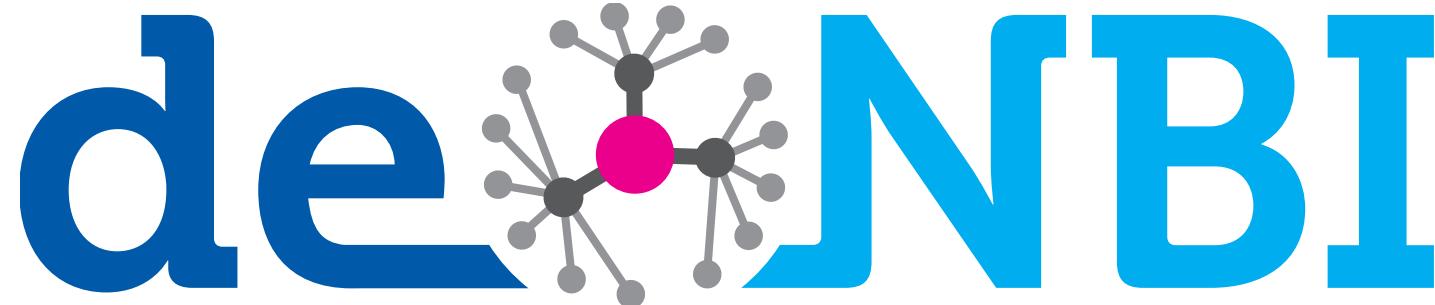


Functional microbiome research – bioinformatics section

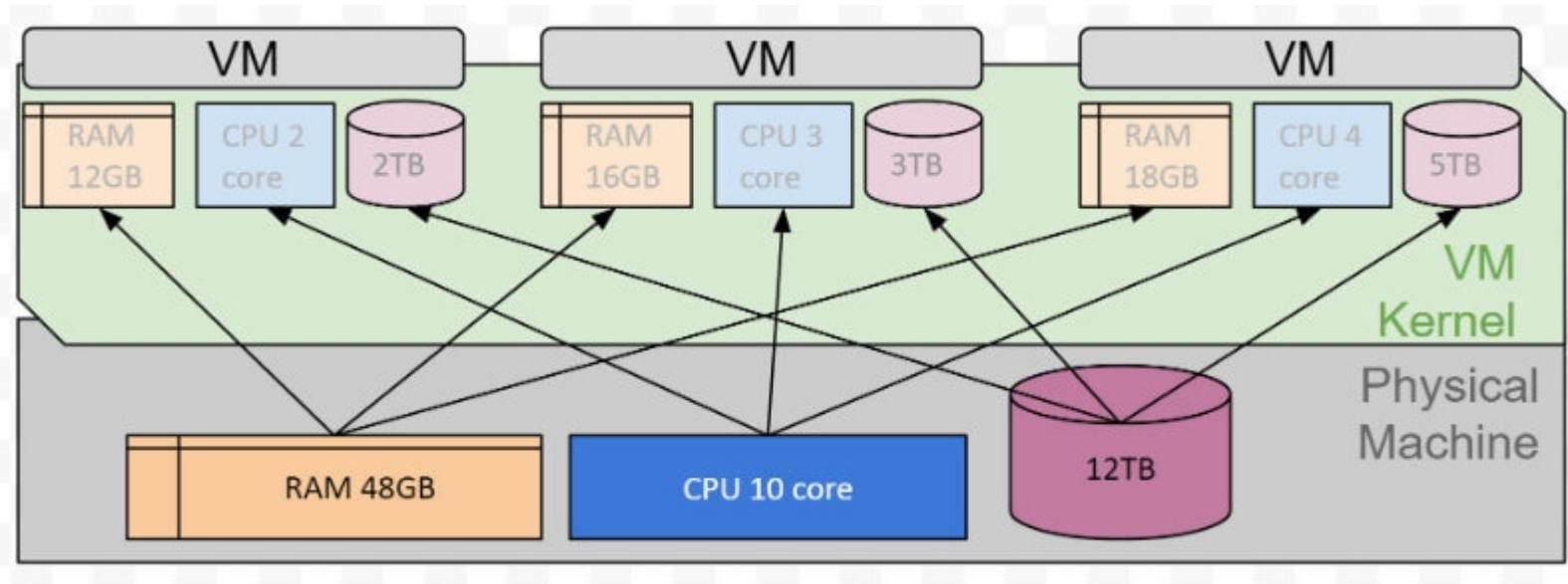
Session 1 – intro to linux



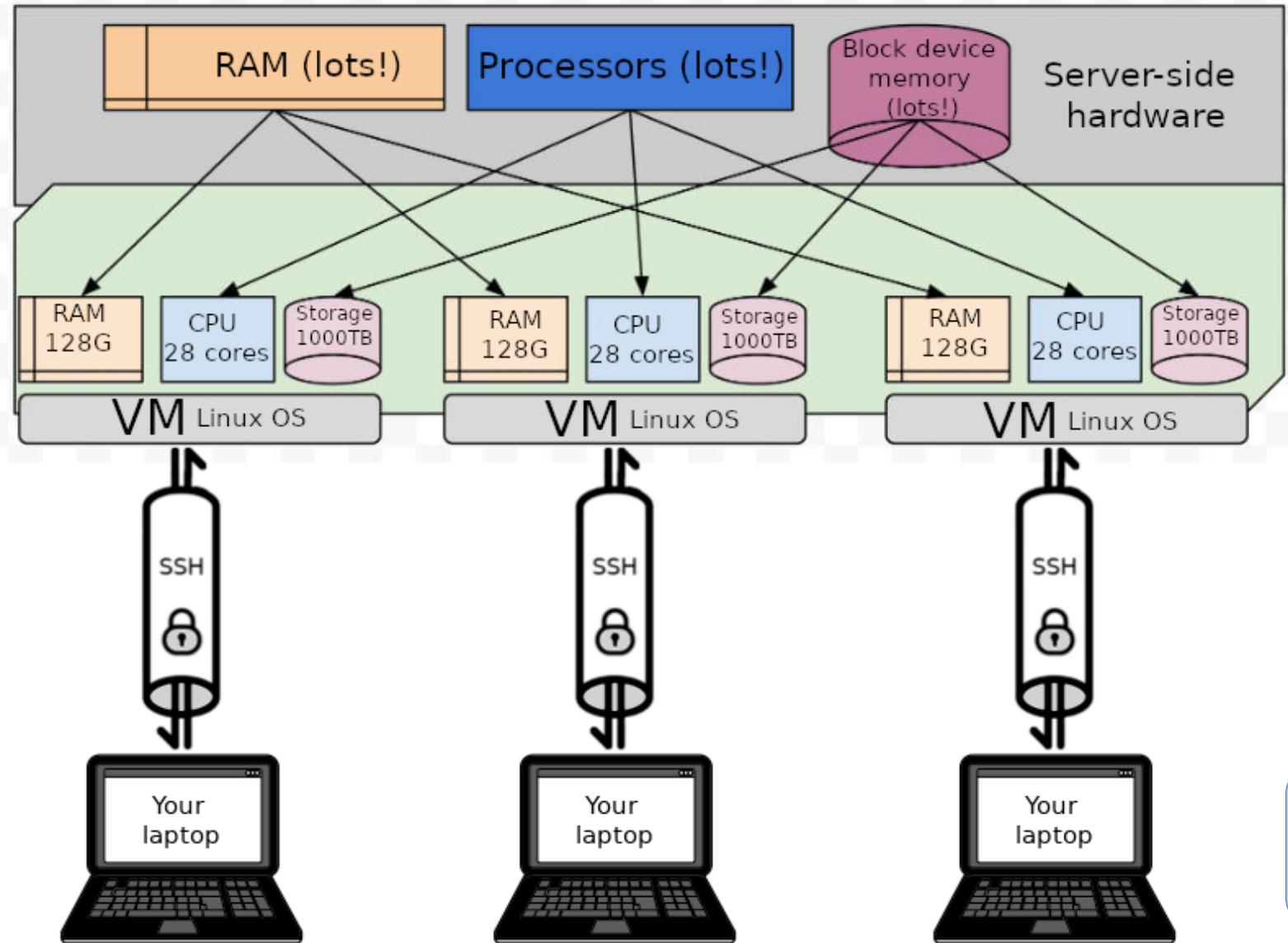
GERMAN NETWORK FOR BIOINFORMATICS INFRASTRUCTURE

