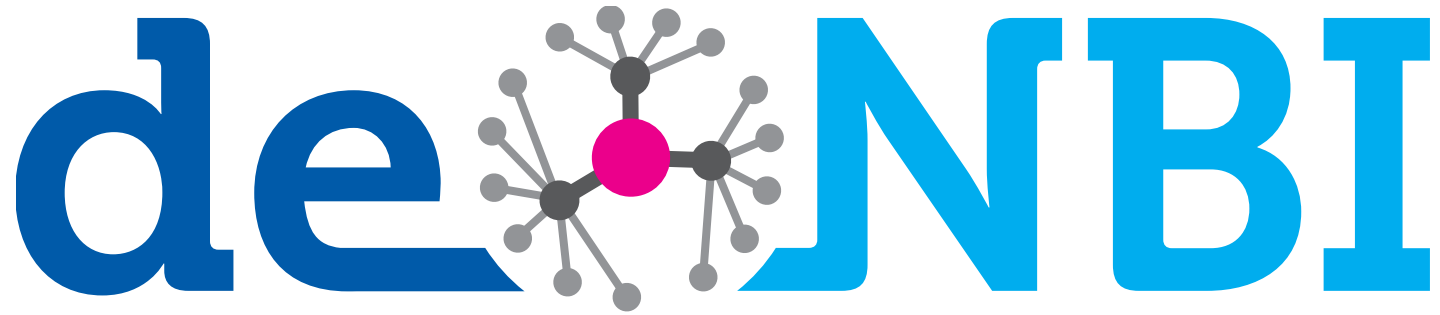


# Functional microbiome research – bioinformatics section

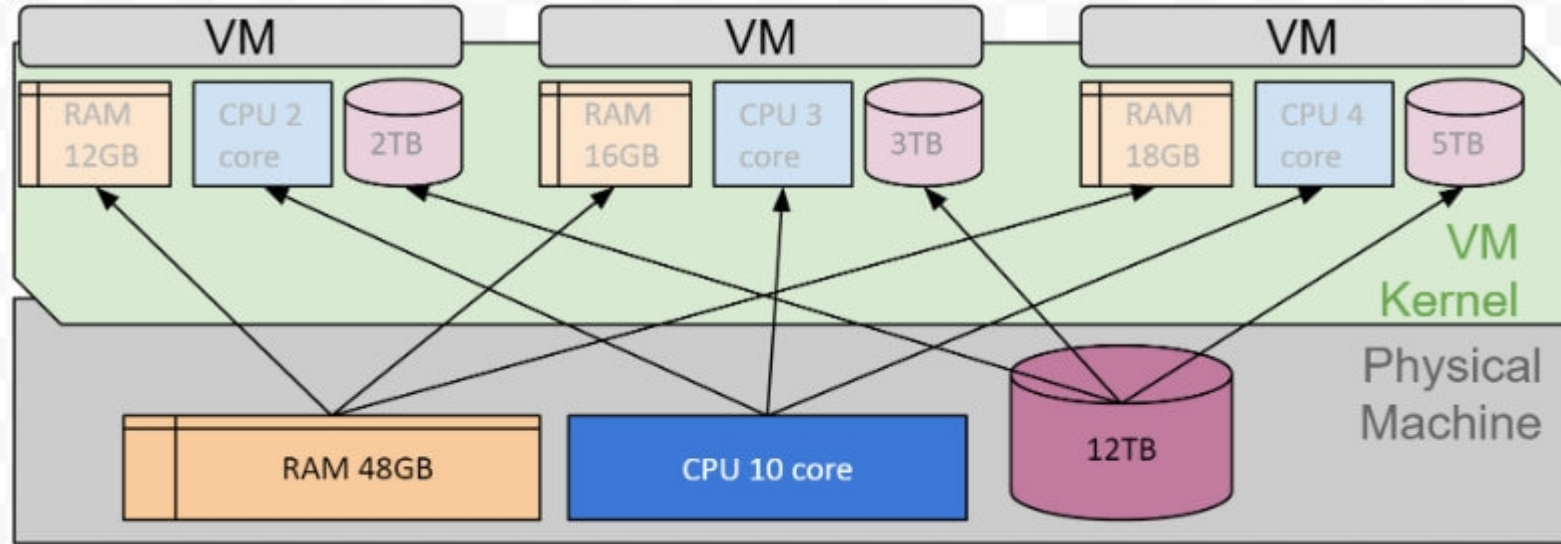
Day 1 – intro to linux



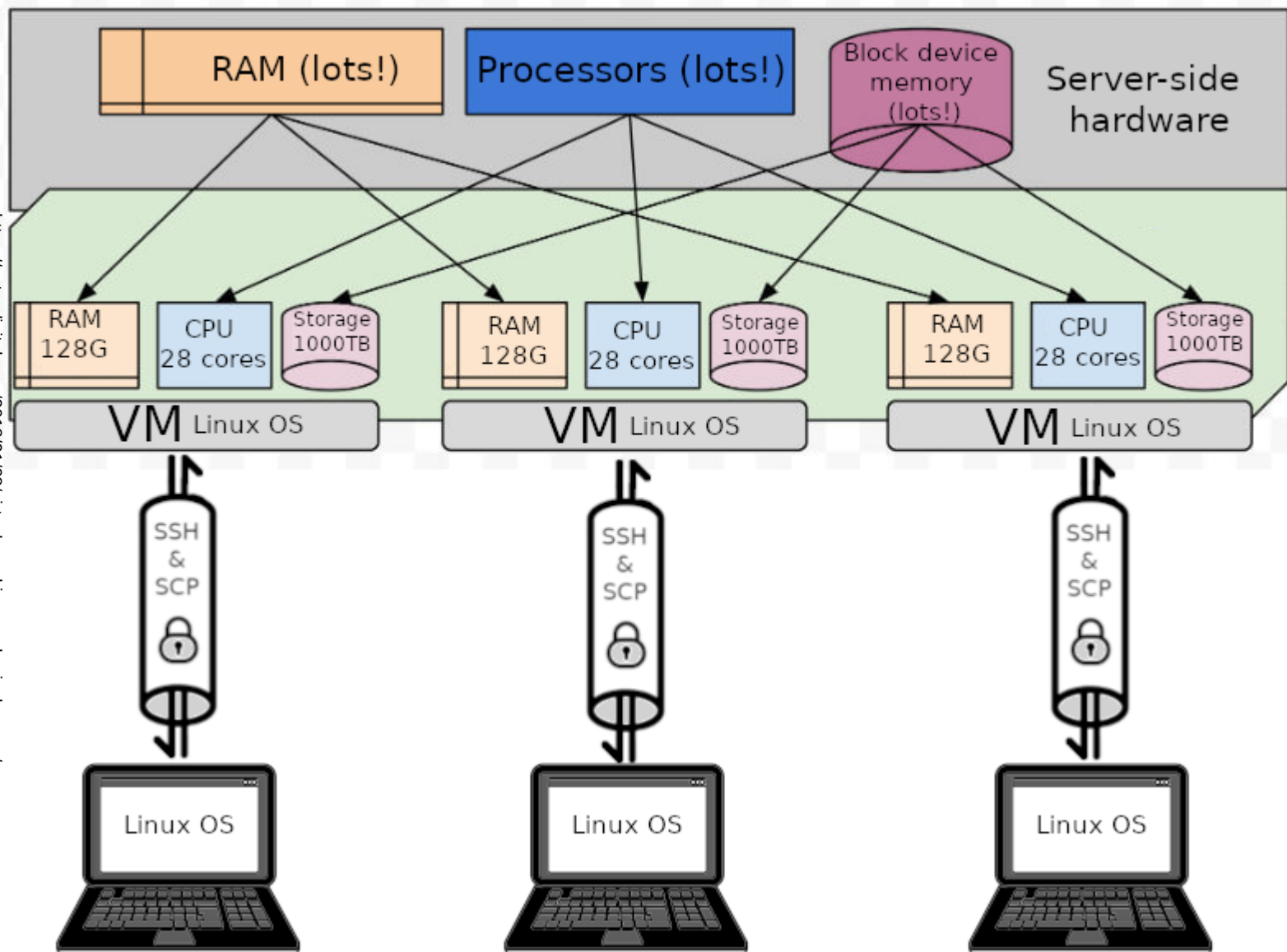
GERMAN NETWORK FOR BIOINFORMATICS INFRASTRUCTURE

