ubuntu@129.70.51.6
Julia	ssh -p 30381 -i /path/to/your/privateKey ubuntu@129.70.51.6
Maria	ssh -p 30355 -i /path/to/your/privateKey ubuntu@129.70.51.6
Anja	ssh -p 30452 -i /path/to/your/privateKey ubuntu@129.70.51.6
Susanne	ssh -p 30461 -i /path/to/your/privateKey ubuntu@129.70.51.6
Rebekka	ssh -p 30372 -i /path/to/your/privateKey ubuntu@129.70.51.6
Or try with a shell variable:	key=/home/daniel/.ssh/funmic2023 ssh -p 30500 -i $key ubuntu@129.70.51.6

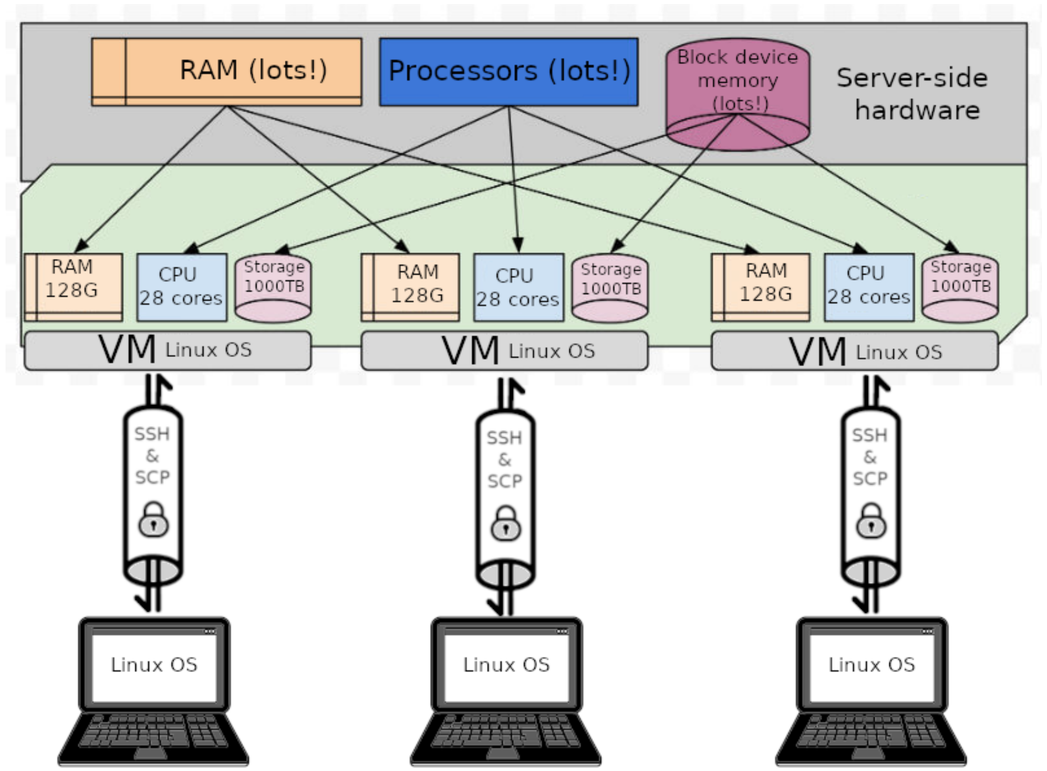
First task: connect to VMs



cd /vol/danBot

Once you are in, try to find your storage volume.

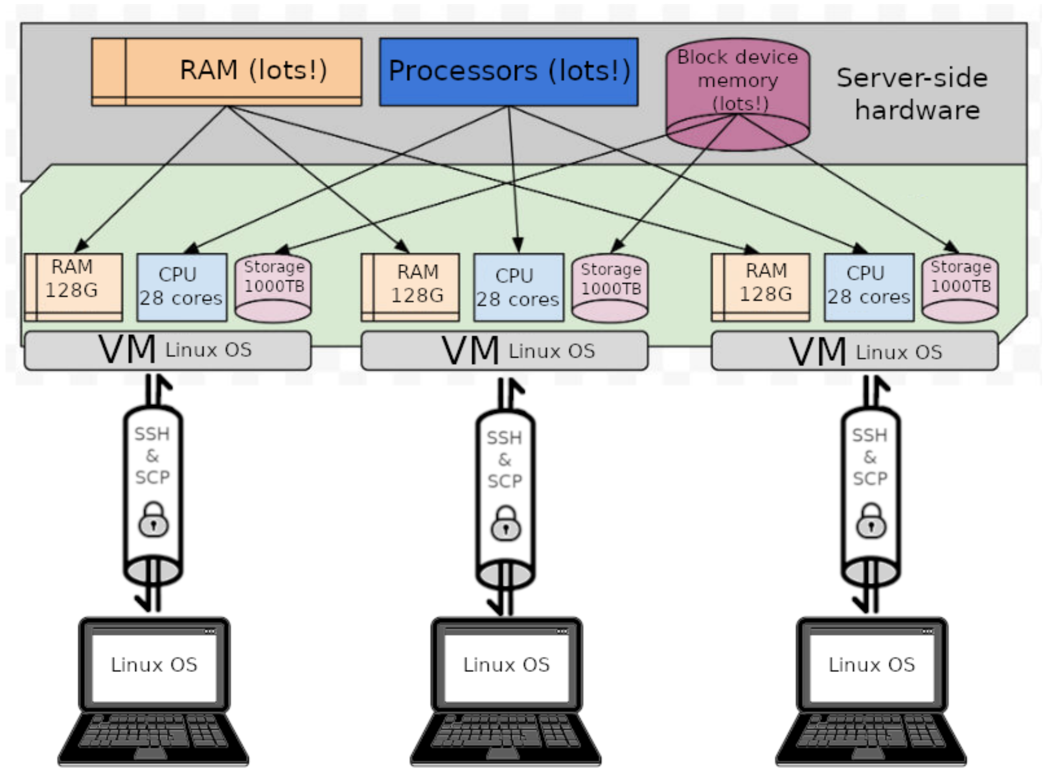
Second task: get a file from your VM to your computer



You will often need to get files from your de.NBM VM, to inspect them locally either in your linux or windows evironment.

$$\begin{tikzcd}\text{de.NBI virtual machine} \arrow[r] & \text{Local linux environment} \arrow[r] & \text{Windows} \arrow[r] & \text{Browsers, Excel, other GUIs…}\end{tikzcd}$$

