Workshop
Introduction to Microbiome analysis

Yu-Wei Wu

Taipei Medical University

Who am I?

- I am Yu-Wei Wu from the Graduate Institute of Biomedical Informatics, Taipei Medical University
- I was a software engineer before I switch my career toward biological sciences
- I am more specialized in genomics and metagenomics analysis

What to expect from this workshop

- Even though I will not talk about everything, I will attempt to bring you guys into the world of metagenomics analysis.
- I will start the introduction from platform, settings, environments, software, and things that we as analysts may pay attention to.
- Things to be introduced include but not limited to
  - Linux platform setup
  - Software installation and compilation
  - Brief steps for metagenomics analysis

Linux

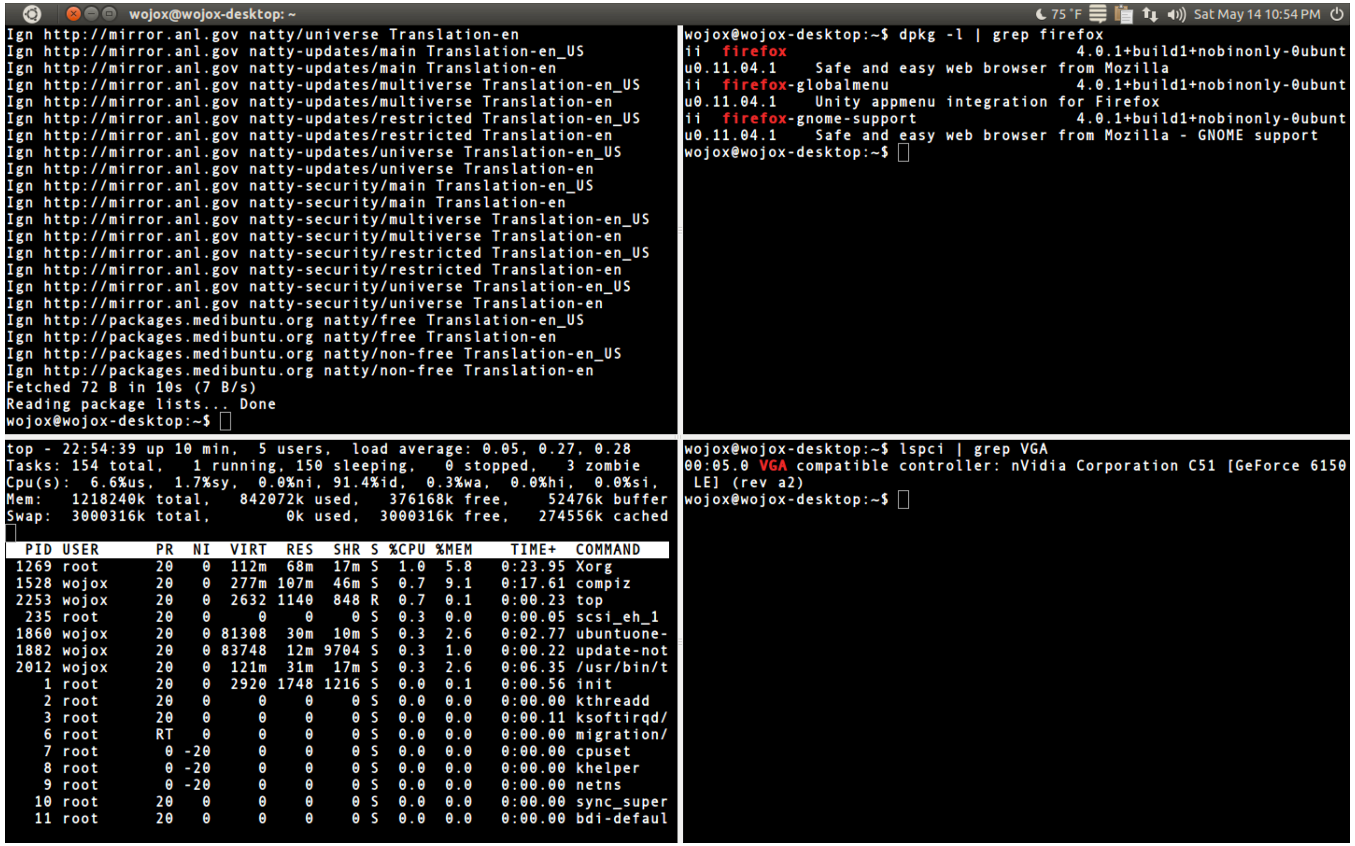
- Firstly, linux
- Due to the convention of bioinformatics development, most analysis software packages were developed and maintained on linux
- But afraid not, as linux is now very easy to be installed everywhere