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

Distributed computing and compute clusters are essential tools for scientists.



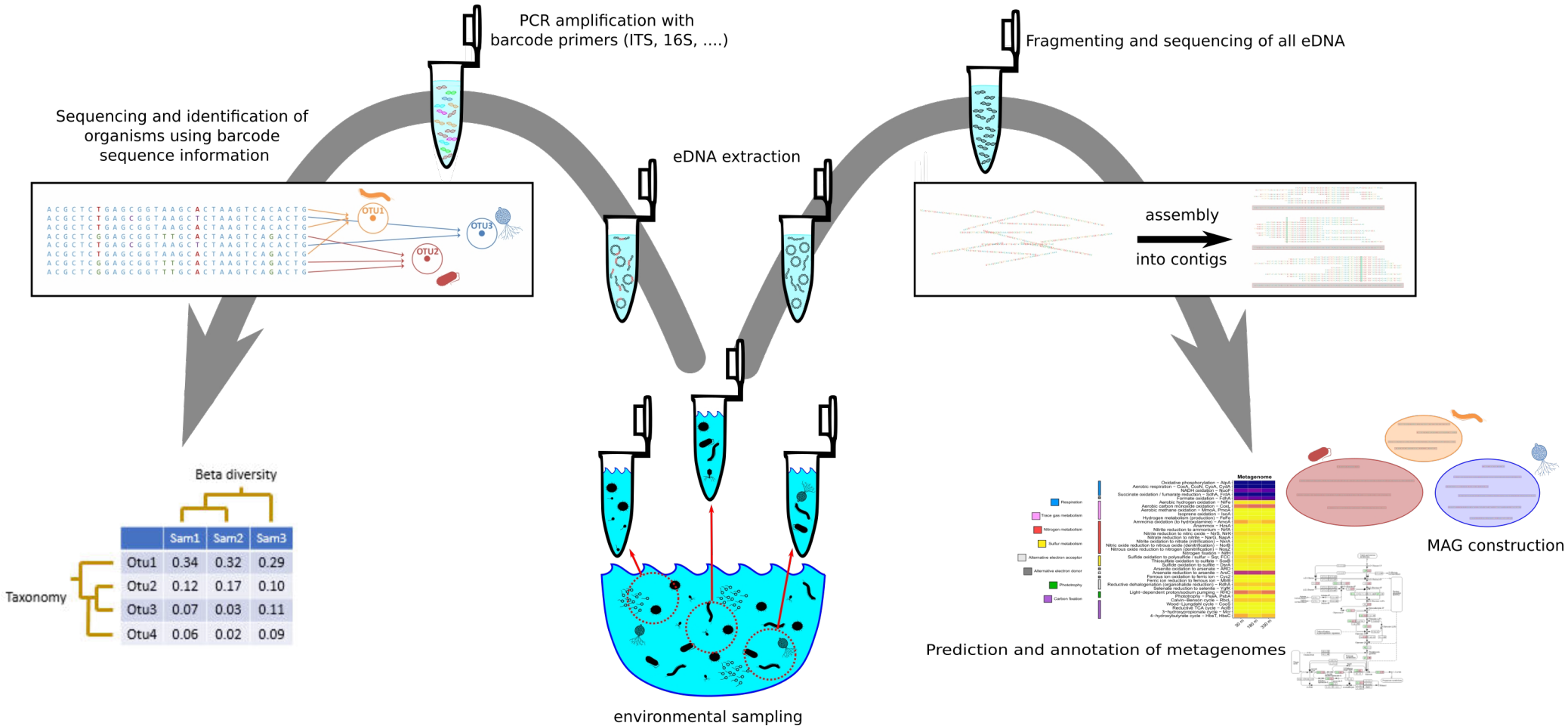
We use our local computers to talk to shared remote servers that have lots of high-powered processors, and storage and RAM.