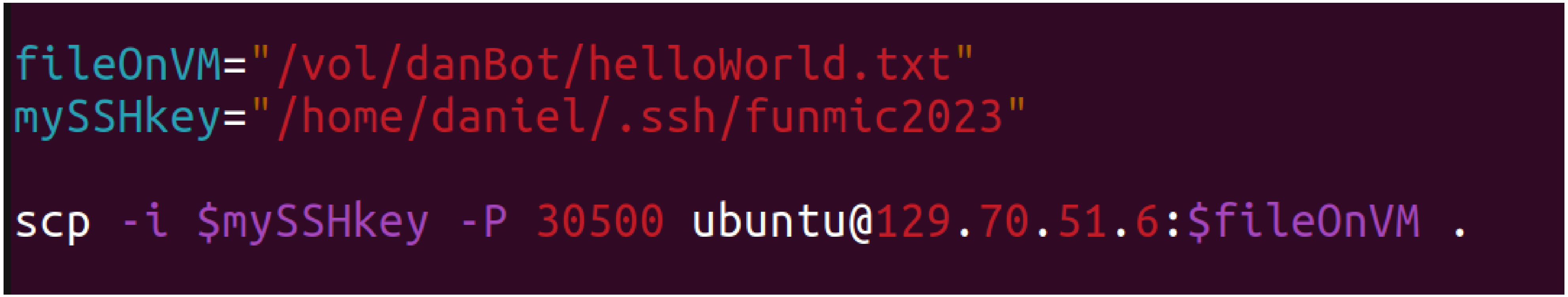
Second task: get a file from your VM to your computer



We will use the **scp** program for this.

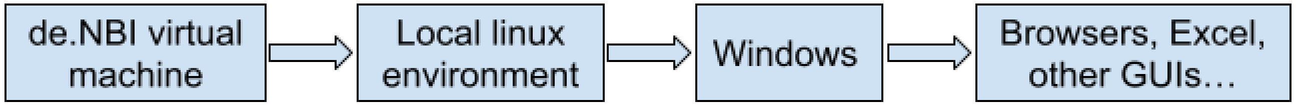
$$\begin{tikzcd}
\text{de.NBI virtual machine} \arrow[r] & \text{Local linux environment} \arrow[r] & \text{Windows} \arrow[r] & \text{Browsers, Excel, other GUIs…}
\end{tikzcd}$$

Second task: get a file from your VM to your computer

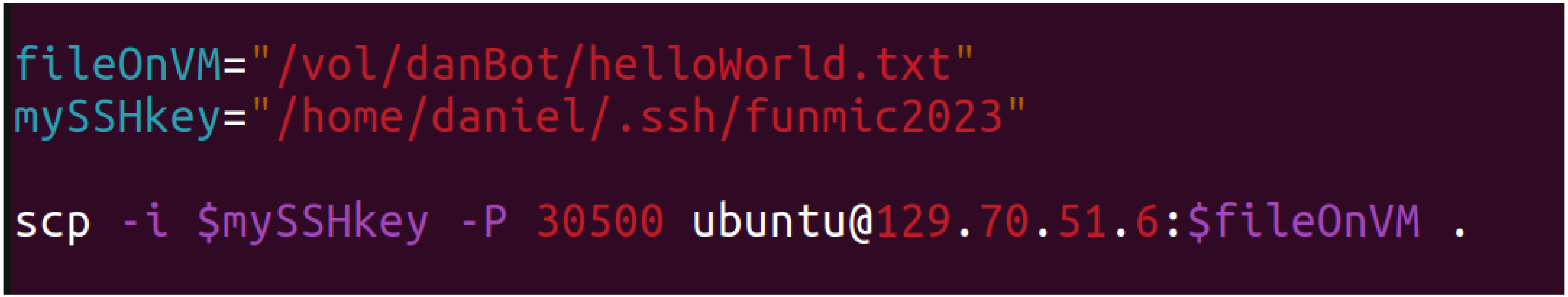


```
fileOnVM="/vol/danBot/helloWorld.txt"
mySSHkey="/home/daniel/.ssh/funmic2023"

scp -i $mySSHkey -P 30500 ubuntu@129.70.51.6:$fileOnVM .
```

Try **scp** , to get “helloWorld.txt” from your de.NBI VM to your local machine.

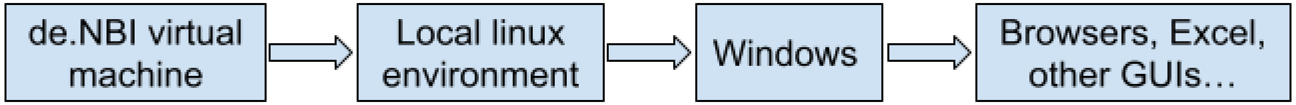


Second task: get a file from your VM to your computer



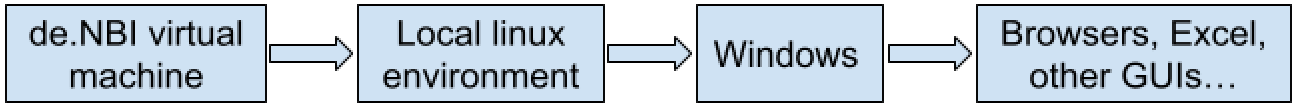
```
fileOnVM="/vol/danBot/helloWorld.txt"
mySSHkey="/home/daniel/.ssh/funmic2023"

scp -i $mySSHkey -P 30500 ubuntu@129.70.51.6:$fileOnVM .
```

(Can copy and paste from scripts, but make sure you understand!)



Second task: get a file from your VM to your computer

Once you have your file in your local linux environment, you will probably want to look at it in your windows environment, too.

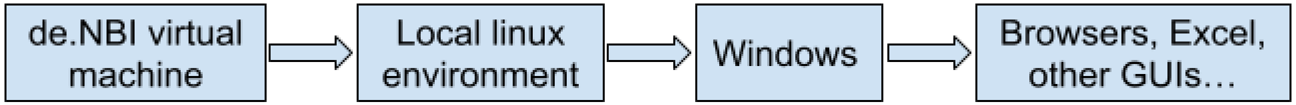


Second task: get a file from your VM to your computer

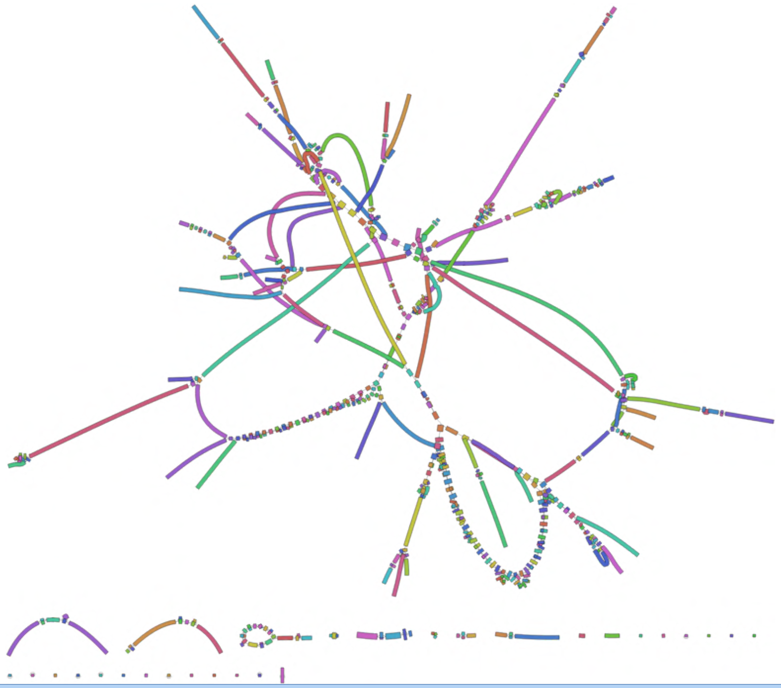
If you are using WSL in windows, the try placing the file here:

```
cd /mnt/c/
```

See if you can find it in windows by looking at your C:\ directory. Considering making a special folder for this kind of file transfer.