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

Distributed computing and compute clusters are essential tools for scientists.



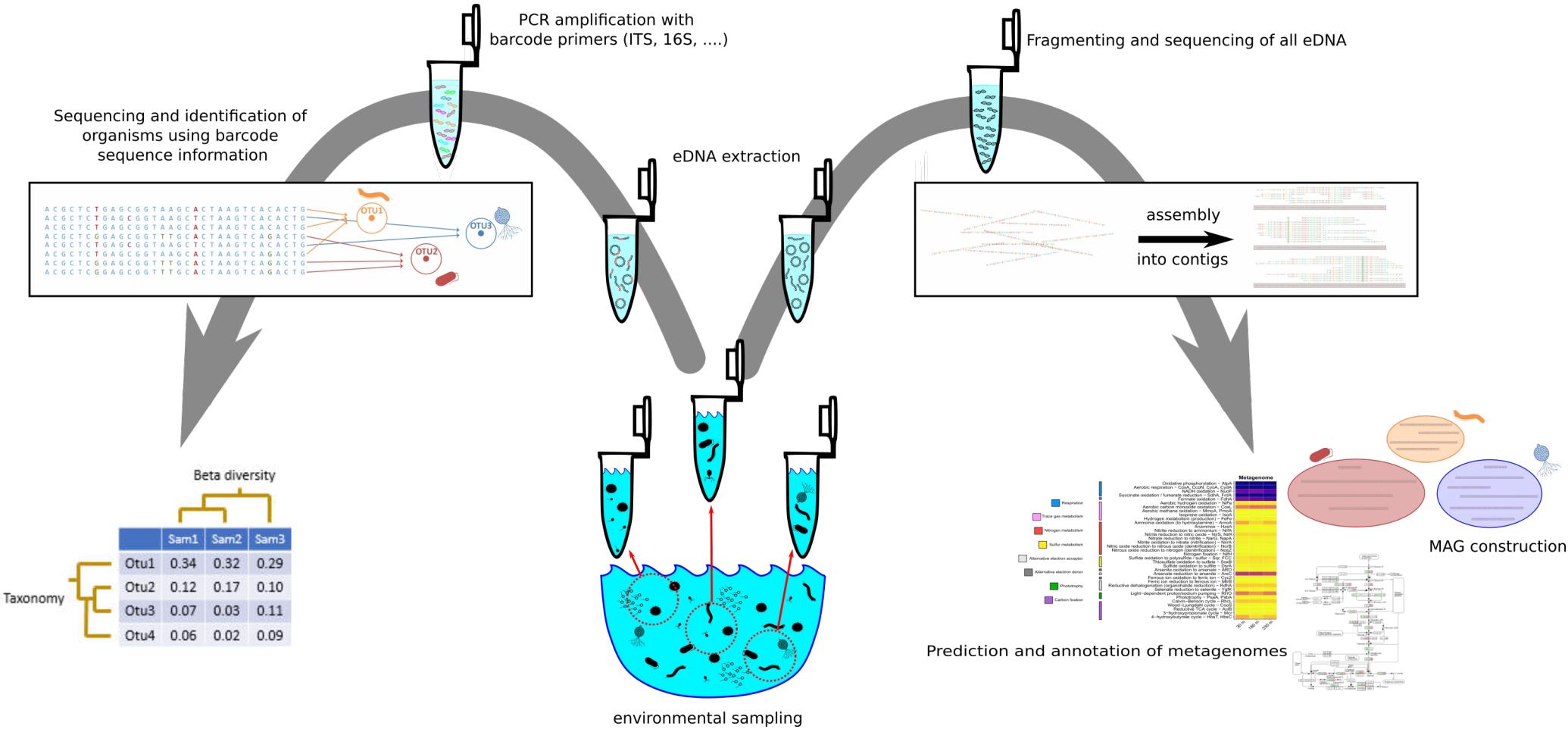
We use our local computers to talk to shared remote servers that have lots of high-powered processors, and storage and RAM.