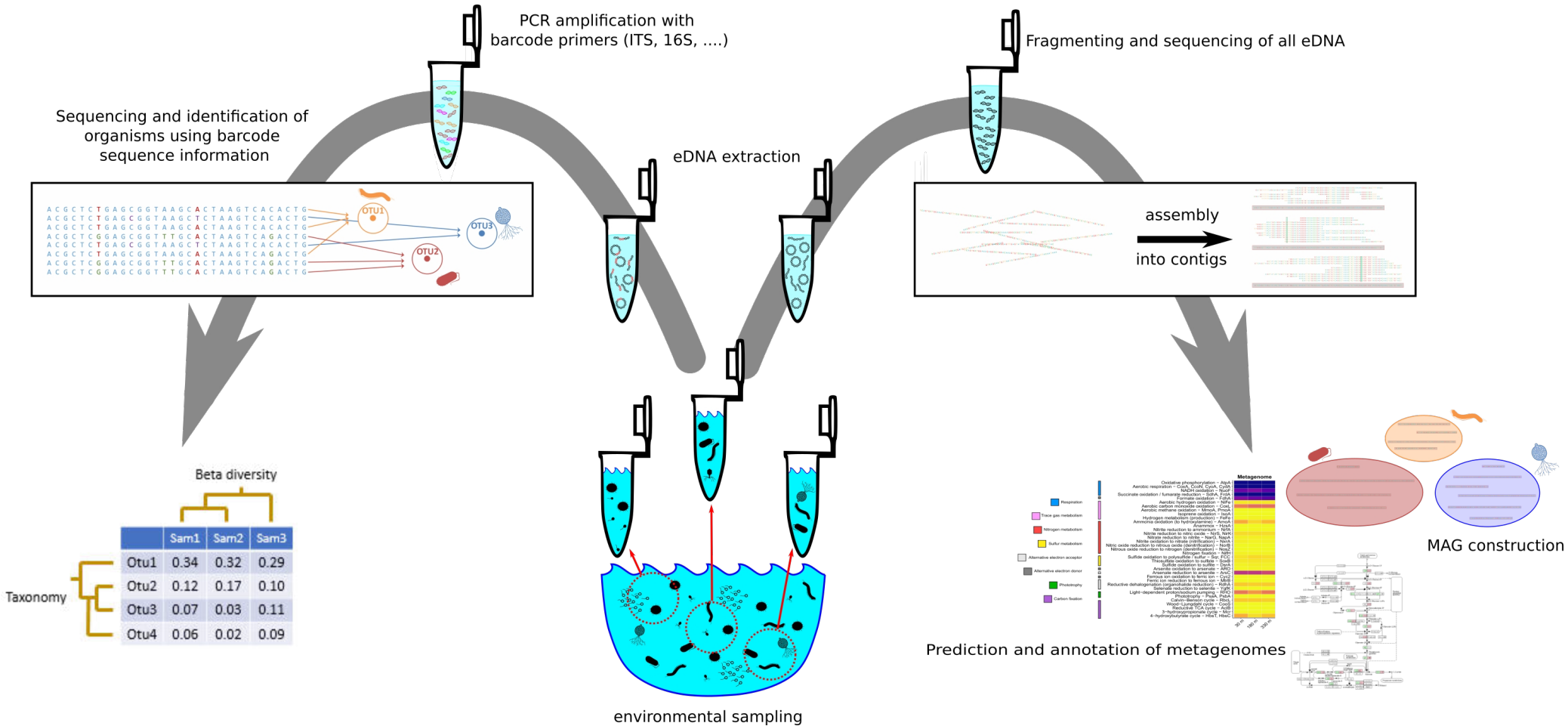
de.NBI / Universität  
Bielefeld

Us.

# Review – what do we do with DNA sequence data?

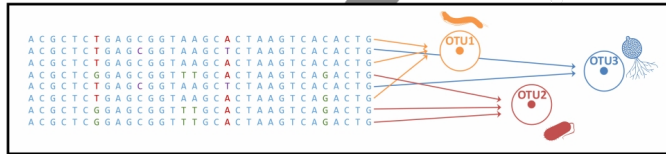


# Why do we need distributed computing to do this?



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

Sequencing and identification of organisms using barcode sequence information



Beta diversity

	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

Taxonomy

PCR amplification with barcode primers (ITS, 16S, ....)

eDNA extraction

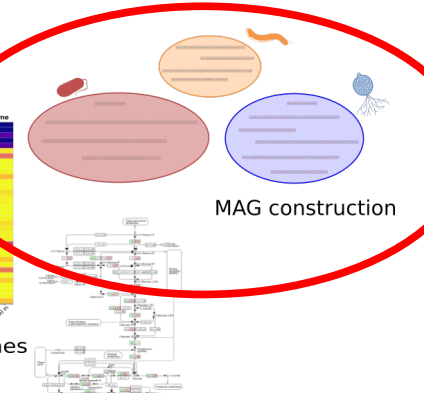
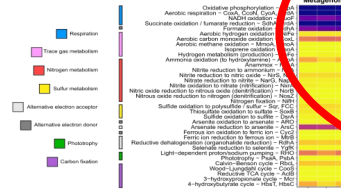
Fragmenting and sequencing of all eDNA