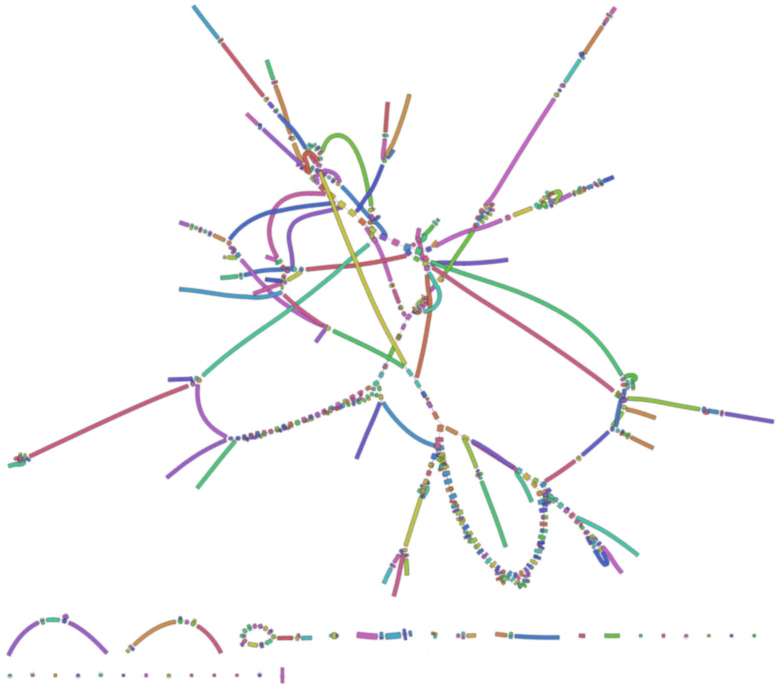


Last task of the day: start your nanopore assembly.



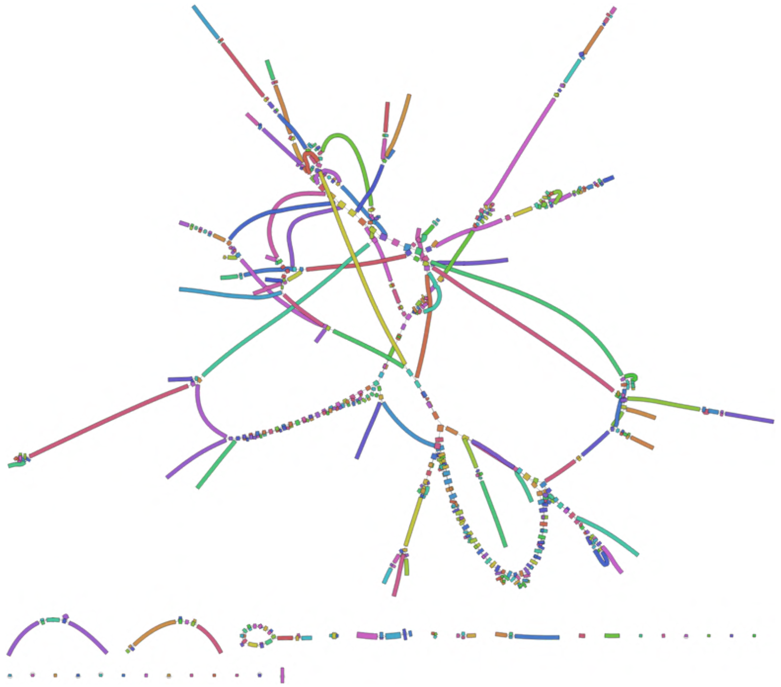
Assembly of metagenomes is one of the most computationally intensive tasks we will ask our de.NBI machines to do for us. We will review metagenome assembly a bit romorrow, but let’s let’s get our first assembly started now, so it can as long as it needs to.

Last task of the day: start your nanopore assembly.



This will be the first of two assemblies we do, since we have both an illumina and a nanopore dataset from the same Zymo mock community. We will start with the nanopore data, since it takes much more time and computing resources.

