Metagenomics & Pathogen Discovery

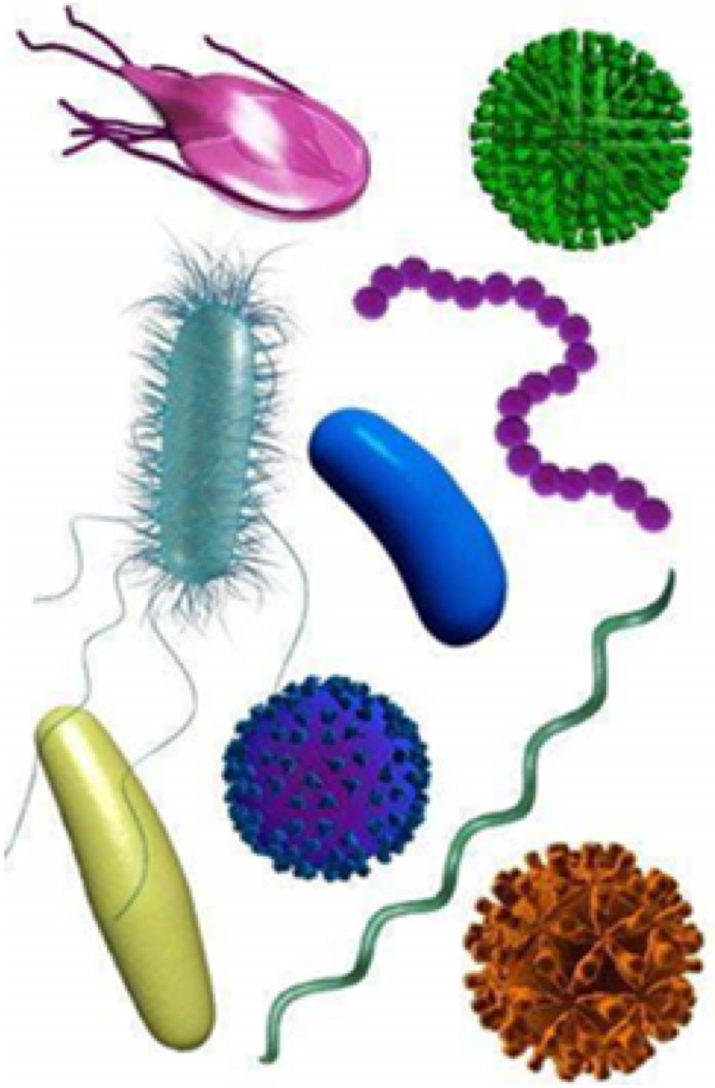
Jackie Mahar & Mang Shi
University of Sydney