assembly  
into contigs

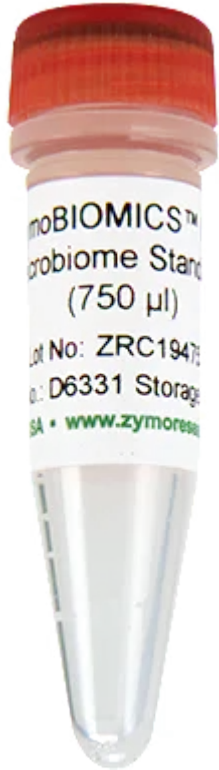
MAG construction

Prediction and annotation of metagenomes

environmental sampling



## Dataset 1: ZymoBIOMICS HMW DNA Standard.



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



## Dataset 2: Seika paper

nature|methods

BRIEF COMMUNICATION

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



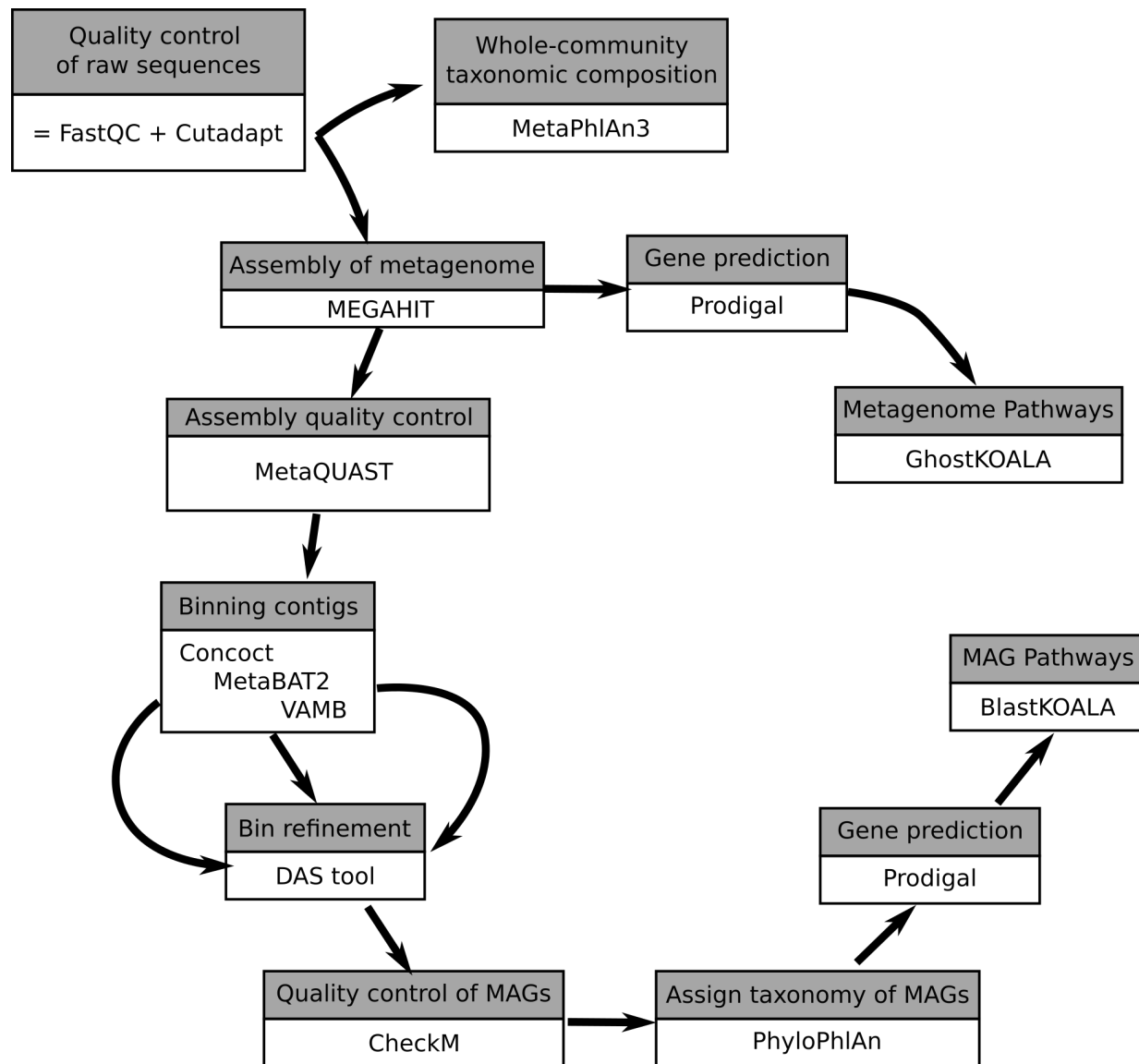
OPEN

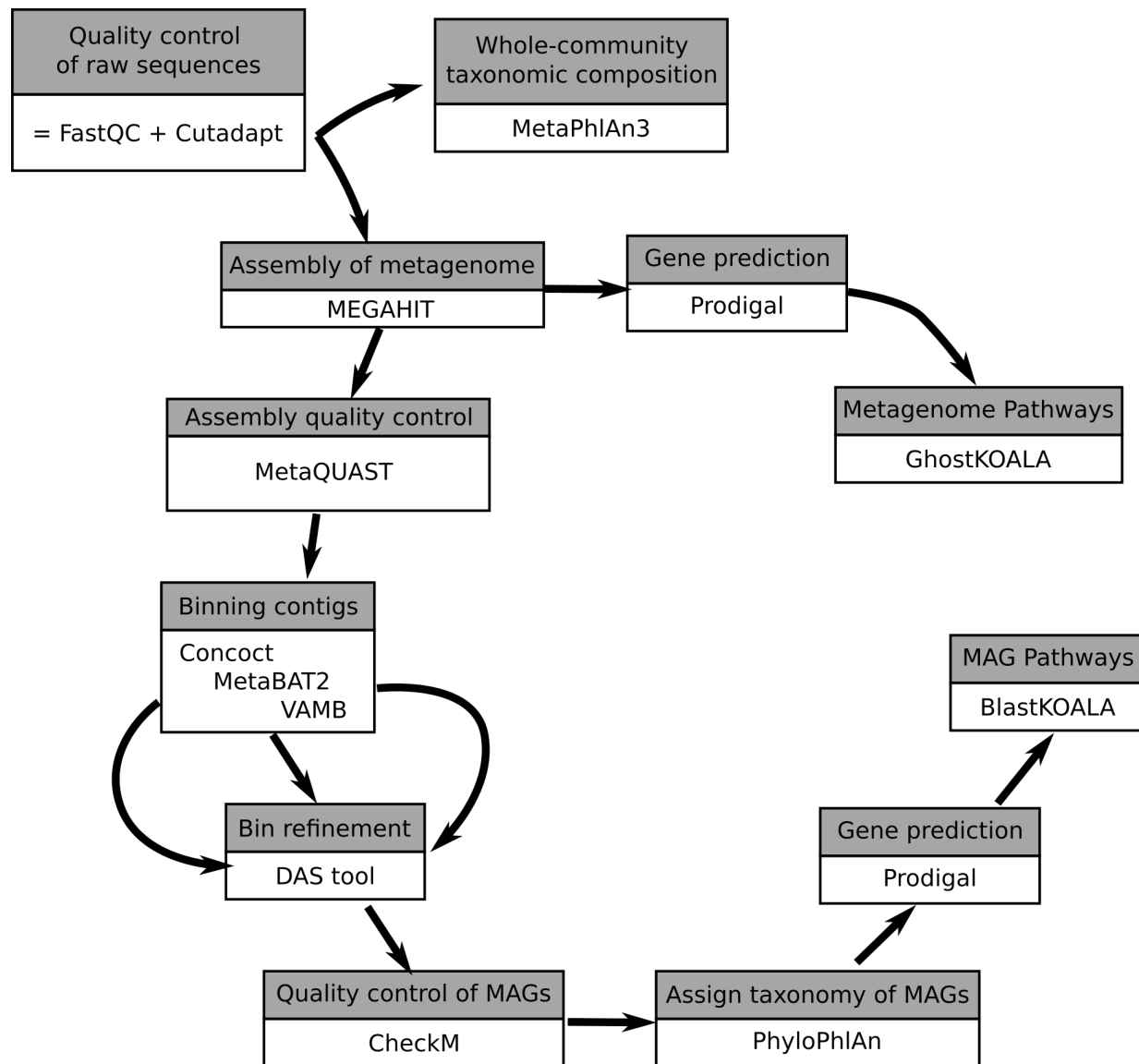
**Oxford Nanopore R10.4 long-read sequencing enables the generation of near-finished bacterial genomes from pure cultures and metagenomes without short-read or reference polishing**