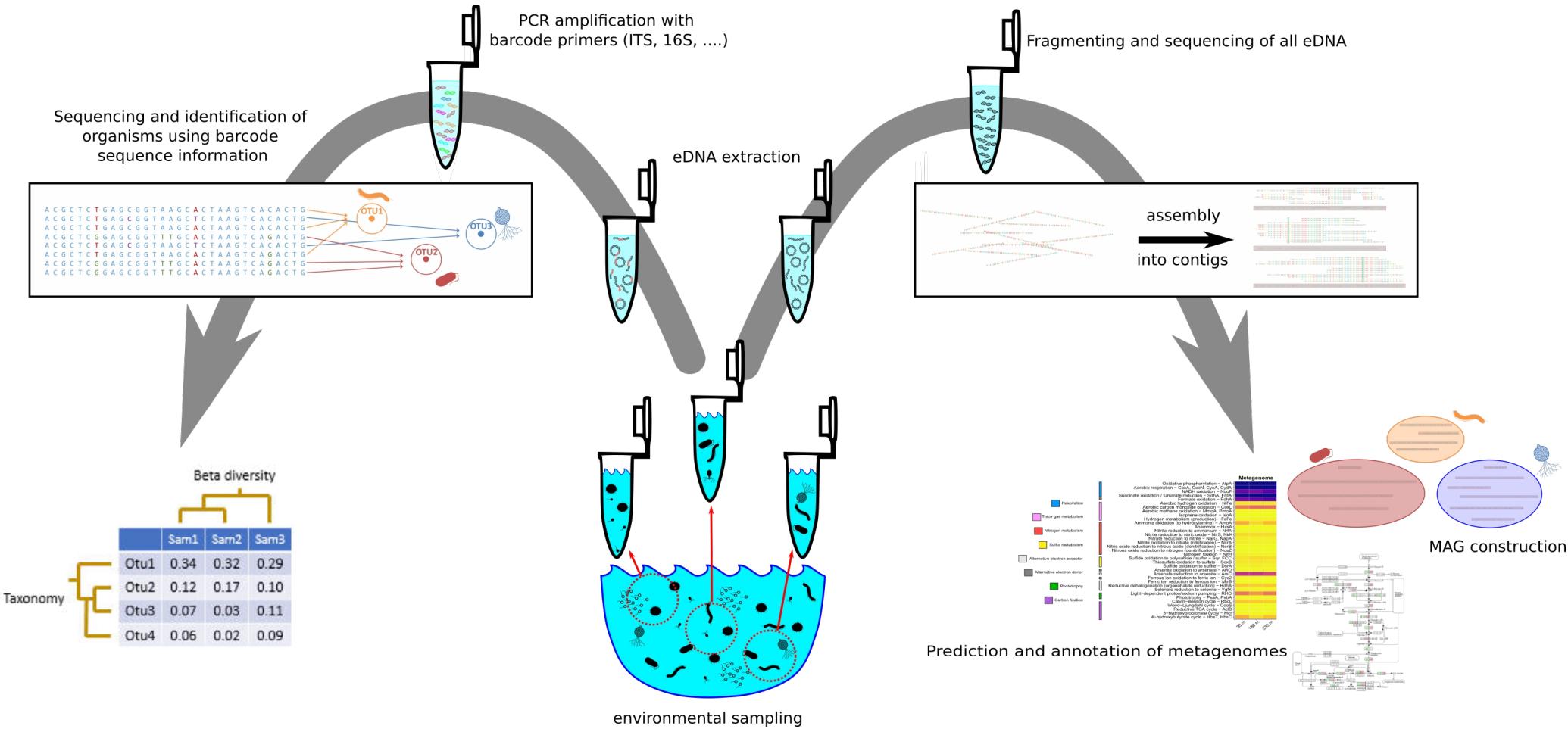
de.NBI / Universität
Bielefeld

Us.

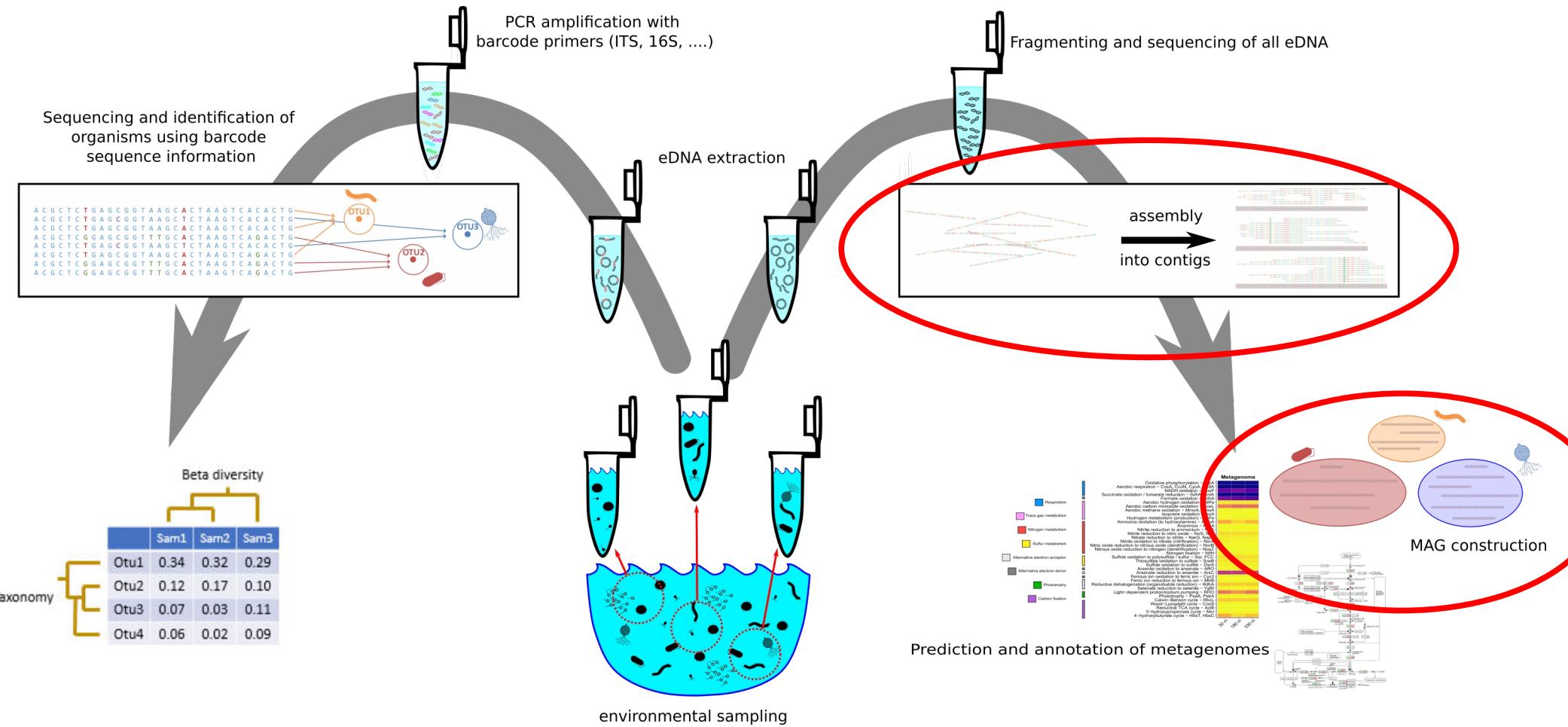
Review – what do we do with DNA sequence data?