jackie.mahar@sydney.edu.au
mang.shi@sydney.edu.au

THE UNIVERSITY OF SYDNEY

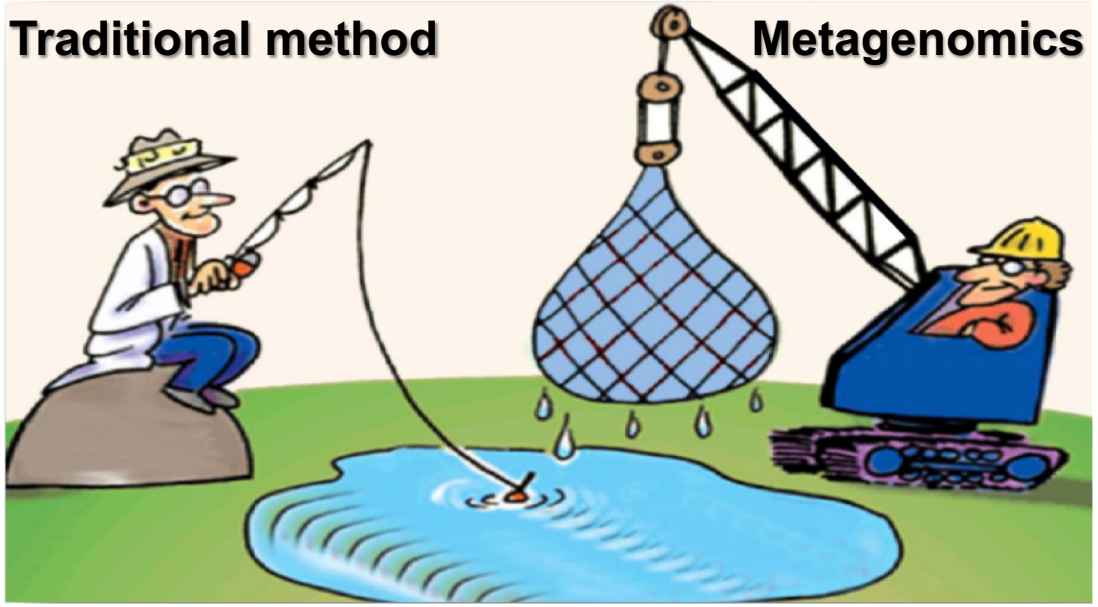
Workshop overview

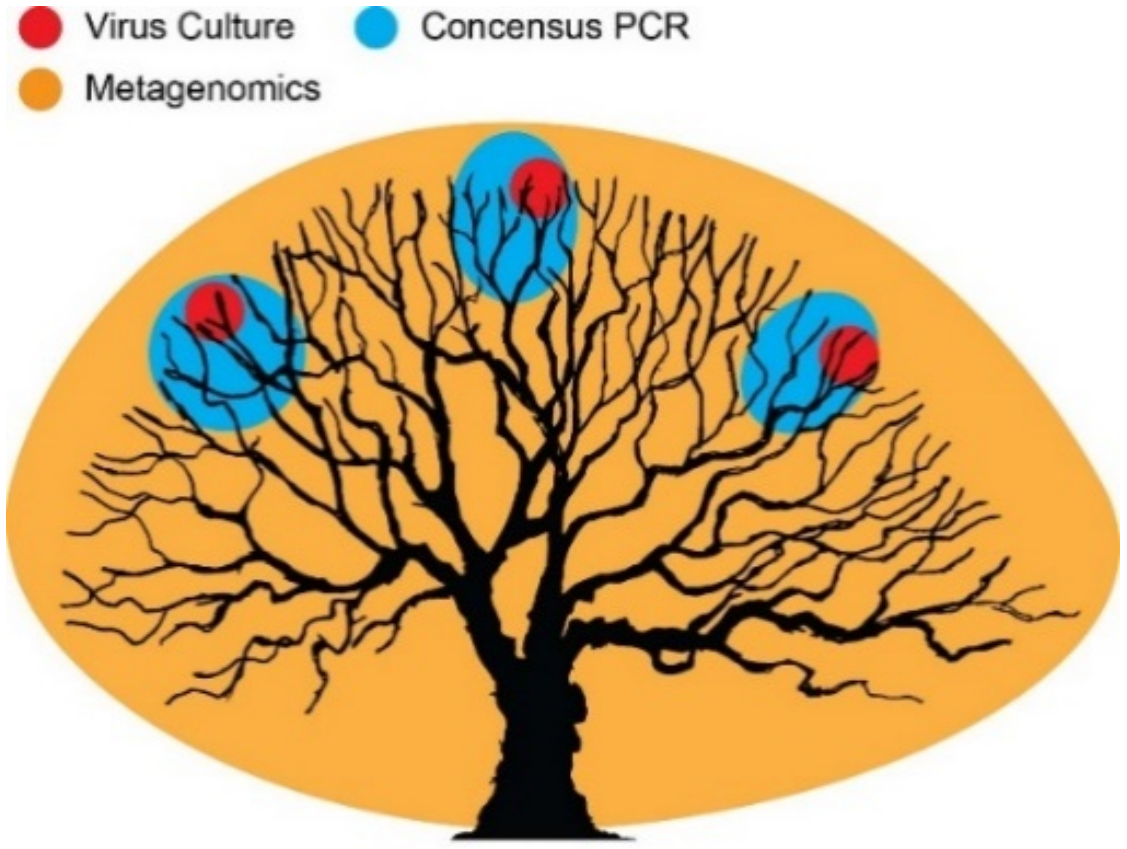
- Background
- Task 1. *De novo* assembly using Megahit
- Task 2. Annotation
  - 2.1 Blastn against nt database
  - 2.2 Blastx against nr database
  - 2.3 Taxonomy lineage annotation
- Task 3: Estimate abundance by mapping