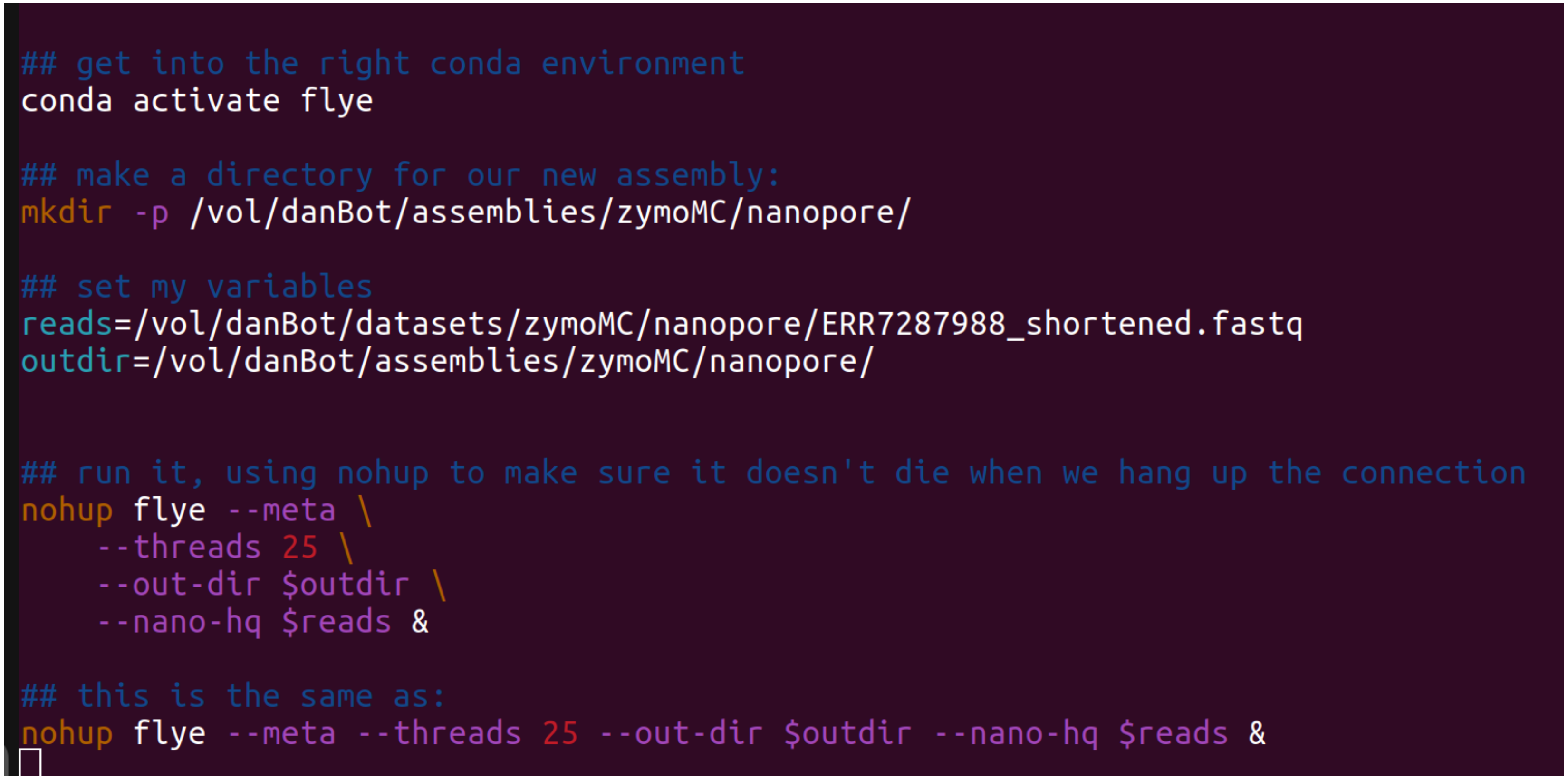
Last task of the day: start your nanopore assembly.



This also gives a chance to see a complex command in action, that requires us to use much of what we just learned about the linux command line.

Last task of the day: start your nanopore assembly.

From my de.NBI machine, the command looks like this:

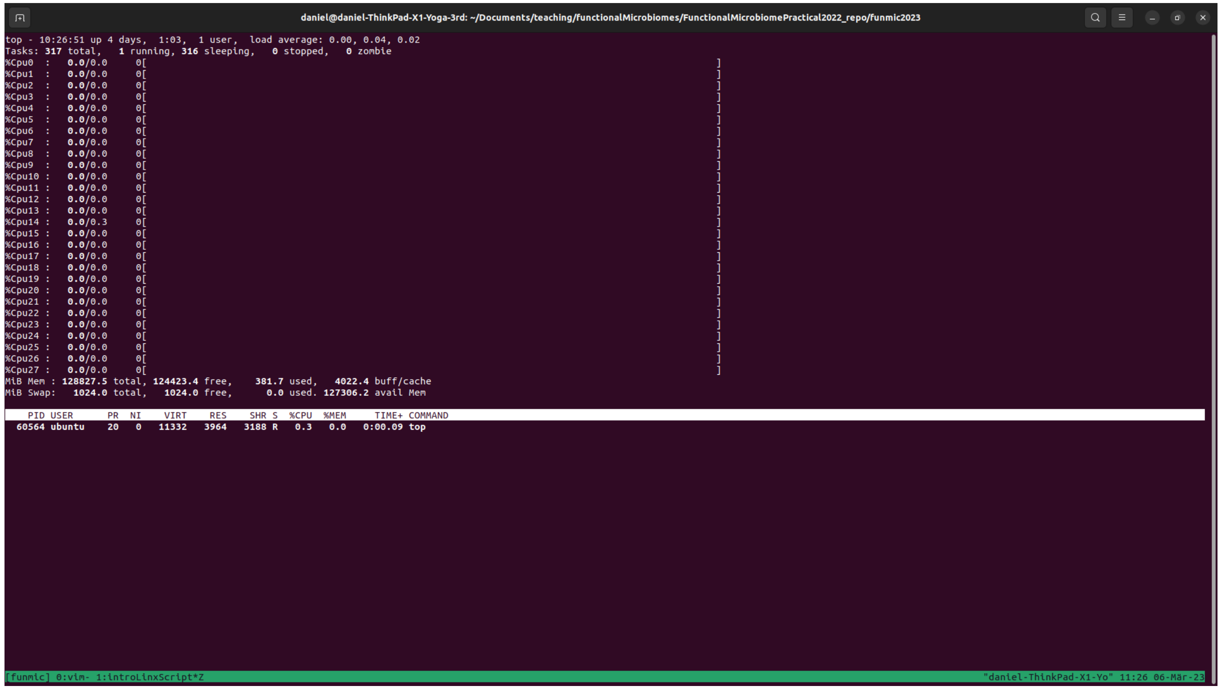


```
## get into the right conda environment
conda activate flye

## make a directory for our new assembly:
mkdir -p /vol/danBot/assemblies/zymoMC/nanopore/

## set my variables
reads=/vol/danBot/datasets/zymoMC/nanopore/ERR7287988_shortened.fastq
outdir=/vol/danBot/assemblies/zymoMC/nanopore/


## run it, using nohup to make sure it doesn't die when we hang up the connection
nohup flye --meta \
    --threads 25 \
    --out-dir $outdir \
    --nano-hq $reads &

## this is the same as:
nohup flye --meta --threads 25 --out-dir $outdir --nano-hq $reads &
```

(You can see the full help file by typing flye --help)

Last task of the day: start your nanopore assembly.