Why do we need distributed computing to do this?



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



Dataset 2: Kelp decomposition

LIMNOLOGY
and
OCEANOGRAPHY



Limnol. Oceanogr. 65, 2020, 3066–3084

© 2020 The Authors. *Limnology and Oceanography* published by Wiley Periodicals LLC on behalf of Association for the Sciences of Limnology and Oceanography.
doi: 10.1002/limo.11574

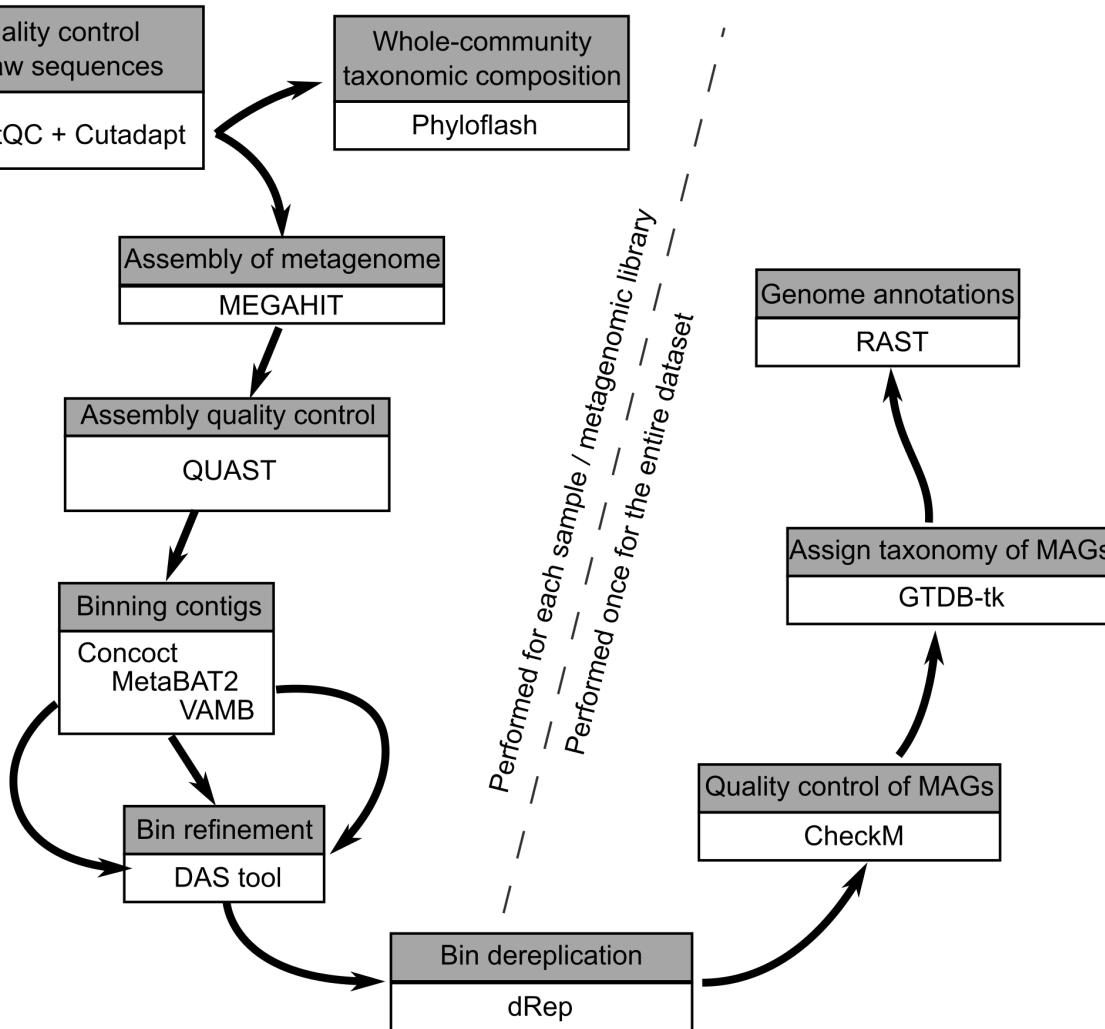
Kelp deposition changes mineralization pathways and microbial communities in a sandy beach

Marit R. van Erk ^{1,*} Dimitri V. Meier,^{1,a} Timothy Ferdelman,¹ Jens Harder,¹ Ingeborg Bussmann,² Dirk de Beer¹

¹Max Planck Institute for Marine Microbiology, Bremen, Germany

²Alfred Wegener Institute, Helmholtz Centre for Polar and Marine Research, Helgoland, Germany





We will use various scientific software packages to assemble Metagenome-Assembled-Genomes (MAGs) from these sequence data.