Background: pathogen discovery methods

- Culture based approaches
- PCR approaches
- Metagenomics approaches

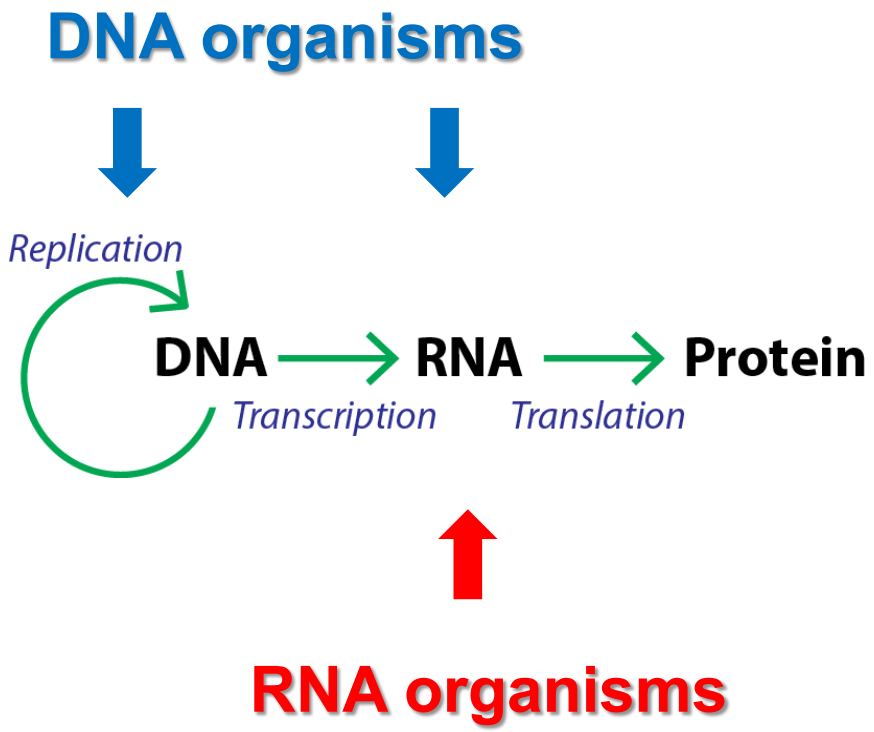




A phylogeny of RNA viruses

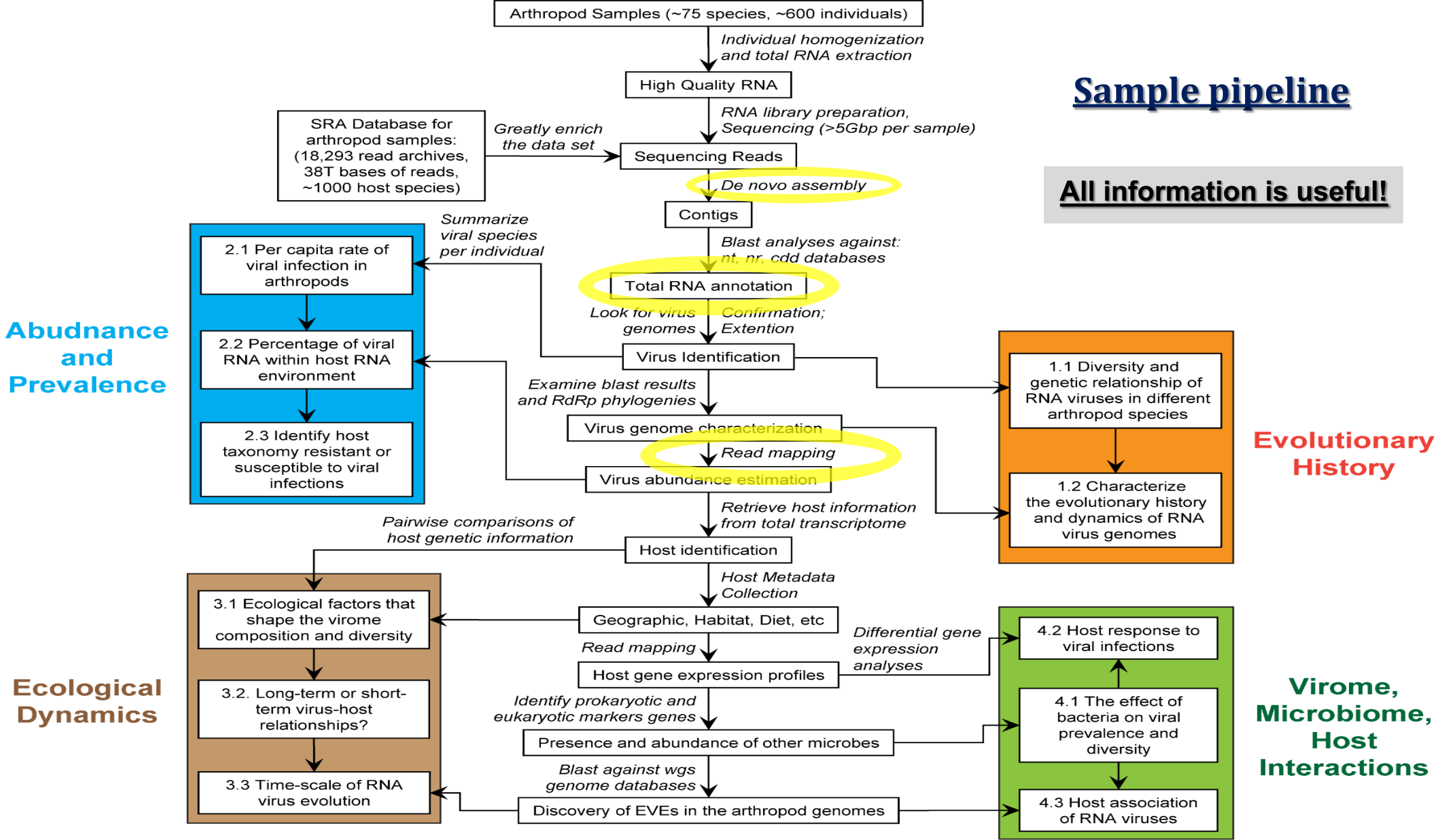
Background: Meta-transcriptomics

- RNA is transcribed in all living organisms
- The “information load” is more **balanced** across domains of life
- Concise and capture the **key biological process** (less “Junk”)
- Obtain and quantify virus (DNA and RNA), bacteria, host transcripts in one go



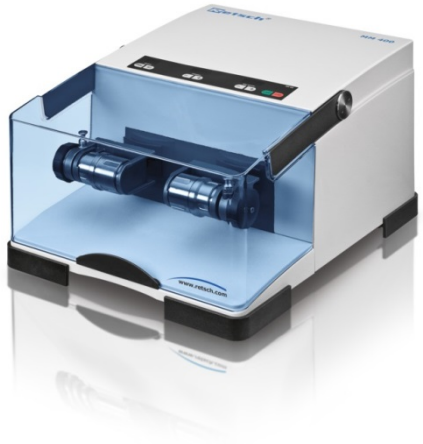
Sample pipeline

All information is useful!



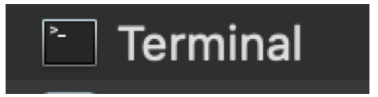
Background: sample used in this study

- Sample: tissue homogenates of 3-4 cattle tick (*Rhipicephalus microplus*)
- Process: Total RNA extraction (including host, virus, bacteria)
- Library preparation: RNA-seq library construction, host rRNA removed by Illumina Ribozero Gold (Human/Mouse/Rat) kit.
- Data: 1 Gbp single-end reads produced by Illumina HiSeq2000 sequencer
  - Fastq format