- (Note: Macintosh/Apple is sometimes not compatible with linux)

Terminal

- And you don’t need to be familiar with linux at all. You only need to know a few tips and commands for using linux terminals.

I very commonly organize the terminal windows like this. Having multiple windows helps a lot in the process of developing and using bioinformatics pipelines.



```
Ign http://mirror.anl.gov natty/universe Translation-en
Ign http://mirror.anl.gov natty-updates/main Translation-en_US
Ign http://mirror.anl.gov natty-updates/main Translation-en
Ign http://mirror.anl.gov natty-updates/multiverse Translation-en_US
Ign http://mirror.anl.gov natty-updates/multiverse Translation-en
Ign http://mirror.anl.gov natty-updates/restricted Translation-en_US
Ign http://mirror.anl.gov natty-updates/restricted Translation-en
Ign http://mirror.anl.gov natty-updates/universe Translation-en_US
Ign http://mirror.anl.gov natty-updates/universe Translation-en
Ign http://mirror.anl.gov natty-security/main Translation-en_US
Ign http://mirror.anl.gov natty-security/main Translation-en
Ign http://mirror.anl.gov natty-security/multiverse Translation-en_US
Ign http://mirror.anl.gov natty-security/multiverse Translation-en
Ign http://mirror.anl.gov natty-security/restricted Translation-en_US
Ign http://mirror.anl.gov natty-security/restricted Translation-en
Ign http://mirror.anl.gov natty-security/universe Translation-en_US
Ign http://mirror.anl.gov natty-security/universe Translation-en
Ign http://packages.medibuntu.org natty/free Translation-en_US
Ign http://packages.medibuntu.org natty/free Translation-en
Ign http://packages.medibuntu.org natty/non-free Translation-en_US
Ign http://packages.medibuntu.org natty/non-free Translation-en
Fetched 72 B in 10s (7 B/s)
Reading package lists... Done
wojox@wojox-desktop:~$ 
```

```
wojox@wojox-desktop:~$ dpkg -l | grep firefox
ii  firefox                                 4.0.1+build1+nobinonly-0ubunt
u0.11.04.1    Safe and easy web browser from Mozilla
ii  firefox-globalmenu                      4.0.1+build1+nobinonly-0ubunt
u0.11.04.1    Unity appmenu integration for Firefox
ii  firefox-gnome-support                   4.0.1+build1+nobinonly-0ubunt
u0.11.04.1    Safe and easy web browser from Mozilla - GNOME support
wojox@wojox-desktop:~$ 
```

```
top - 22:54:39 up 10 min,  5 users,  load average: 0.05, 0.27, 0.28
Tasks: 154 total,   1 running, 150 sleeping,   0 stopped,   3 zombie
Cpu(s):  6.6%us,  1.7%sy,  0.0%ni, 91.4%id,  0.3%wa,  0.0%hi,  0.0%si,
Mem:   1218240k total,   842072k used,   376168k free,    52476k buffer
Swap:  3000316k total,        0k used,  3000316k free,   274556k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 1269 root      20   0  112m  68m  17m S  1.0  5.8   0:23.95 Xorg
 1528 wojox     20   0  277m 107m  46m S  0.7  9.1   0:17.61 compiz
 2253 wojox     20   0  2632 1140  848 R  0.7  0.1   0:00.23 top
  235 root      20   0     0    0    0 S  0.3  0.0   0:00.05 scsi_eh_1
 1860 wojox     20   0 81308  30m  10m S  0.3  2.6   0:02.77 ubuntuone-
 1882 wojox     20   0 83748  12m 9704 S  0.3  1.0   0:00.22 update-not
 2012 wojox     20   0  121m  31m  17m S  0.3  2.6   0:06.35 /usr/bin/t
    1 root      20   0  2920 1748 1216 S  0.0  0.1   0:00.56 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kthreadd
    3 root      20   0     0    0    0 S  0.0  0.0   0:00.11 ksoftirqd/
    6 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/
    7 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 cpuset
    8 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 khelper
    9 root       0 -20     0    0    0 S  0.0  0.0   0:00.00 netns
   10 root      20   0     0    0    0 S  0.0  0.0   0:00.00 sync_super
   11 root      20   0     0    0    0 S  0.0  0.0   0:00.00 bdi-defaul
```