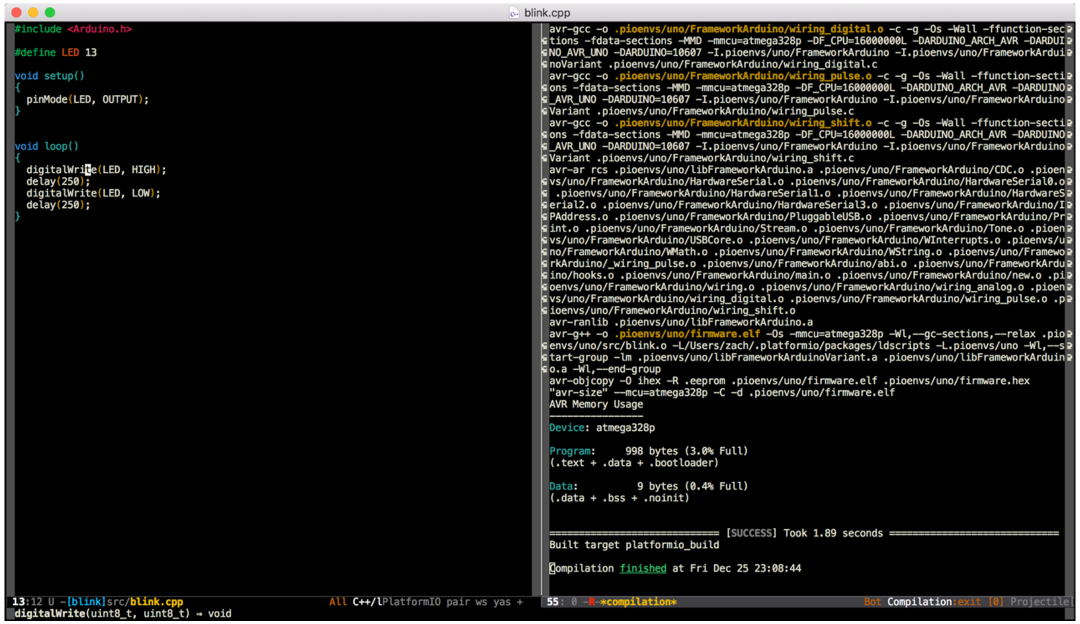
Did that work?



```
daniel@daniel-ThinkPad-X1-Yoga-3rd: ~/Documents/teaching/functionalMicrobiomes/FunctionalMicrobiomePractical2022_repo/funmic2023

top - 10:26:51 up 4 days,  1:03,  1 user,  load average: 0.00, 0.04, 0.02
Tasks: 317 total,   1 running, 316 sleeping,   0 stopped,   0 zombie
%Cpu0  :   0.0/0.0     0[                                                                                                   ]
%Cpu1  :   0.0/0.0     0[                                                                                                   ]
%Cpu2  :   0.0/0.0     0[                                                                                                   ]
%Cpu3  :   0.0/0.0     0[                                                                                                   ]
%Cpu4  :   0.0/0.0     0[                                                                                                   ]
%Cpu5  :   0.0/0.0     0[                                                                                                   ]
%Cpu6  :   0.0/0.0     0[                                                                                                   ]
%Cpu7  :   0.0/0.0     0[                                                                                                   ]
%Cpu8  :   0.0/0.0     0[                                                                                                   ]
%Cpu9  :   0.0/0.0     0[                                                                                                   ]
%Cpu10 :   0.0/0.0     0[                                                                                                   ]
%Cpu11 :   0.0/0.0     0[                                                                                                   ]
%Cpu12 :   0.0/0.0     0[                                                                                                   ]
%Cpu13 :   0.0/0.0     0[                                                                                                   ]
%Cpu14 :   0.0/0.3     0[                                                                                                   ]
%Cpu15 :   0.0/0.0     0[                                                                                                   ]
%Cpu16 :   0.0/0.0     0[                                                                                                   ]
%Cpu17 :   0.0/0.0     0[                                                                                                   ]
%Cpu18 :   0.0/0.0     0[                                                                                                   ]
%Cpu19 :   0.0/0.0     0[                                                                                                   ]
%Cpu20 :   0.0/0.0     0[                                                                                                   ]
%Cpu21 :   0.0/0.0     0[                                                                                                   ]
%Cpu22 :   0.0/0.0     0[                                                                                                   ]
%Cpu23 :   0.0/0.0     0[                                                                                                   ]
%Cpu24 :   0.0/0.0     0[                                                                                                   ]
%Cpu25 :   0.0/0.0     0[                                                                                                   ]
%Cpu26 :   0.0/0.0     0[                                                                                                   ]
%Cpu27 :   0.0/0.0     0[                                                                                                   ]
MiB Mem : 128827.5 total, 124423.4 free,    381.7 used,   4022.4 buff/cache
MiB Swap:   1024.0 total,   1024.0 free,      0.0 used. 127306.2 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  60564 ubuntu    20   0   11332   3964   3188 R   0.3   0.0   0:00.09 top

[funmic] 0:vim- 1:introLinxScript*Z                                           "daniel-ThinkPad-X1-Yo" 11:26 06-Mär-23
```

Let’s check with the top program.

And final note for the day: keep a script of what you do!!!