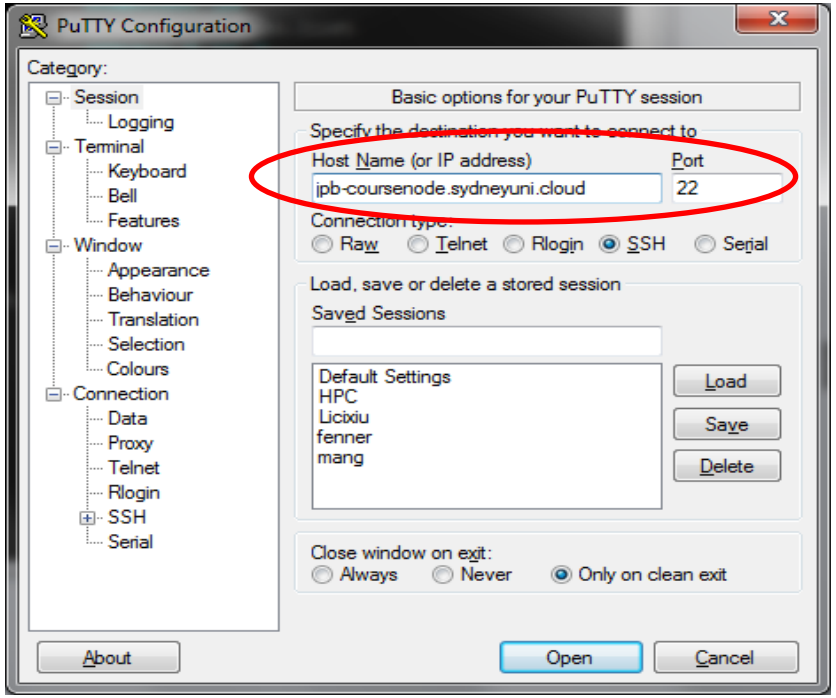


Preparation: setup terminal

- Open up pipeline.txt – downloaded from the github
  - https://github.com/genomes-biodiversity/gab-metagenomics
- Accessing the linux server from a terminal using command line
  - Mac users can use the inbuilt terminal in Applications>utilities>terminal



  - Windows users will use “Putty.exe” for terminal (or another terminal you familiar with)



Under Host Name, type in:
jpb-coursenode.sydneyuni.cloud
Port: **22**
& click open

Preparation: log in to supercomputer

- Everyone has their own temp username and pw to log on to the linux computer

- **On windows machine/Putty**
  - Login by typing your username
  - Then type your pw (same as username)

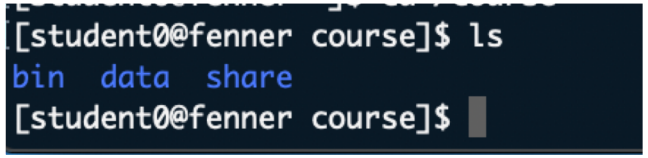
- **In the mac terminal**
  - log on using ssh

```
ssh <username>@jpb-coursenode.sydneyuni.cloud
```

  - Then type pw (same as username)

- You will be in your home directory /~
  - → This will be your working directory
- Course files in /course

```
ls /course
```



Course directory

```
###~~~~~~~~~~~~~~~~~~~~ Don't panic ~~~~~~~~~~~~~~~~~~~~~~~~###
[student0@fenner ~]$ cd /course
[student0@fenner course]$ ls
bin  data  share
[student0@fenner course]$
```

Some tools

(other tools elsewhere)

- Everyone can read

Space for sharing files

- Everyone can read and write

Input dataset (fastq file) and pipeline

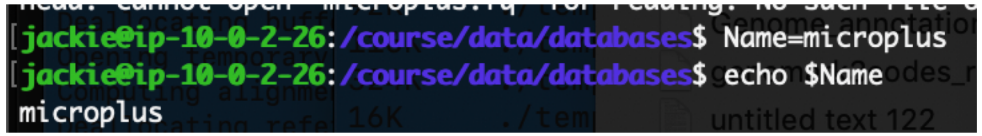
- Everyone can read

```
[student0@fenner ~]$ ls /course/data
databases  microplus.fq  output  pipeline.txt
```

Pre-prepared output files in case required

Preparation: Define variables

- Variables are named symbols that represent either a string or numeric value. When you use them in commands, they are treated as if you had typed the value they hold instead of the name of the variable.
- Eg using **Name** as variable in place of the file stem name:
  - Fastq file is microplus.fq and all output files will include “microplus” in the name, therefore define variable for “microplus”