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



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





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
```

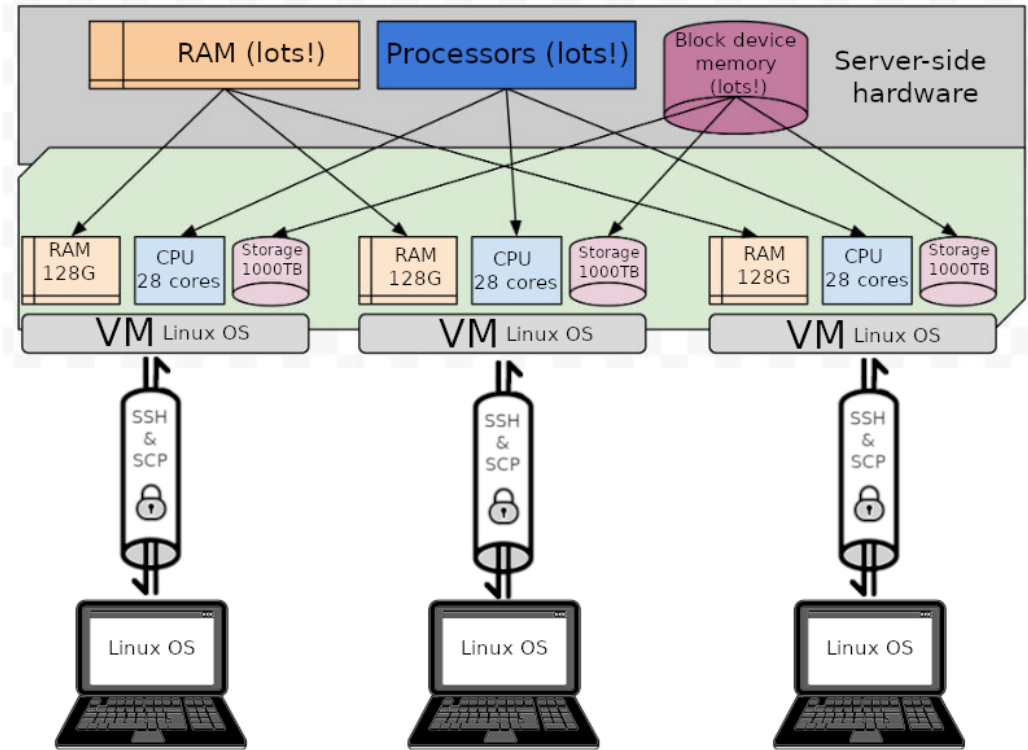
(Maybe better to follow the link in the skript PDF on page 20, or download this presentation, then copy-paste)



## First task: connect to VMs

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

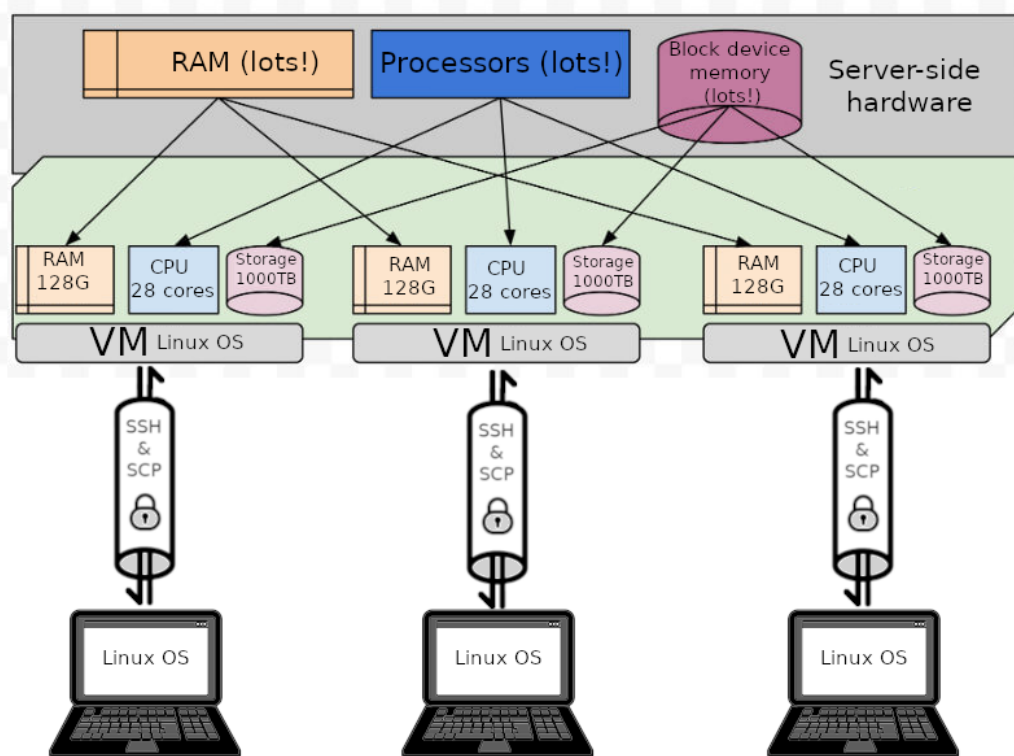
## First task: connect to VMs



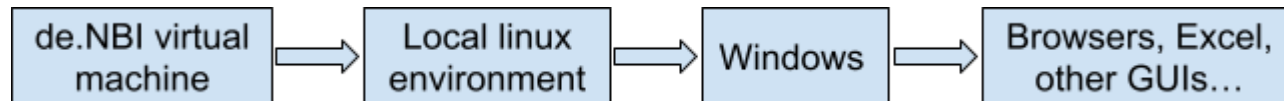
```
cd /vol/danBot
```

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

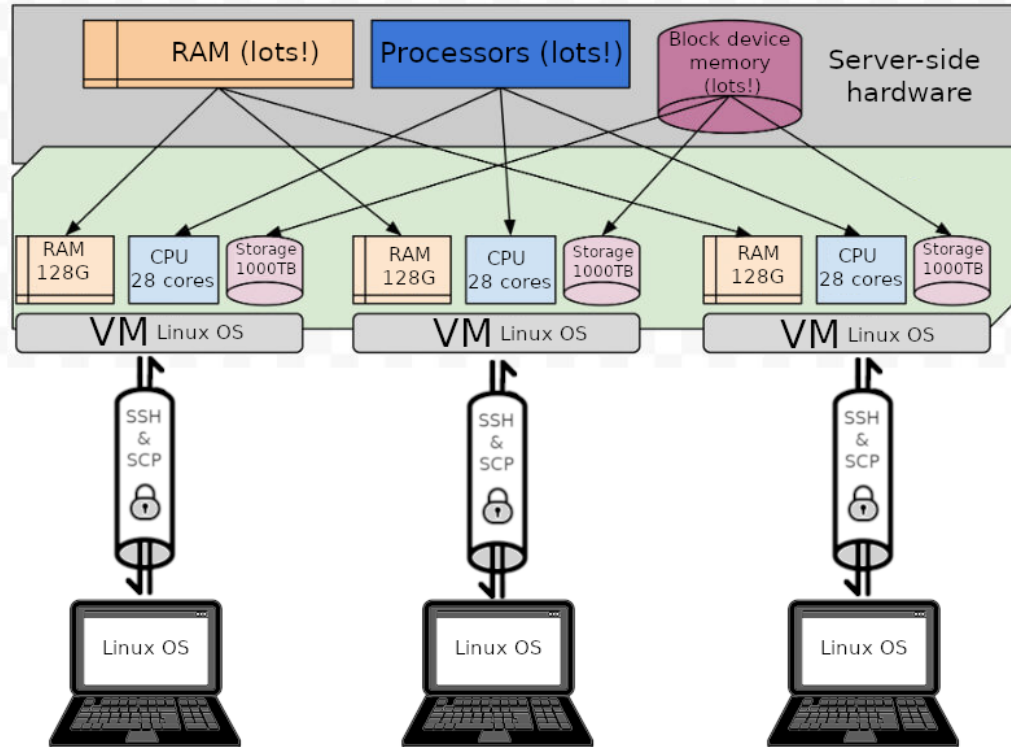
## Second task: get a file from your VM to your computer



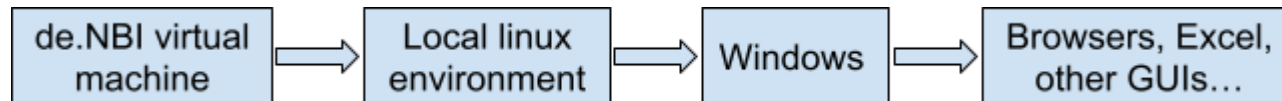
You will often need to get files from your de.NBM VM, to inspect them locally either in your linux or windows environment.



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



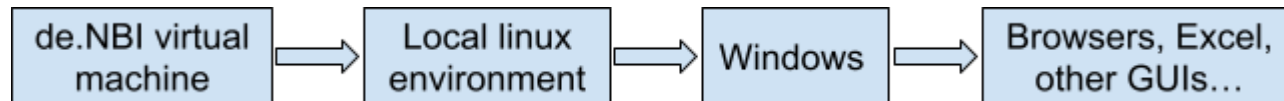
We will use the **scp** program for this.



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