Quality control
of raw sequences

= FastQC + Cutadapt

Whole-community
taxonomic composition

Phyloflash

Assembly of metagenome

MEGAHIT

Assembly quality control

QUAST

Binning contigs

Concoct
MetaBAT2
VAMB

Bin refinement

DAS tool

Bin derePLICATION

dRep

Performed for each sample / metagenomic library

Genome annotations

RAST

Assign taxonomy of MAGs

GTDB-tk

Quality control of MAGs

CheckM

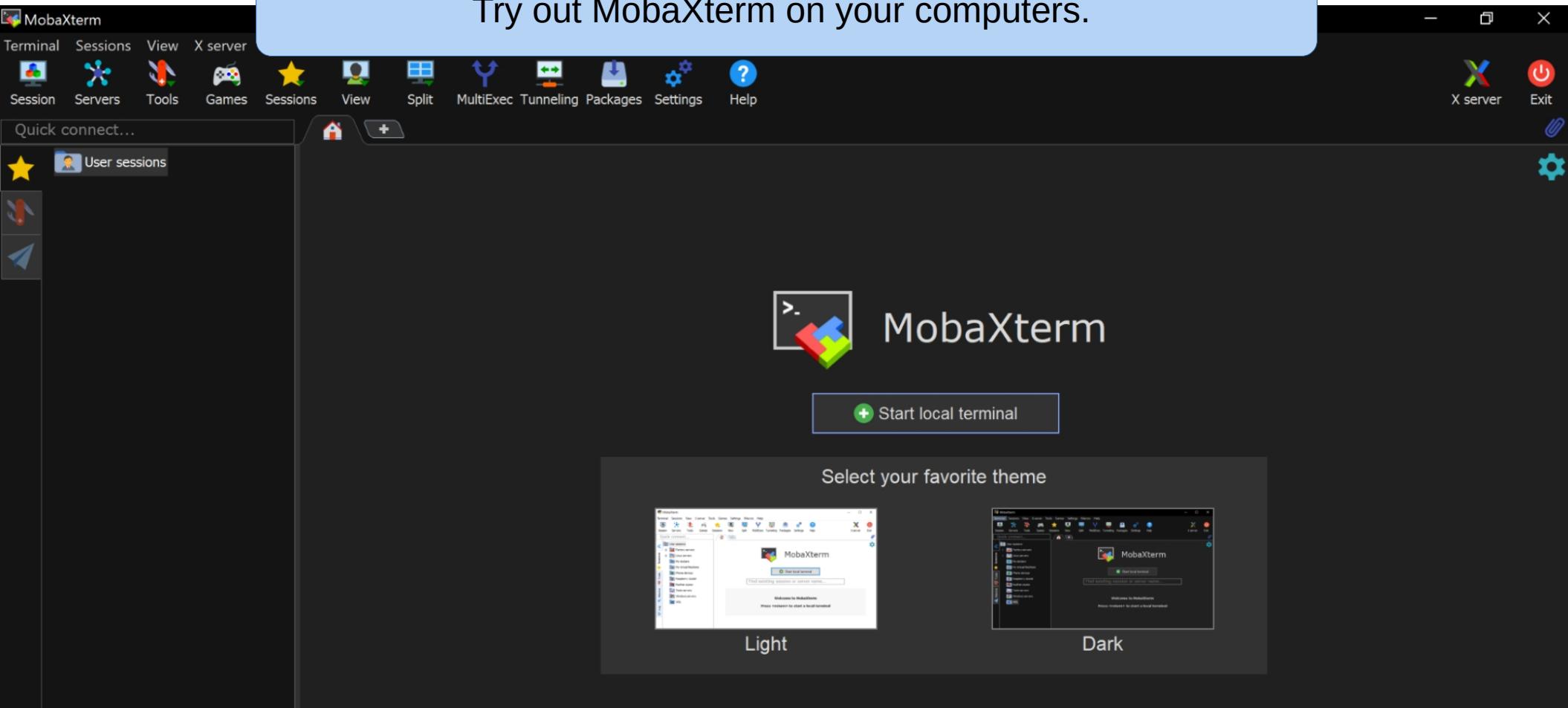
You will use these tools to
discover the ecological story
in the Kelp decomposition dataset.



We'll keep track of what we do in scripts that will be kept in an online
github repository.

1. Scientific computing and Linux (today!)
2. Metagenomic methods
3. Metabarcoding methods
4. Multivariate SIP analysis

Try out MobaXterm on your computers.



Let's check everyone's installation. Try starting a local terminal.

Now let's try to connect to our de.NBI virtual machines.

Sessions View X server

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

X server Exit

Quick connect... Session settings

SSH Telnet Rsh Xdmcp RDP VNC SFTP Serial File Shell Browser Mosh Aws S3 WSL

Choose a session type...