- `Name=microplus` ##define variable
- `echo $Name` ##check the value held in the variable
  - Then use “$Name” in place of microplus
  - eg `less microplus.fq` can also be written as `less "$Name".fq`

Preparation: Define variables

- We can use the **Name** variable throughout the pipeline
- To analyse a different file, instead of changing the file names throughout the pipeline you just redefine the **Name** variable.
  - i.e. for a fastq input file named tissue1.fq, define the variable as **Name=tissue1**

- Variables are
  - Assigned using an equals sign (=) eg. **Name=microplus**
  - Referenced using dollar sign ($) eg. **$Name**
  - Case sensitive
  - Deleted when shell script completes - need to define variables again if changing between machines or environments

Preparation: Define variables

- Variables for this workshop - enter into terminal

```
Name=microplus  ## sample stem name
inpath=/course/data  ## dir containing fq input file
dbnt=/course/data/databases  ## path to databases
dbnr=/course/data/databases/diamond  ## path to diamond nr database
access2taxid=/course/data/databases  ## path to prot.accession2taxid db
taxonomist=/course/bin/simbiont/ncbi/tools  ## path to ncbi.taxonomist.py
```

- check

```
echo $taxonomist
```

```
vlan2765-d17-10-65-126-132:~ jackiemahar$ taxonomist=/course/bin/simbiont/ncbi/tools
vlan2765-d17-10-65-126-132:~ jackiemahar$ echo $taxonomist
/course/bin/simbiont/ncbi/tools
```

Preparation: Data observation

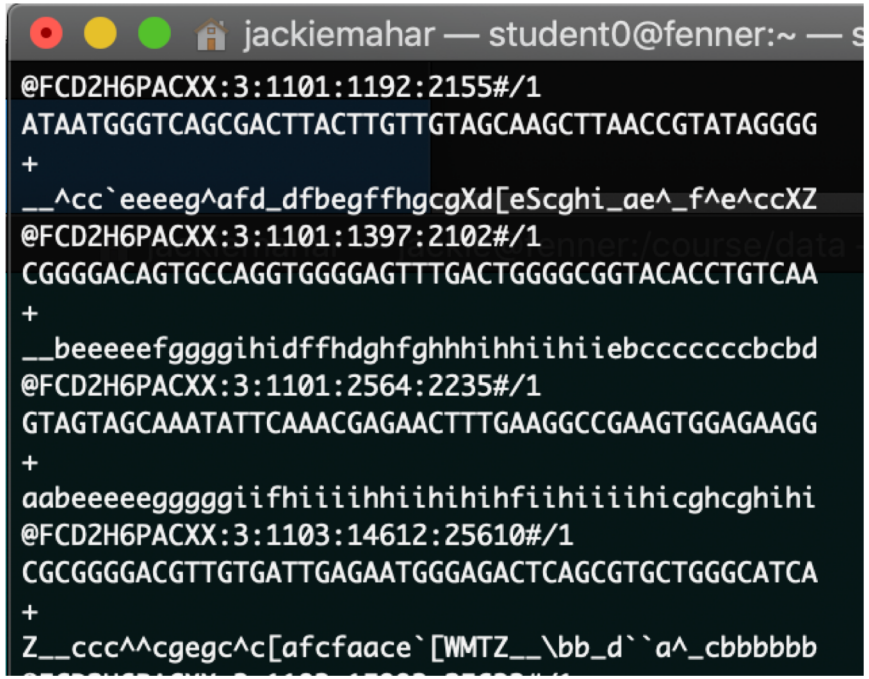
- Output file from RNA-seq is a fastq file
- Our pipeline therefore uses a fastq file as first input file
- The fastq input file for this workshop is microplus.fq
- Microplus.fq is in course/data
- View file

```
less /course/data/microplus.fq
```