blink.cpp

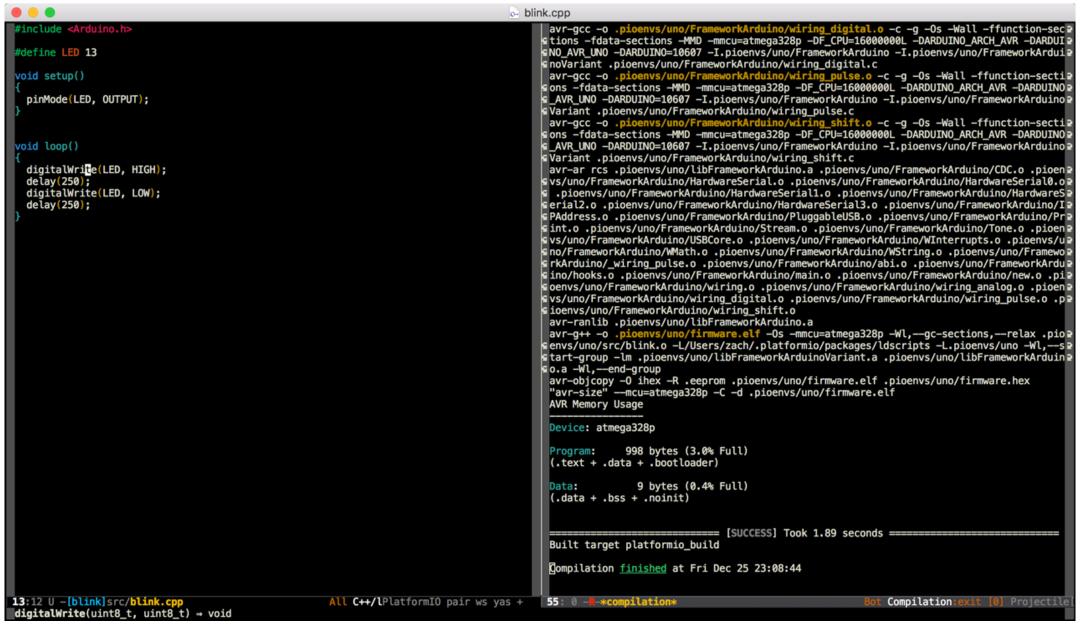
```cpp
#include <Arduino.h>

#define LED 13

void setup()
{
  pinMode(LED, OUTPUT);
}


void loop()
{
  digitalWrite(LED, HIGH);
  delay(250);
  digitalWrite(LED, LOW);
  delay(250);
}
```

```
13:12 U -[blink]src/blink.cpp                  All C++/lPlatformIO pair ws yas +
digitalWrite(uint8_t, uint8_t) → void
```

```
avr-gcc -o .pioenvs/uno/FrameworkArduino/wiring_digital.o -c -g -Os -Wall -ffunction-sec
tions -fdata-sections -MMD -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUI
NO_AVR_UNO -DARDUINO=10607 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArdui
noVariant .pioenvs/uno/FrameworkArduino/wiring_digital.c
avr-gcc -o .pioenvs/uno/FrameworkArduino/wiring_pulse.o -c -g -Os -Wall -ffunction-secti
ons -fdata-sections -MMD -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO
_AVR_UNO -DARDUINO=10607 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduino
Variant .pioenvs/uno/FrameworkArduino/wiring_pulse.c
avr-gcc -o .pioenvs/uno/FrameworkArduino/wiring_shift.o -c -g -Os -Wall -ffunction-secti
ons -fdata-sections -MMD -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO
_AVR_UNO -DARDUINO=10607 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduino
Variant .pioenvs/uno/FrameworkArduino/wiring_shift.c
avr-ar rcs .pioenvs/uno/libFrameworkArduino.a .pioenvs/uno/FrameworkArduino/CDC.o .pioen
vs/uno/FrameworkArduino/HardwareSerial.o .pioenvs/uno/FrameworkArduino/HardwareSerial0.o
 .pioenvs/uno/FrameworkArduino/HardwareSerial1.o .pioenvs/uno/FrameworkArduino/HardwareS
erial2.o .pioenvs/uno/FrameworkArduino/HardwareSerial3.o .pioenvs/uno/FrameworkArduino/I
PAddress.o .pioenvs/uno/FrameworkArduino/PluggableUSB.o .pioenvs/uno/FrameworkArduino/Pr
int.o .pioenvs/uno/FrameworkArduino/Stream.o .pioenvs/uno/FrameworkArduino/Tone.o .pioen
vs/uno/FrameworkArduino/USBCore.o .pioenvs/uno/FrameworkArduino/WInterrupts.o .pioenvs/u
no/FrameworkArduino/WMath.o .pioenvs/uno/FrameworkArduino/WString.o .pioenvs/uno/Framewo
rkArduino/_wiring_pulse.o .pioenvs/uno/FrameworkArduino/abi.o .pioenvs/uno/FrameworkArdu
ino/hooks.o .pioenvs/uno/FrameworkArduino/main.o .pioenvs/uno/FrameworkArduino/new.o .pi
oenvs/uno/FrameworkArduino/wiring.o .pioenvs/uno/FrameworkArduino/wiring_analog.o .pioen
vs/uno/FrameworkArduino/wiring_digital.o .pioenvs/uno/FrameworkArduino/wiring_pulse.o .p
ioenvs/uno/FrameworkArduino/wiring_shift.o
avr-ranlib .pioenvs/uno/libFrameworkArduino.a
avr-g++ -o .pioenvs/uno/firmware.elf -Os -mmcu=atmega328p -Wl,--gc-sections,--relax .pio
envs/uno/src/blink.o -L/Users/zach/.platformio/packages/ldscripts -L.pioenvs/uno -Wl,--s
tart-group -lm .pioenvs/uno/libFrameworkArduinoVariant.a .pioenvs/uno/libFrameworkArduin
o.a -Wl,--end-group
avr-objcopy -O ihex -R .eeprom .pioenvs/uno/firmware.elf .pioenvs/uno/firmware.hex
"avr-size" --mcu=atmega328p -C -d .pioenvs/uno/firmware.elf
AVR Memory Usage
----------------
Device: atmega328p

Program:     998 bytes (3.0% Full)
(.text + .data + .bootloader)

Data:          9 bytes (0.4% Full)
(.data + .bss + .noinit)


============================= [SUCCESS] Took 1.89 seconds =============================
Built target platformio_build

Compilation finished at Fri Dec 25 23:08:44

55: 0 -R-*compilation*                                      Bot Compilation:exit [0] Projectile
```

With scientific computing, it can sometimes take hours to build, debug, and successfully execute a single program, sometimes just one line of code.

And final note for the day: keep a script of what you do!!!



blink.cpp

```
#include <Arduino.h>

#define LED 13

void setup()
{
  pinMode(LED, OUTPUT);
}


void loop()
{
  digitalWrite(LED, HIGH);
  delay(250);
  digitalWrite(LED, LOW);
  delay(250);
}
```