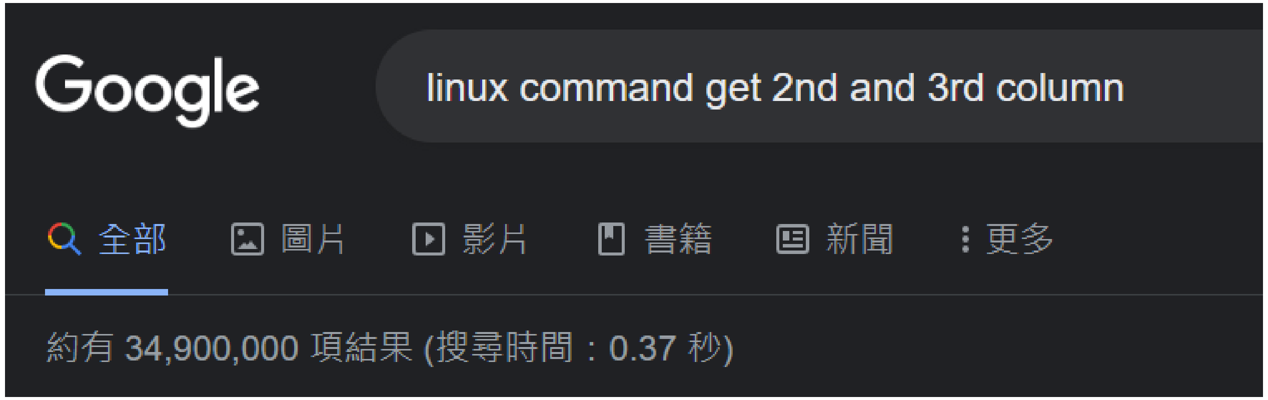
```
wojox@wojox-desktop:~$ lspci | grep VGA
00:05.0 VGA compatible controller: nVidia Corporation C51 [GeForce 6150
 LE] (rev a2)
wojox@wojox-desktop:~$ 
```

Linux commands?

- Yes you need to know how to use linux commands. But afraid not again, as the internet is your good friend.