OR because we have defined variables:

```
less $inpath/"$Name".fq
```

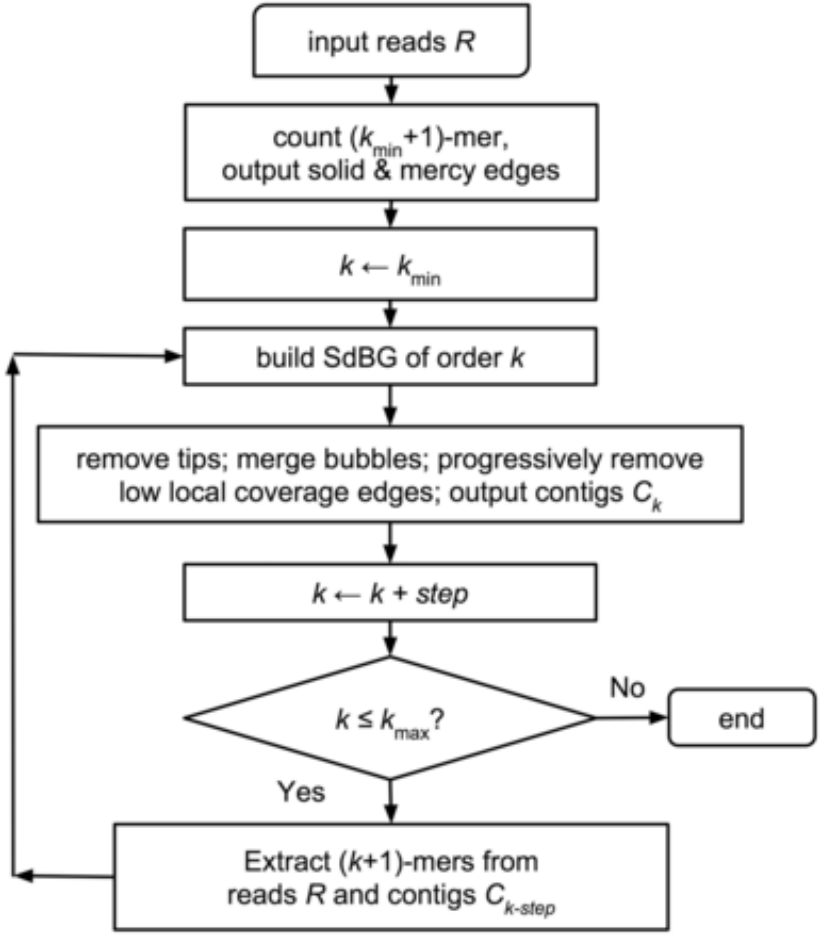
Type q to exit



Variables
Name=microplus
inpath=/course/data

Task 1: *de novo* assembly

- *De novo* (no genome template)
- RNA sequencing data (short reads 100-250bp)
- Megahit: **ultra-fast** and **memory-efficient** metagenomic assembler



Dinghua Li et al. 2015

Dinghua Li et al. 2015. *Bioinformatics*, Volume 31, Issue 10 https://doi.org/10.1093/bioinformatics/btv033

Task 1: *de novo* assembly

- Command to run the assembly:

```
megahit --num-cpu-threads 2 --memory 0.1 -r $inpath/"$Name".fq -o "$Name"_out
```

```
--num-cpu-threads <int> number of CPU threads
--memory <float> max memory in byte to be used in SdBG construction
                 (if set between 0-1, fraction of the machine's total memory) [0.9]
-r comma-separated list of fasta/q single-end files
                 (Use -1 and -2 for R1 and R2 paired end read files respectively)
-o <string> output directory [./megahit_out]
```

- Job should take less than 1 min (normally ~1 day)
- Output file microplus_out should be created