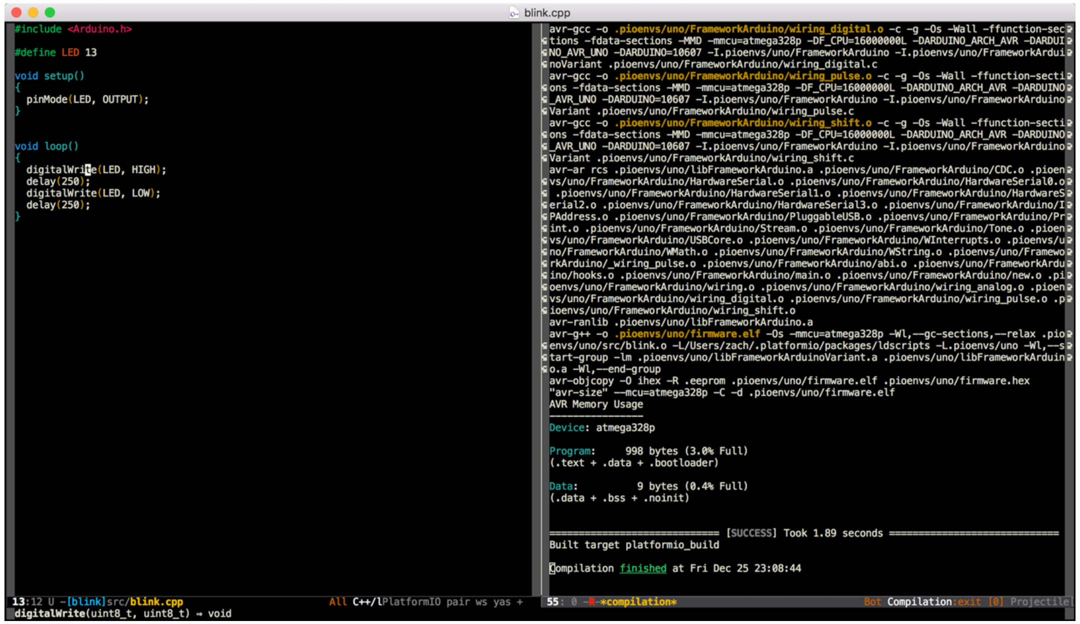
```
13:12 U -[blink]src/blink.cpp                    All C++/lPlatformIO pair ws yas +
digitalWrite(uint8_t, uint8_t) → void
```

```
avr-gcc -o .pioenvs/uno/FrameworkArduino/wiring_digital.o -c -g -Os -Wall -ffunction-sec
tions -fdata-sections -MMD -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUI
NO_AVR_UNO -DARDUINO=10607 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArdui
noVariant .pioenvs/uno/FrameworkArduino/wiring_digital.c
avr-gcc -o .pioenvs/uno/FrameworkArduino/wiring_pulse.o -c -g -Os -Wall -ffunction-secti
ons -fdata-sections -MMD -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO
_AVR_UNO -DARDUINO=10607 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduino
Variant .pioenvs/uno/FrameworkArduino/wiring_pulse.c
avr-gcc -o .pioenvs/uno/FrameworkArduino/wiring_shift.o -c -g -Os -Wall -ffunction-secti
ons -fdata-sections -MMD -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO
_AVR_UNO -DARDUINO=10607 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduino
Variant .pioenvs/uno/FrameworkArduino/wiring_shift.c
avr-ar rcs .pioenvs/uno/libFrameworkArduino.a .pioenvs/uno/FrameworkArduino/CDC.o .pioen
vs/uno/FrameworkArduino/HardwareSerial.o .pioenvs/uno/FrameworkArduino/HardwareSerial0.o
 .pioenvs/uno/FrameworkArduino/HardwareSerial1.o .pioenvs/uno/FrameworkArduino/HardwareS
erial2.o .pioenvs/uno/FrameworkArduino/HardwareSerial3.o .pioenvs/uno/FrameworkArduino/I
PAddress.o .pioenvs/uno/FrameworkArduino/PluggableUSB.o .pioenvs/uno/FrameworkArduino/Pr
int.o .pioenvs/uno/FrameworkArduino/Stream.o .pioenvs/uno/FrameworkArduino/Tone.o .pioen
vs/uno/FrameworkArduino/USBCore.o .pioenvs/uno/FrameworkArduino/WInterrupts.o .pioenvs/u
no/FrameworkArduino/WMath.o .pioenvs/uno/FrameworkArduino/WString.o .pioenvs/uno/Framewo
rkArduino/_wiring_pulse.o .pioenvs/uno/FrameworkArduino/abi.o .pioenvs/uno/FrameworkArdu
ino/hooks.o .pioenvs/uno/FrameworkArduino/main.o .pioenvs/uno/FrameworkArduino/new.o .pi
oenvs/uno/FrameworkArduino/wiring.o .pioenvs/uno/FrameworkArduino/wiring_analog.o .pioen
vs/uno/FrameworkArduino/wiring_digital.o .pioenvs/uno/FrameworkArduino/wiring_pulse.o .p
ioenvs/uno/FrameworkArduino/wiring_shift.o
avr-ranlib .pioenvs/uno/libFrameworkArduino.a
avr-g++ -o .pioenvs/uno/firmware.elf -Os -mmcu=atmega328p -Wl,--gc-sections,--relax .pio
envs/uno/src/blink.o -L/Users/zach/.platformio/packages/ldscripts -L.pioenvs/uno -Wl,--s
tart-group -lm .pioenvs/uno/libFrameworkArduinoVariant.a .pioenvs/uno/libFrameworkArduin
o.a -Wl,--end-group
avr-objcopy -O ihex -R .eeprom .pioenvs/uno/firmware.elf .pioenvs/uno/firmware.hex
"avr-size" --mcu=atmega328p -C -d .pioenvs/uno/firmware.elf
AVR Memory Usage
----------------
Device: atmega328p

Program:     998 bytes (3.0% Full)
(.text + .data + .bootloader)

Data:          9 bytes (0.4% Full)
(.data + .bss + .noinit)


============================ [SUCCESS] Took 1.89 seconds ============================
Built target platformio_build

Compilation finished at Fri Dec 25 23:08:44
```

```
55: 0 -R-*compilation*                                Bot Compilation:exit [0] Projectile
```

You will often want to return to your commands days or months later. Other scientists will want to know precisely what commands and settings you used.

And final note for the day: keep a script of what you do!!!



```
#include <Arduino.h>

#define LED 13

void setup()
{
  pinMode(LED, OUTPUT);
}


void loop()
{
  digitalWrite(LED, HIGH);
  delay(250);
  digitalWrite(LED, LOW);
  delay(250);
}
```

```
13:12 U -[blink]src/blink.cpp                    All C++/lPlatformIO pair ws yas +
digitalWrite(uint8_t, uint8_t) → void
```

```
avr-gcc -o .pioenvs/uno/FrameworkArduino/wiring_digital.o -c -g -Os -Wall -ffunction-sec
tions -fdata-sections -MMD -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUI
NO_AVR_UNO -DARDUINO=10607 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArdui
noVariant .pioenvs/uno/FrameworkArduino/wiring_digital.c
avr-gcc -o .pioenvs/uno/FrameworkArduino/wiring_pulse.o -c -g -Os -Wall -ffunction-secti
ons -fdata-sections -MMD -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO
_AVR_UNO -DARDUINO=10607 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduino
Variant .pioenvs/uno/FrameworkArduino/wiring_pulse.c
avr-gcc -o .pioenvs/uno/FrameworkArduino/wiring_shift.o -c -g -Os -Wall -ffunction-secti
ons -fdata-sections -MMD -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO
_AVR_UNO -DARDUINO=10607 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduino
Variant .pioenvs/uno/FrameworkArduino/wiring_shift.c
avr-ar rcs .pioenvs/uno/libFrameworkArduino.a .pioenvs/uno/FrameworkArduino/CDC.o .pioen
vs/uno/FrameworkArduino/HardwareSerial.o .pioenvs/uno/FrameworkArduino/HardwareSerial0.o
 .pioenvs/uno/FrameworkArduino/HardwareSerial1.o .pioenvs/uno/FrameworkArduino/HardwareS
erial2.o .pioenvs/uno/FrameworkArduino/HardwareSerial3.o .pioenvs/uno/FrameworkArduino/I
PAddress.o .pioenvs/uno/FrameworkArduino/PluggableUSB.o .pioenvs/uno/FrameworkArduino/Pr
int.o .pioenvs/uno/FrameworkArduino/Stream.o .pioenvs/uno/FrameworkArduino/Tone.o .pioen
vs/uno/FrameworkArduino/USBCore.o .pioenvs/uno/FrameworkArduino/WInterrupts.o .pioenvs/u
no/FrameworkArduino/WMath.o .pioenvs/uno/FrameworkArduino/WString.o .pioenvs/uno/Framewo
rkArduino/_wiring_pulse.o .pioenvs/uno/FrameworkArduino/abi.o .pioenvs/uno/FrameworkArdu
ino/hooks.o .pioenvs/uno/FrameworkArduino/main.o .pioenvs/uno/FrameworkArduino/new.o .pi
oenvs/uno/FrameworkArduino/wiring.o .pioenvs/uno/FrameworkArduino/wiring_analog.o .pioen
vs/uno/FrameworkArduino/wiring_digital.o .pioenvs/uno/FrameworkArduino/wiring_pulse.o .p
ioenvs/uno/FrameworkArduino/wiring_shift.o
avr-ranlib .pioenvs/uno/libFrameworkArduino.a
avr-g++ -o .pioenvs/uno/firmware.elf -Os -mmcu=atmega328p -Wl,--gc-sections,--relax .pio
envs/uno/src/blink.o -L/Users/zach/.platformio/packages/ldscripts -L.pioenvs/uno -Wl,--s
tart-group -lm .pioenvs/uno/libFrameworkArduinoVariant.a .pioenvs/uno/libFrameworkArduin
o.a -Wl,--end-group
avr-objcopy -O ihex -R .eeprom .pioenvs/uno/firmware.elf .pioenvs/uno/firmware.hex
"avr-size" --mcu=atmega328p -C -d .pioenvs/uno/firmware.elf
AVR Memory Usage
----------------
Device: atmega328p

Program:     998 bytes (3.0% Full)
(.text + .data + .bootloader)

Data:          9 bytes (0.4% Full)
(.data + .bss + .noinit)


============================ [SUCCESS] Took 1.89 seconds ============================
Built target platformio_build

Compilation finished at Fri Dec 25 23:08:44
```

```
55: 0 -R-*compilation*                              Bot Compilation:exit [0] Projectile[
```