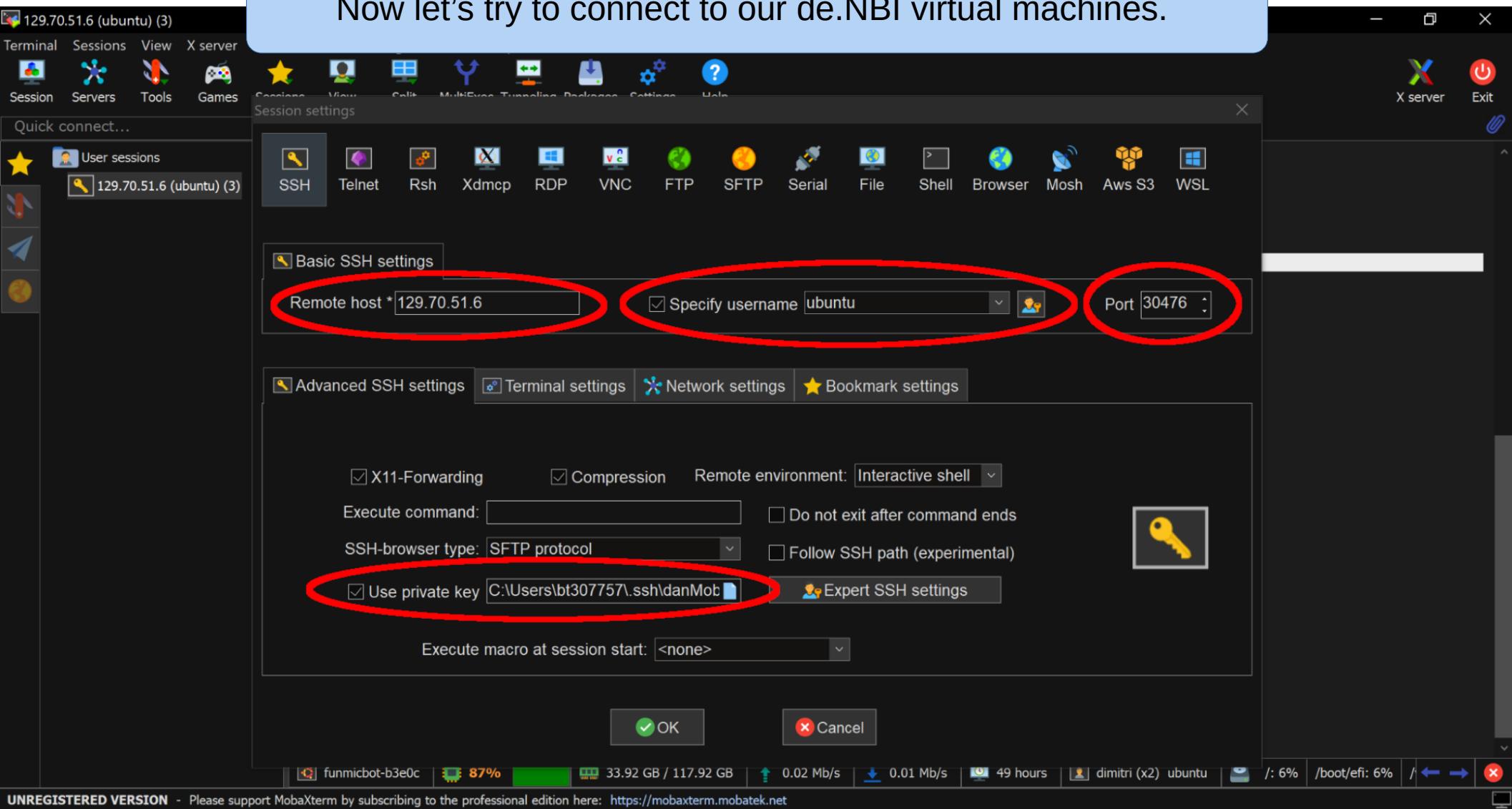
OK Cancel

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

/boot/efi: 6% / ← → ×

The screenshot shows the MobaXterm interface. In the top left, there's a session titled "129.70.51.6 (ubuntu) (3)". The top menu bar includes "Terminal", "Sessions", "View", and "X server". Below the menu is a toolbar with icons for "Session" (circled in red), "Servers", "Tools", "Games", "Sessions", "View", "Split", "MultiExec", "Tunneling", "Packages", "Settings", and "Help". On the right side of the screen are "X server" and "Exit" buttons. A blue callout box at the top center contains the text "Now let's try to connect to our de.NBI virtual machines.". The main area shows a "Quick connect..." list with a session for "/home/ubuntu/" and a "Name" field. To the right is a "Session settings" dialog with a title "Session settings" and a close button "X". It contains a list of session types: SSH (circled in red), Telnet, Rsh, Xdmcp, RDP, VNC, SFTP, Serial, File, Shell, Browser, Mosh, Aws S3, and WSL. Below the list is a placeholder text "Choose a session type..." with a small icon of a computer monitor. At the bottom of the dialog are "OK" and "Cancel" buttons. The bottom of the screen has a status bar with "UNREGISTERED VERSION" and a link to the professional edition, battery level "6%", and navigation icons. A large red arrow points from the "Session" icon in the toolbar to the "SSH" icon in the session settings dialog, and another red arrow points from the "SSH" icon in the session list to the "SSH" icon in the session settings dialog.

Now let's try to connect to our de.NBI virtual machines.