Variables
Name=microplus
inpath=/course/data

Task 1: *de novo* assembly

```
[student0@fenner ~]$ head "$Name"_out/final.contigs.fa
>k39_1 flag=1 multi=43.0000 len=347
GTGAGCGGCGTACGCGAAAGCGACCGGGTGTCTGATATGCGGCGGGCTTCGGCCTCGTCAAGCGTGTCCTCGGGTCTG
CTCCAGGGGAATCCACGGCAGTCGTCTGCAGCCTCATCCGCTTGATGCGTTAGGGGCTGGTTGTCGGACGGTGCCGTT
TTACCACGATATCGAGGTGTGTTCCGTGTCGTCCTCGGGCGATTCAGATGCGAAAGCGCCGAAGACGCGGGCGTGACC
CGTCGTCTGGCGGCTTTGCAGTCTCGGCTCCGTTGCTAGTTCCGGCCGGTCCACCGACAGTGCAGCGGGCTTGGGCAA
CCCGCACGGCGCGACCGAGTCGATGCAACGAAAAA
>k39_2 flag=1 multi=161.0000 len=1430
CCGCTTCTGGTACAATTCACTTCCATGGTGTGACGGGCGGTGTGTACAAGGCCCGAGAACGTATTCACCGCGACATTC
TGATTCGCGATTACTAGCGATTCCAACTTCATGGAGTCGAGTTGCAGACTCCAATCCGGACTACGAACAGTTTTTTGG
```

Contig names

Blast will ignore everything after the space (replace space with underscore)

substitute
space
underscore
Repeat for all instances on line

➢edit the contig name for later blast analyses

```
cat ./"$Name"_out/final.contigs.fa | sed "s/=//g" | sed "s/ /_/g" > "$Name".contigs.fa
```

cat is a utility that reads files and writes them to standard output

sed is a “stream editor” utility that performs many functions including substitutions

Variables
Name=microplus

Task 1: *de novo* assembly

```
[student0@fenner ~]$ head "$Name"_out/final.contigs.fa
>k39_1 flag=1 multi=43.0000 len=347
GTGAGCGGCGTACGCGAAAGCGACCGGGTGTCTGATATGCGGCGGGCTTCGGCCTCGTCAAGCGTGTCCTCGGGTCTG
CTCCAGGGGAATCCACGGCAGTCGTCTGCAGCCTCATCCGCTTGATGCGTTAGGGGCTGGTTGTCGGACGGTGCCGTT
TTACCACGATATCGAGGTGTGTTCCGTGTCGTCCTCGGGCGATTCAGATGCGAAAGCGCCGAAGACGCGGGCGTGACC
CGTCGTCTGGCGGCTTTGCAGTCTCGGCTCCGTTGCTAGTTCCGGCCGGTCCACCGACAGTGCAGCGGGCTTGGGCAA
CCCGCACGGCGCGACCGAGTCGATGCAACGAAAAA
>k39_2 flag=1 multi=161.0000 len=1430
CCGCTTCTGGTACAATTCACTTCCATGGTGTGACGGGCGGTGTGTACAAGGCCCGAGAACGTATTCACCGCGACATTC
```

Contig names
Blast will ignore everything after the space (replace space with underscore)

- edit the contig name for later blast analyses

```
cat ./"$Name"_out/final.contigs.fa | sed "s/=//g" | sed "s/ /_/g" > "$Name".contigs.fa
```

- Visualize the assembly result

```
less "$Name".contigs.fa
```

```
jackiemahar — student0@fenner:~ — ssh student0@fenner.bio.usyd.edu.au — 102×31
>k39_1_flag1_multi43.0000_len347
GTGAGCGGCGTACGCGAAAGCGACCGGGTGTCTGATATGCGGCGGGCTTCGGCCTCGTCAAGCGTGTCCTCGGGTCTGCTCCAGGGGAATCCACGGCAGTCG
TCTGCAGCCTCATCCGCTTGATGCGTTAGGGGCTGGTTGTCGGACGGTGCCGTTTTACCACGATATCGAGGTGTGTTCCGTGTCGTCCTCGGGCGATTCAGA
TGCGAAAGCGCCGAAGACGCGGGCGTGACCCGTCGTCTGGCGGCTTTGCAGTCTCGGCTCCGTTGCTAGTTCCGGCCGGTCCACCGACAGTGCAGCGGGCTT
GGGCAACCCGCACGGCGCGACCGAGTCGATGCAACGAAAAA
>k39_2_flag1_multi161.0000_len1430
CCGCTTCTGGTACAATTCACTTCCATGGTGTGACGGGCGGTGTGTACAAGGCCCGAGAACGTATTCACCGCGACATTCTGATTCGCGATTACTAGCGATTCC
```

Variables
Name=microplus