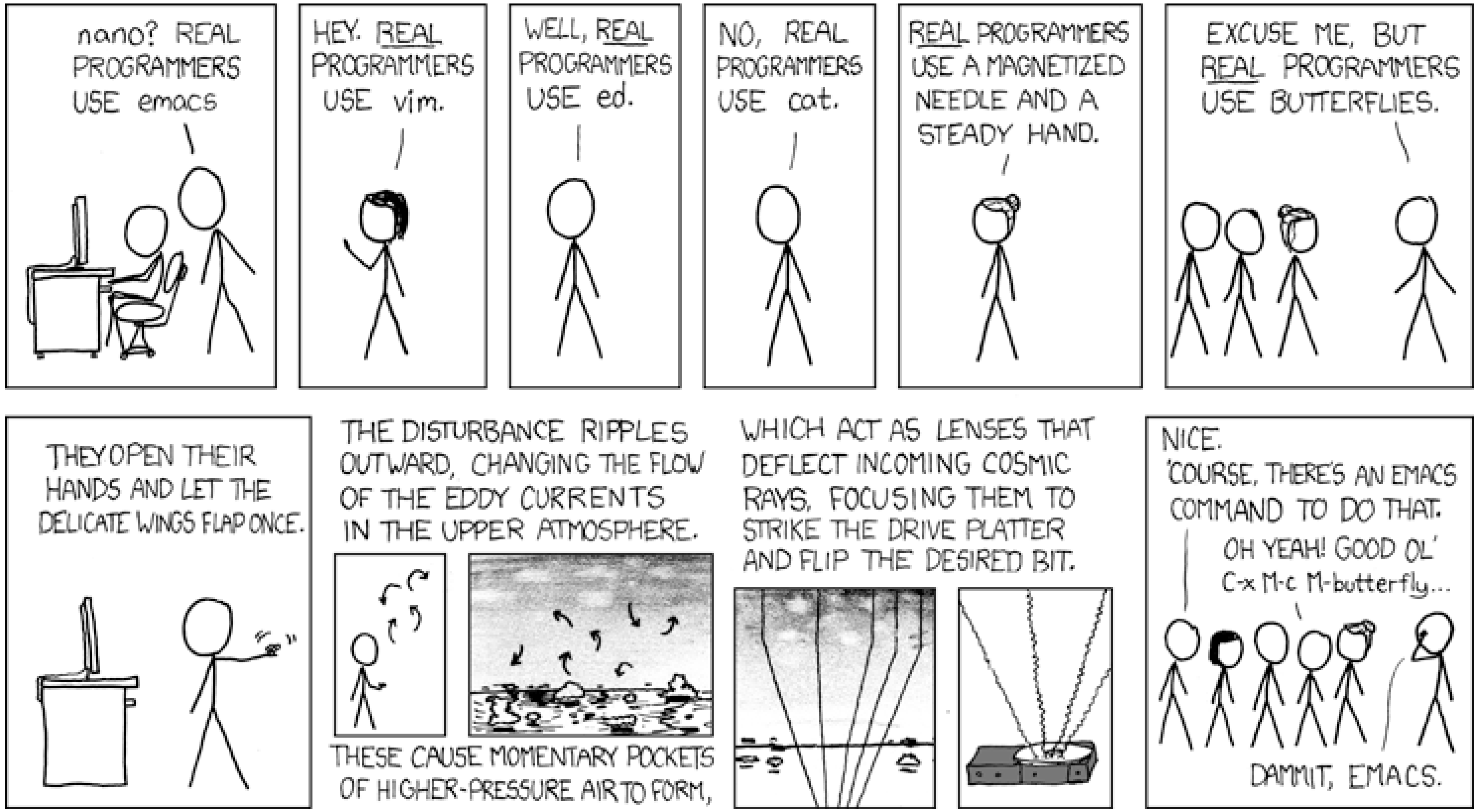
notepad-plus-plus.org

www.vim.org

www.gnu.org/software/emacs/tour/

Keep a running text-file script of all that you do. Comment it with your own mental notes so you can remember what you did and why in human language.
Pick a good **text editor** (not word processor!!) and learn to use it.

And final note for the day: keep a script of what you do!!!


nano? REAL PROGRAMMERS USE emacs
HEY. REAL PROGRAMMERS USE vim.
WELL, REAL PROGRAMMERS USE ed.
NO, REAL PROGRAMMERS USE cat.
REAL PROGRAMMERS USE A MAGNETIZED NEEDLE AND A STEADY HAND.
EXCUSE ME, BUT REAL PROGRAMMERS USE BUTTERFLIES.
THEY OPEN THEIR HANDS AND LET THE DELICATE WINGS FLAP ONCE.
THE DISTURBANCE RIPPLES OUTWARD, CHANGING THE FLOW OF THE EDDY CURRENTS IN THE UPPER ATMOSPHERE.
THESE CAUSE MOMENTARY POCKETS OF HIGHER-PRESSURE AIR TO FORM,
WHICH ACT AS LENSES THAT DEFLECT INCOMING COSMIC RAYS, FOCUSING THEM TO STRIKE THE DRIVE PLATTER AND FLIP THE DESIRED BIT.
NICE.
'COURSE, THERE'S AN EMACS COMMAND TO DO THAT.
OH YEAH! GOOD OL' C-x M-c M-butterfly...
DAMMIT, EMACS.

https://xkcd.com/378