```
fileOnVM="/vol/danBot/helloWorld.txt"  
mySSHkey="/home/daniel/.ssh/funmic2023.pub"  
  
scp -i /home/daniel/.ssh/funmic2023 -P 30500 ubuntu@129.70.51.6:$fileOnVM .
```

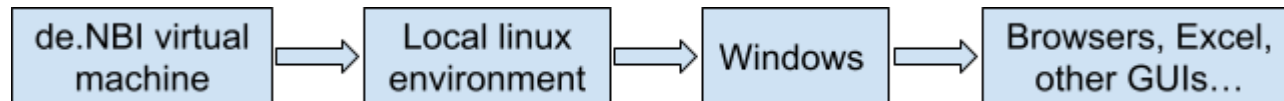
Try **scp** , to get “helloWorld.txt” from your de.NBI VM to your local machine.



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

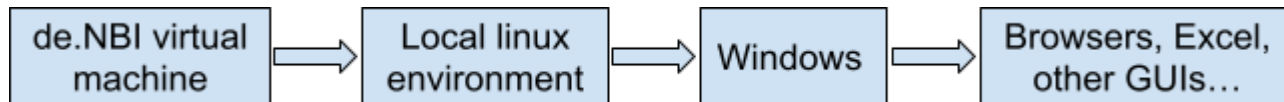
```
fileOnVM="/vol/danBot/helloWorld.txt"  
mySSHkey="/home/daniel/.ssh/funmic2023.pub"  
  
scp -i /home/daniel/.ssh/funmic2023 -P 30500 ubuntu@129.70.51.6:$fileOnVM .
```

(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.

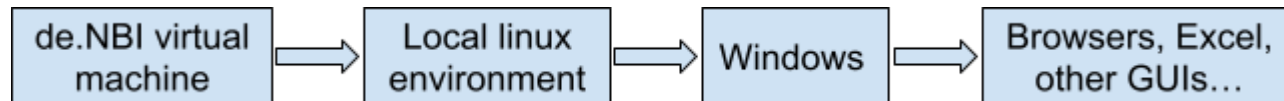


Second task: get a file from your VM to your computer

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

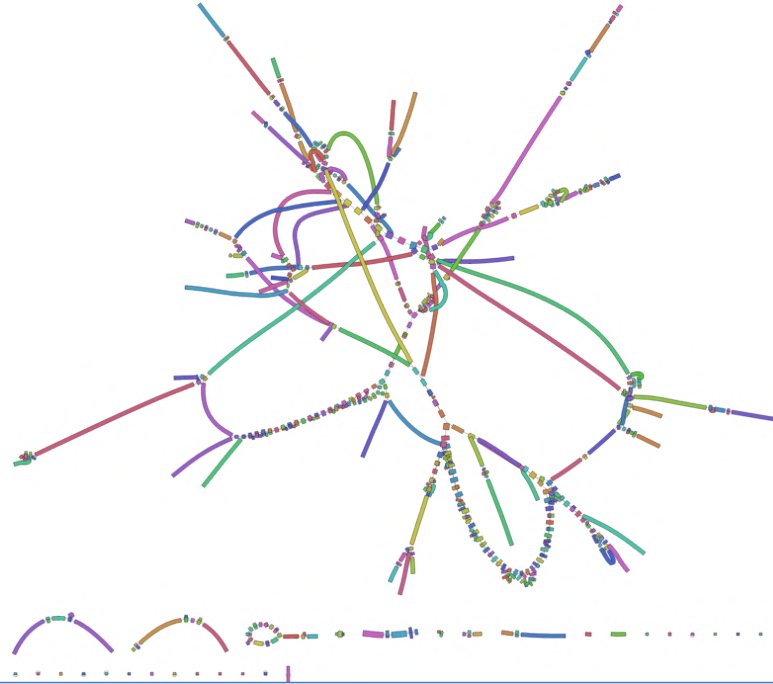
```
cd /mnt/c/
```

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.



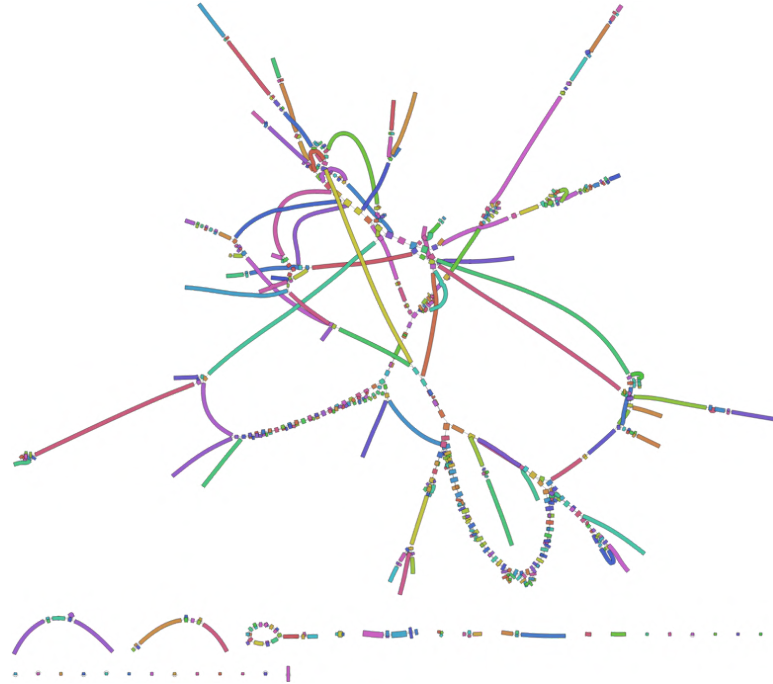


Last task of the day: start your nanopore assembly.



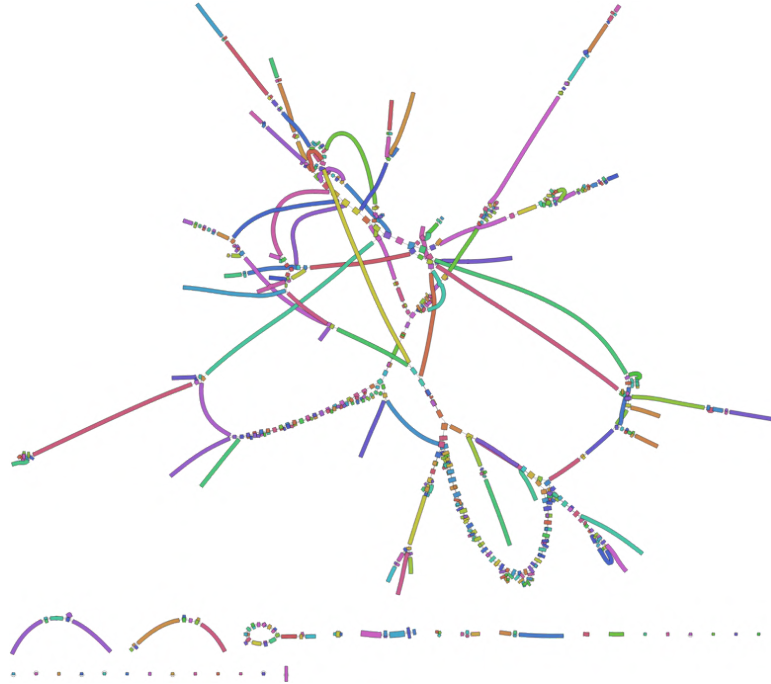
Assembly of metagenomes is one of the most computationally intensive tasks we will ask our de.NBI machines to do for us. We will review metagenome assembly a bit tomorrow, but let's get our first assembly started now, so it can do as long as it needs to.

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 wastewater sample. 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/sludge/nanopore/

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

(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
  rCPU0 : 0.0/0.0 0%
  rCPU1 : 0.0/0.0 0%
  rCPU2 : 0.0/0.0 0%
  rCPU3 : 0.0/0.0 0%
  rCPU4 : 0.0/0.0 0%
  rCPU5 : 0.0/0.0 0%
  rCPU6 : 0.0/0.0 0%
  rCPU7 : 0.0/0.0 0%
  rCPU8 : 0.0/0.0 0%
  rCPU9 : 0.0/0.0 0%
  rCPU10 : 0.0/0.0 0%
  rCPU11 : 0.0/0.0 0%
  rCPU12 : 0.0/0.0 0%
  rCPU13 : 0.0/0.0 0%
  rCPU14 : 0.0/0.3 0%
  rCPU15 : 0.0/0.0 0%
  rCPU16 : 0.0/0.0 0%
  rCPU17 : 0.0/0.0 0%
  rCPU18 : 0.0/0.0 0%
  rCPU19 : 0.0/0.0 0%
  rCPU20 : 0.0/0.0 0%
  rCPU21 : 0.0/0.0 0%
  rCPU22 : 0.0/0.0 0%
  rCPU23 : 0.0/0.0 0%
  rCPU24 : 0.0/0.0 0%
  rCPU25 : 0.0/0.0 0%
  rCPU26 : 0.0/0.0 0%
  rCPU27 : 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
```