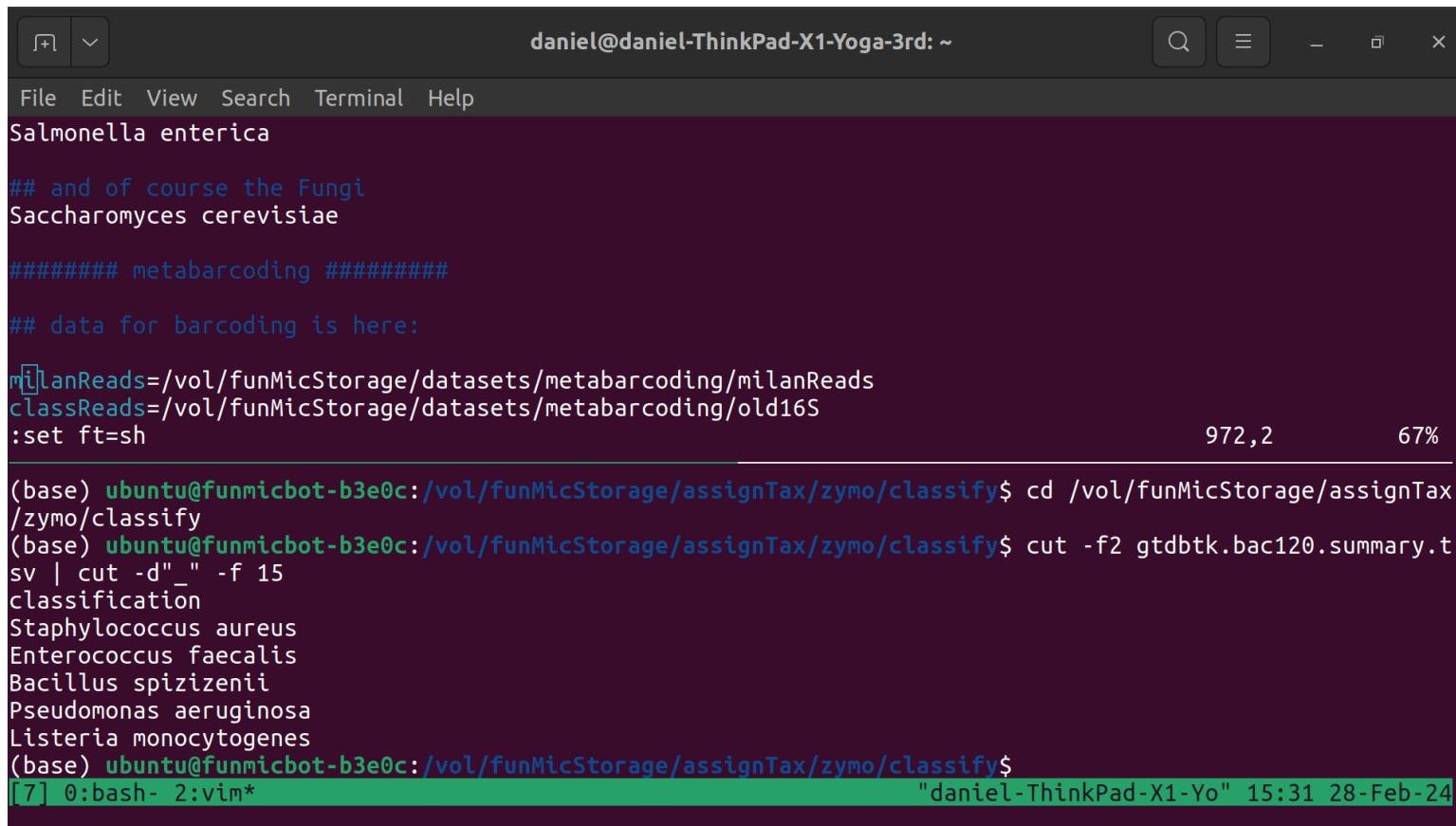


First task: connect to VMs

Dennis Büchner	ssh -X ubuntu@129.70.51.6 -p 30323	funmicStudent01
Federico Campo	ssh -X ubuntu@129.70.51.6 -p 30281	funmicStudent02
Zoi Hock	ssh -X ubuntu@129.70.51.6 -p 30365	funmicStudent03
Dongun Kim	ssh -X ubuntu@129.70.51.6 -p 30500	funmicStudent04
Fiona Peters	ssh -X ubuntu@129.70.51.6 -p 30272	funmicStudent05
Hannes Reichel	ssh -X ubuntu@129.70.51.6 -p 30453	funmicStudent06
Romina Retsch	ssh -X ubuntu@129.70.51.6 -p 30306	funmicStudent07
Isabelle Storm	ssh -X ubuntu@129.70.51.6 -p 30460	funmicStudent08
backup	ssh -X ubuntu@129.70.51.6 -p 30414	funmicStudent09

Second task: learn about linux terminals

Once you are in, open up the script for today (here) and let's learn about linux.

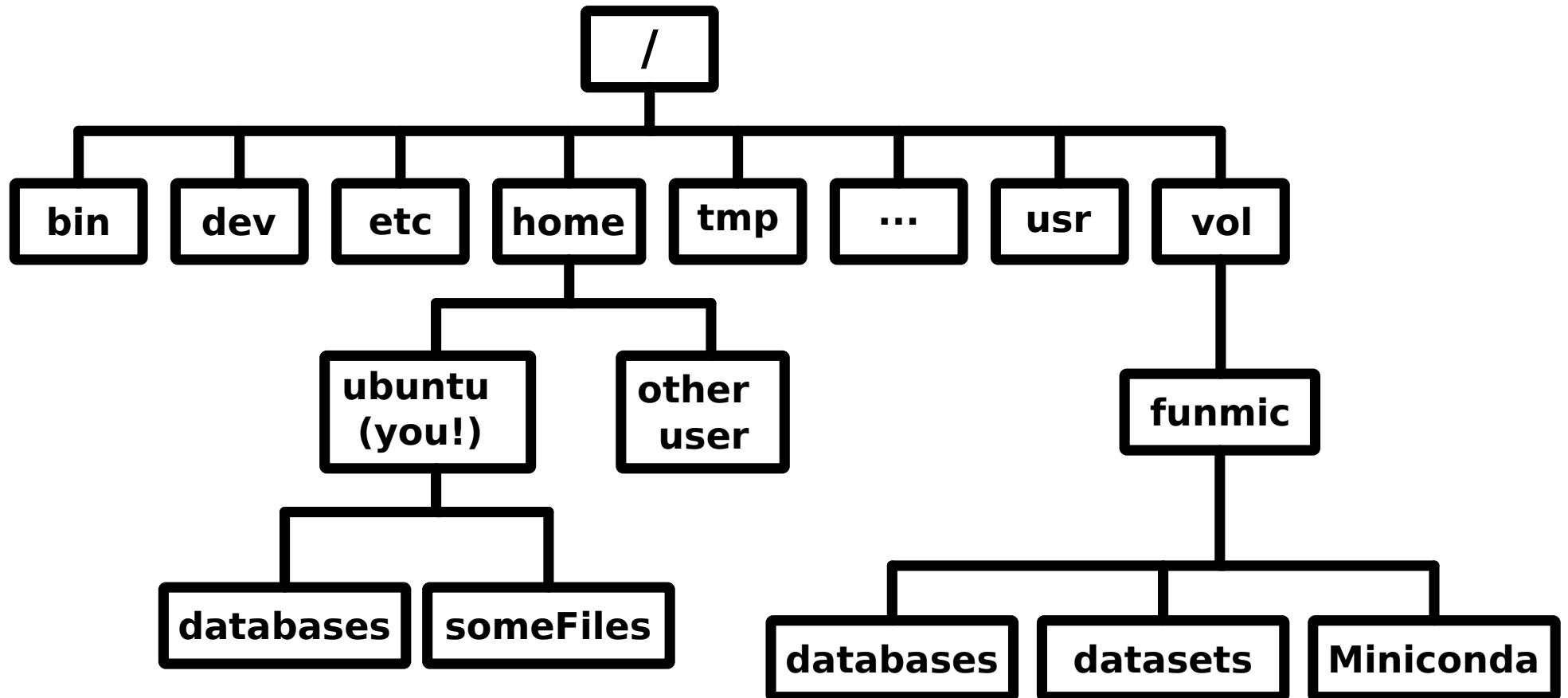


The screenshot shows a terminal window titled "daniel@daniel-ThinkPad-X1-Yoga-3rd: ~". The window includes standard OS X-style controls (minimize, maximize, close) and a menu bar with File, Edit, View, Search, Terminal, and Help. The main pane displays a script for metabarcoding analysis:

```
File Edit View Search Terminal Help
Salmonella enterica
## and of course the Fungi
Saccharomyces cerevisiae
##### metabarcoding #####
## data for barcoding is here:
milanReads=/vol/funMicStorage/datasets/metabarcoding/milanReads
classReads=/vol/funMicStorage/datasets/metabarcoding/old16S
:set ft=sh
(base) ubuntu@funmicbot-b3e0c:/vol/funMicStorage/assignTax/zymo/classify$ cd /vol/funMicStorage/assignTax/zymo/classify
(base) ubuntu@funmicbot-b3e0c:/vol/funMicStorage/assignTax/zymo/classify$ cut -f2 gtdbtk.bac120.summary.tsv | cut -d"_" -f 15
classification
Staphylococcus aureus
Enterococcus faecalis
Bacillus spizizenii
Pseudomonas aeruginosa
Listeria monocytogenes
(base) ubuntu@funmicbot-b3e0c:/vol/funMicStorage/assignTax/zymo/classify$ [7] 0:bash- 2:vim*                                     "daniel-ThinkPad-X1-Yo" 15:31 28-Feb-24
```

(cut to terminal)

Image of a file tree in Linux:



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

MiB Mem : 4.2/120747.0 [|||||]
MiB Swap: 0.2/1024.0 [

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
310297	dimitri	20	0	55.1g	3.4g	23084	S	1252	2.9	4:22.26	java
293349	ubuntu	20	0	7764	3444	3172	S	0.7	0.0	0:02.88	bash
14	root	20	0	0	0	0	I	0.3	0.0	0:45.25	rcu_sched
287908	ubuntu	20	0	17444	8644	6008	S	0.3	0.0	0:02.17	sshd
297392	ubuntu	20	0	11032	4020	3172	R	0.3	0.0	0:02.78	top
308737	dimitri	20	0	7764	3468	3192	S	0.3	0.0	0:00.25	bash

```
(base) ubuntu@funmicbot-b3e0c:~$ vol
(base) ubuntu@funmicbot-b3e0c:/vol/funMicStorage$ ls
Kelp_logs assemblies assignTax databases lost+found mapping phyloFlashOut ref
anaconda3 assemblyQC binning datasets magQC metabarcoding rawReadQC refiningBins
(base) ubuntu@funmicbot-b3e0c:/vol/funMicStorage$ cd binning/
(base) ubuntu@funmicbot-b3e0c:/vol/funMicStorage/binning$ ls
MetaBAT zymo
(base) ubuntu@funmicbot-b3e0c:/vol/funMicStorage/binning$ cd zymo/
(base) ubuntu@funmicbot-b3e0c:/vol/funMicStorage/binning/zymo$ ls
concoct metabat vamb
(base) ubuntu@funmicbot-b3e0c:/vol/funMicStorage/binning/zymo$ cd concoct
(base) ubuntu@funmicbot-b3e0c:/vol/funMicStorage/binning/zymo/concoct$ ls
PCA_components_data_gt1000.csv clustering_gt1000.csv concoctContigs_10K.bed fasta_bins
```

You will need to transfer files from your de.NBI virtual machine to your local computer often.

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

X server Exit

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

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

The screenshot shows a MobaXterm interface with a terminal window titled "12. 129.70.51.6 (ubuntu) (3)". The terminal window displays the following information:

- System statistics:
 - MiB Mem : 4.2/120747.0 [|||||]
 - MiB Swap: 0.2/1024.0 []
- Process list (top command):

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
310297	dimitri	20	0	55.1g	3.4g	23084	S	1252	2.9	4:22.26	java
293349	ubuntu	20	0	7764	3444	3172	S	0.7	0.0	0:02.88	bash
14	root	20	0	0	0	0	I	0.3	0.0	0:45.25	rcu_sched
287908	ubuntu	20	0	17444	8644	6008	S	0.3	0.0	0:02.17	sshd
297392	ubuntu	20	0	11032	4020	3172	R	0.3	0.0	0:02.78	top
308737	dimitri	20	0	7764	3468	3192	S	0.3	0.0	0:00.25	bash

- File list (ls command):
 - (base) ubuntu@funmicbot-b3e0c:~\$ vol
 - (base) ubuntu@funmicbot-b3e0c:/vol/funMicStorage\$ ls
 - Kelp_logs assemblies assignTax databases lost+found mapping phyloFlashOut ref anaconda3 assemblyQC binning datasets magQC metabarcoding rawReadQC refiningBins
 - (base) ubuntu@funmicbot-b3e0c:/vol/funMicStorage\$ cd binning/
 - (base) ubuntu@funmicbot-b3e0c:/vol/funMicStorage/binning\$ ls
 - MetaBAT zymo
 - (base) ubuntu@funmicbot-b3e0c:/vol/funMicStorage/binning\$ cd zymo/
 - (base) ubuntu@funmicbot-b3e0c:/vol/funMicStorage/binning/zymo\$ ls
 - concoct metabat vamb
 - (base) ubuntu@funmicbot-b3e0c:/vol/funMicStorage/binning/zymo\$ cd concoct
 - (base) ubuntu@funmicbot-b3e0c:/vol/funMicStorage/binning/zymo/concoct\$ ls
 - PCA_components_data_gt1000.csv clustering_gt1000.csv concoctContigs_10K.bed fasta_bins

On the left side of the terminal window, there is a file browser pane showing a list of files in the directory "/vol/funMicStorage/binning/zymo". The files listed are: .., 0.fa, 1.fa, 10.fa, 11.fa, 12.fa, 13.fa, 14.fa, 2.fa, 3.fa, 4.fa, 5.fa, 6.fa, 7.fa, 8.fa, 9.fa. The file "3.fa" is highlighted with a blue selection bar. Two red arrows point from the text "There are many ways to do this, but we'll try MobaXterm's builtin abilities." to the "3.fa" file entry in the file browser.

At the bottom of the screen, there is a status bar with various system information and a message: "UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net".

Text at the bottom of the slide: "There are many ways to do this, but we'll try MobaXterm's builtin abilities."

And final note: 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 -MMD -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO_NO_AVR_UNO -DARDUINO=10607 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduinoVariant .pioenvs/uno/FrameworkArduino/wiring_digital.c
avr-gcc -o .pioenvs/uno/FrameworkArduino/wiring_pulse.o -c -g -Os -Wall -ffunction-sections -fdata-sections -MMD -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO_NO_AVR_UNO -DARDUINO=10607 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduinoVariant .pioenvs/uno/FrameworkArduino/wiring_pulse.c
avr-gcc -o .pioenvs/uno/FrameworkArduino/wiring_shift.o -c -g -Os -Wall -ffunction-sections -fdata-sections -MMD -mmcu=atmega328p -DF_CPU=16000000L -DARDUINO_ARCH_AVR -DARDUINO_NO_AVR_UNO -DARDUINO=10607 -I.pioenvs/uno/FrameworkArduino -I.pioenvs/uno/FrameworkArduinoVariant .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/IPAddress.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/WInterrupts.o .pioenvs/uno/FrameworkArduino/WMath.o .pioenvs/uno/FrameworkArduino/WString.o .pioenvs/uno/FrameworkArduino_wiring_pulse.o .pioenvs/uno/FrameworkArduino_abi.o .pioenvs/uno/FrameworkArduino_hooks.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 -mmcu=atmega328p -Wl,--gc-sections,--relax .pioenvs/uno/src/blink.o -L/Users/zach/.platformio/packages/ldscripts -L.pioenvs/uno -Wl,--start-group -Lm .pioenvs/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" --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)

=====
[BUILD 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++/lPlatformIO pair ws yes +
digitalWrite(uint8_t, uint8_t) -> void
55: 0 -R-*compilation*
Bot Compilation:exit [0] Projectile
```