Let's check with the **top** program.

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

```
#include <Arduino.h>

#define LED L13

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

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

```
avr-gcc -o .pioenvs/uno/FrameworkArduino/wiring_digital.o -c -g -O5 -Wall -ffunction-sections -fdata-sections -WMD -mcpu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO_VARIANT=-I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduino/wiring_pulse.c  
avr-gcc -o .pioenvs/uno/FrameworkArduino/wiring_pulse.o -c -g -O5 -Wall -ffunction-sections -fdata-sections -WMD -mcpu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO_VARIANT=-I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduino/wiring_shift.c  
avr-gcc -o .pioenvs/uno/FrameworkArduino/wiring_shift.o -c -g -O5 -Wall -ffunction-sections -fdata-sections -WMD -mcpu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO_VARIANT=-I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduino/libraries/Serial.o  
avr-ar rcs .pioenvs/uno/LibFrameworkArduino.a .pioenvs/uno/FrameworkArduino/CDC.o .pioenvs/uno/FrameworkArduino/HardwareSerial.o .pioenvs/uno/FrameworkArduino/HardwareSerial2.o .pioenvs/uno/FrameworkArduino/HardwareSerial3.o .pioenvs/uno/FrameworkArduino/Pins.o .pioenvs/uno/FrameworkArduino/Stream.o .pioenvs/uno/FrameworkArduino/Tone.o .pioenvs/uno/FrameworkArduino/USBCore.o .pioenvs/uno/FrameworkArduino/Interrupts.o .pioenvs/uno/FrameworkArduino/Math.o .pioenvs/uno/FrameworkArduino/String.o .pioenvs/uno/FrameworkArduino/wiring_pulse.o .pioenvs/uno/FrameworkArduino/wiring_analog.o .pioenvs/uno/FrameworkArduino/main.o .pioenvs/uno/FrameworkArduino/new.o .pioenvs/uno/FrameworkArduino/wiring_digital.o .pioenvs/uno/FrameworkArduino/wiring_pulse.o .pioenvs/uno/FrameworkArduino/wiring_shift.o  
avr-ranlib .pioenvs/uno/LibFrameworkArduino.a  
avr-objcopy -o .pioenvs/uno/firmware.elf -mcpu=atmega328p -Wl,-gc-sections,-relax -pie -Tlinker.ld -Btext -zcommon -zstack -znoexecstack -znow -zrelro -zseparate -lm .pioenvs/uno/LibFrameworkArduinoVariant.a .pioenvs/uno/LibFrameworkArduino.a -Wl,-end-group  
avr-objcopy -O hex -R .eeprom .pioenvs/uno/firmware.elf .pioenvs/uno/firmware.hex  
avr-size -mcpu=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 [=====]
```

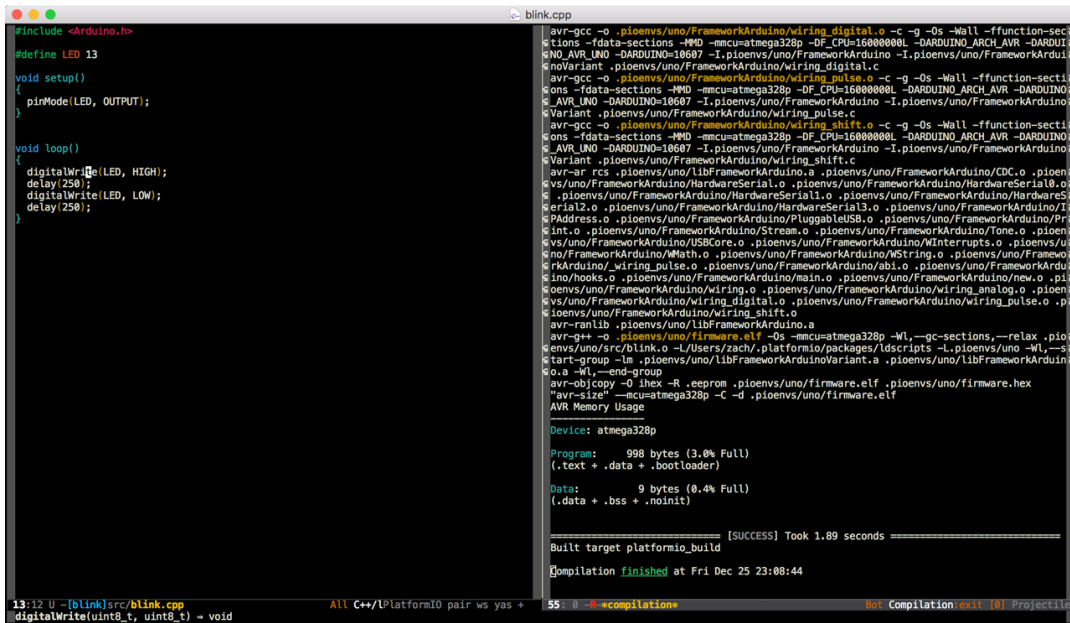
```
Built target platformio_build
```

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

```
13:12 U ~[blink]src\blink.cpp All C++\PlatformIO pair ws yas + 55 0 0 compilation time Not Compilation exit [N] Project file
```

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!!!



```
#include <Arduino.h>

#define LED 13

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

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

```
avr-gcc -o .pioenvs/uno/FrameworkArduino/wiring_digital.o -c -g -Os -Wall -ffunction-sections -fdata-sections -mcpu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO_AVR_UNO -DARDUINO=10607 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduinoVariant -I.pioenvs/uno/FrameworkArduino/wiring_digital.c
avr-gcc -o .pioenvs/uno/FrameworkArduino/wiring_pulse.o -c -g -Os -Wall -ffunction-sections -fdata-sections -mcpu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO_AVR_UNO -DARDUINO=10607 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduinoVariant -I.pioenvs/uno/FrameworkArduino/wiring_pulse.c
avr-gcc -o .pioenvs/uno/FrameworkArduino/wiring_shift.o -c -g -Os -Wall -ffunction-sections -fdata-sections -mcpu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO_AVR_UNO -DARDUINO=10607 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduinoVariant -I.pioenvs/uno/FrameworkArduino/wiring_shift.c
avr-ar rcs .pioenvs/uno/libFrameworkArduino.a .pioenvs/uno/FrameworkArduino/CDC.o .pioenvs/uno/FrameworkArduino/HardwareSerial.o .pioenvs/uno/FrameworkArduino/HardwareSerial0.o .pioenvs/uno/FrameworkArduino/HardwareSerial1.o .pioenvs/uno/FrameworkArduino/HardwareSerial2.o .pioenvs/uno/FrameworkArduino/HardwareSerial3.o .pioenvs/uno/FrameworkArduino/EEPROM.o .pioenvs/uno/FrameworkArduino/PluggableUSB.o .pioenvs/uno/FrameworkArduino/Print.o .pioenvs/uno/FrameworkArduino/Stream.o .pioenvs/uno/FrameworkArduino/Tone.o .pioenvs/uno/FrameworkArduino/USBCore.o .pioenvs/uno/FrameworkArduino/Interrupts.o .pioenvs/uno/FrameworkArduino/Math.o .pioenvs/uno/FrameworkArduino/HString.o .pioenvs/uno/FrameworkArduino/wiring_pulse.o .pioenvs/uno/FrameworkArduino/abi.o .pioenvs/uno/FrameworkArduino/main.o .pioenvs/uno/FrameworkArduino/new.o .pioenvs/uno/FrameworkArduino/wiring.o .pioenvs/uno/FrameworkArduino/wiring_analog.o .pioenvs/uno/FrameworkArduino/wiring_digital.o .pioenvs/uno/FrameworkArduino/wiring_pulse.o .pioenvs/uno/FrameworkArduino/wiring_shift.o
avr-ranlib .pioenvs/uno/libFrameworkArduino.a
avr-g++ -o .pioenvs/uno/firmware.elf -Os -mcpu=atmega328p -Wl,--gc-sections,-relax -I.pioenvs/uno/src/libraries/USB/usb/usb_host/platformio/packages/ldscripts -I.pioenvs/uno -Wl,--start-group -lpioenvs/uno/libFrameworkArduinoVariant.a .pioenvs/uno/libFrameworkArduino.a -Wl,--end-group
avr-objcopy -O ihex -R .eeprom .pioenvs/uno/firmware.elf .pioenvs/uno/firmware.hex
avr-size -mcpu=atmega328p -C -o .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 1.09 seconds

Built target platformio_build
Compilation finished at Fri Dec 25 23:08:44
```

13:12 U - [blink]src\blink.cpp ALL C++\L\platformio pair ws yas + 55 0 0 - compilations Not Compilation:exit [0] Project:U

digitalWrite(uint8\_t, uint8\_t) = void

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 L3  
  
void setup()  
{  
    pinMode(LED, OUTPUT);  
  
  
}  
  
void loop()  
{  
    digitalWrite(LED, HIGH);  
    delay(250);  
    digitalWrite(LED, LOW);  
    delay(250);  
}
```

```
avr-gcc -o .pioenvs/uno/FrameworkArduino/wiring_digital.o -c -g -Os -Wall -ffunction-sections -fdata-sections -WMD -mcpu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO_VARIANT=-I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduino\nVariant_1.pioenvs/uno/FrameworkArduino/wiring_pulse.c  
avr-gcc -o .pioenvs/uno/FrameworkArduino/wiring_pulse.o -c -g -Os -Wall -ffunction-sections -fdata-sections -WMD -mcpu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO_VARIANT=-I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduino\nVariant_1.pioenvs/uno/FrameworkArduino/wiring_shift.c  
avr-gcc -o .pioenvs/uno/FrameworkArduino/wiring_shift.o -c -g -Os -Wall -ffunction-sections -fdata-sections -WMD -mcpu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO_VARIANT=-I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduino\nVariant_1.pioenvs/uno/FrameworkArduino/wiring_shift.o  
avr-ar rcs .pioenvs/uno/LibFrameworkArduina.a .pioenvs/uno/FrameworkArduino/CDC.o .pioenvs/uno/FrameworkArduino/HardwareSerial.o .pioenvs/uno/FrameworkArduino/HardwareSerial2.o .pioenvs/uno/FrameworkArduino/HardwareSerial3.o .pioenvs/uno/FrameworkArduino/Pins.o .pioenvs/uno/FrameworkArduino/Tone.o .pioenvs/uno/FrameworkArduino/USBCore.o .pioenvs/uno/FrameworkArduino/WInterrupts.o .pioenvs/uno/FrameworkArduino/Math.o .pioenvs/uno/FrameworkArduino/String.o .pioenvs/uno/FrameworkArduino/_wiring_pulse.o .pioenvs/uno/FrameworkArduino/_wiring_shift.o .pioenvs/uno/FrameworkArduino/main.o .pioenvs/uno/FrameworkArduino/new.o .pioenvs/uno/FrameworkArduino/wiring_analog.o .pioenvs/uno/FrameworkArduino/wiring_digital.o .pioenvs/uno/FrameworkArduino/wiring_pulse.o .pioenvs/uno/FrameworkArduino/wiring_shift.o  
avr-ranlib .pioenvs/uno/LibFrameworkArduina.a  
avr-objcopy -O elf32-i586 -mcpu=atmega328p -WL --gc-sections,-relax -pie -x .pioenvs/uno/SrcBlink.cpp.o -luserspace.plainfirmwareelf -lpthread -ldspnewt -lm -lcrtstart-group -la .pioenvs/uno/LibFrameworkArduinavariant.a .pioenvs/uno/LibFrameworkArduinoc.a -Wl,-end-group  
avr-objcopy -O hex -R .eeprom .pioenvs/uno/firmware.elf .pioenvs/uno/firmware.hex  
avr-size -mcpu=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
```

```
13:12 U ~[blink]src\blink.cpp All C++\PlatformIO pair ws yas + S5 R compilation Not Compilation exit IN Projectile  
digitalWrite(uint_t, uint8_t) = void
```



notepad-plus-plus.org



www.vim.org



# GNU Emacs

[www.gnu.org/software/emacs/tour/](http://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!!!