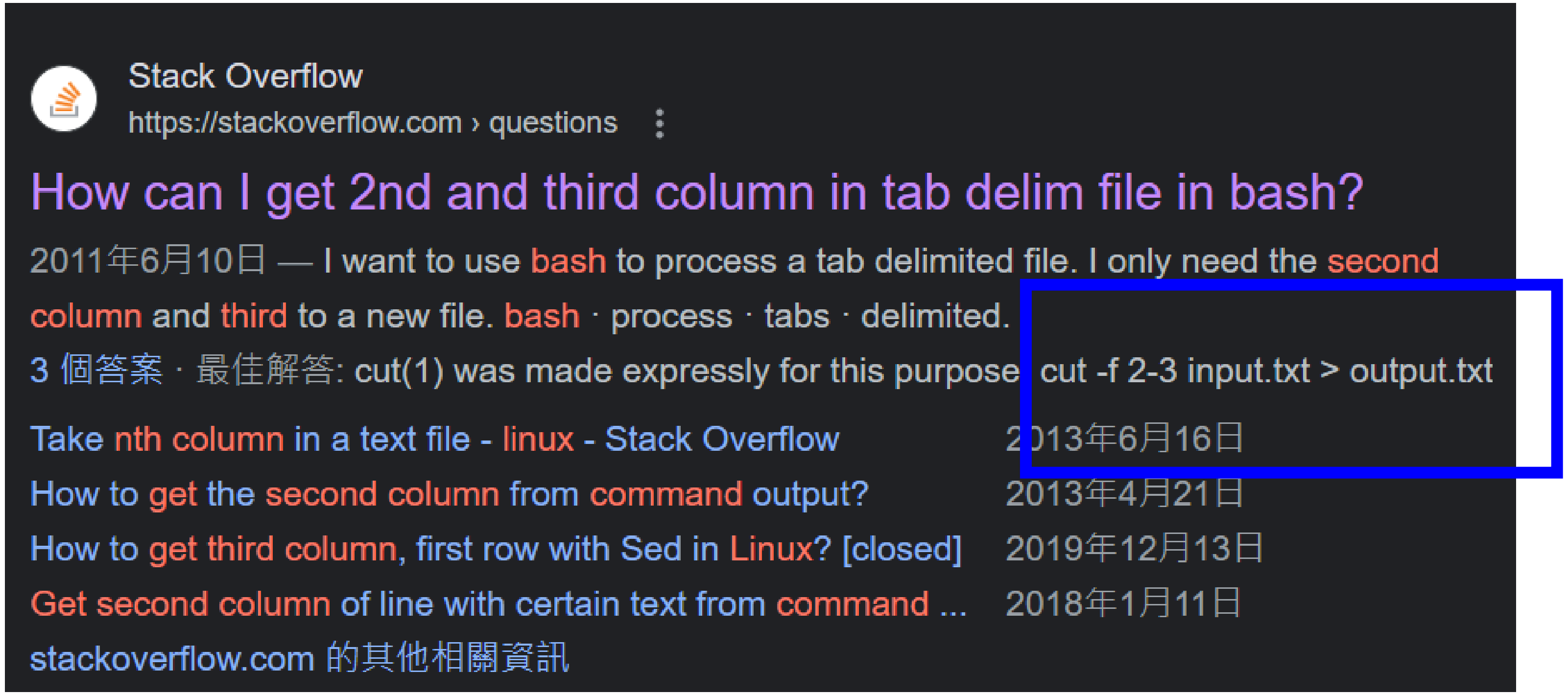
- Just “Google” your question.

Example question: get columns



linux command get 2nd and 3rd column

約有 34,900,000 項結果 (搜尋時間：0.37 秒)



Stack Overflow
https://stackoverflow.com › questions

How can I get 2nd and third column in tab delim file in bash?

2011年6月10日 — I want to use bash to process a tab delimited file. I only need the second column and third to a new file. bash · process · tabs · delimited.

3 個答案 · 最佳解答: cut(1) was made expressly for this purpose. cut -f 2-3 input.txt > output.txt

Take nth column in a text file - linux - Stack Overflow	2013年6月16日
How to get the second column from command output?	2013年4月21日
How to get third column, first row with Sed in Linux? [closed]	2019年12月13日
Get second column of line with certain text from command ...	2018年1月11日

stackoverflow.com 的其他相關資訊

Software installation

- We are not learning linux, so I will just put forth a few hints such that you can “install” software more easily.

- Usually, the downloaded software can be “compiled” to make an executable that can run on the linux

Our example linux environment: Ubuntu

- In this workshop I will use Ubuntu as the example for everything
- To prepare our work environment, we need to prepare the Ubuntu system for a few things
  - “Things that are needed to compiled programs”

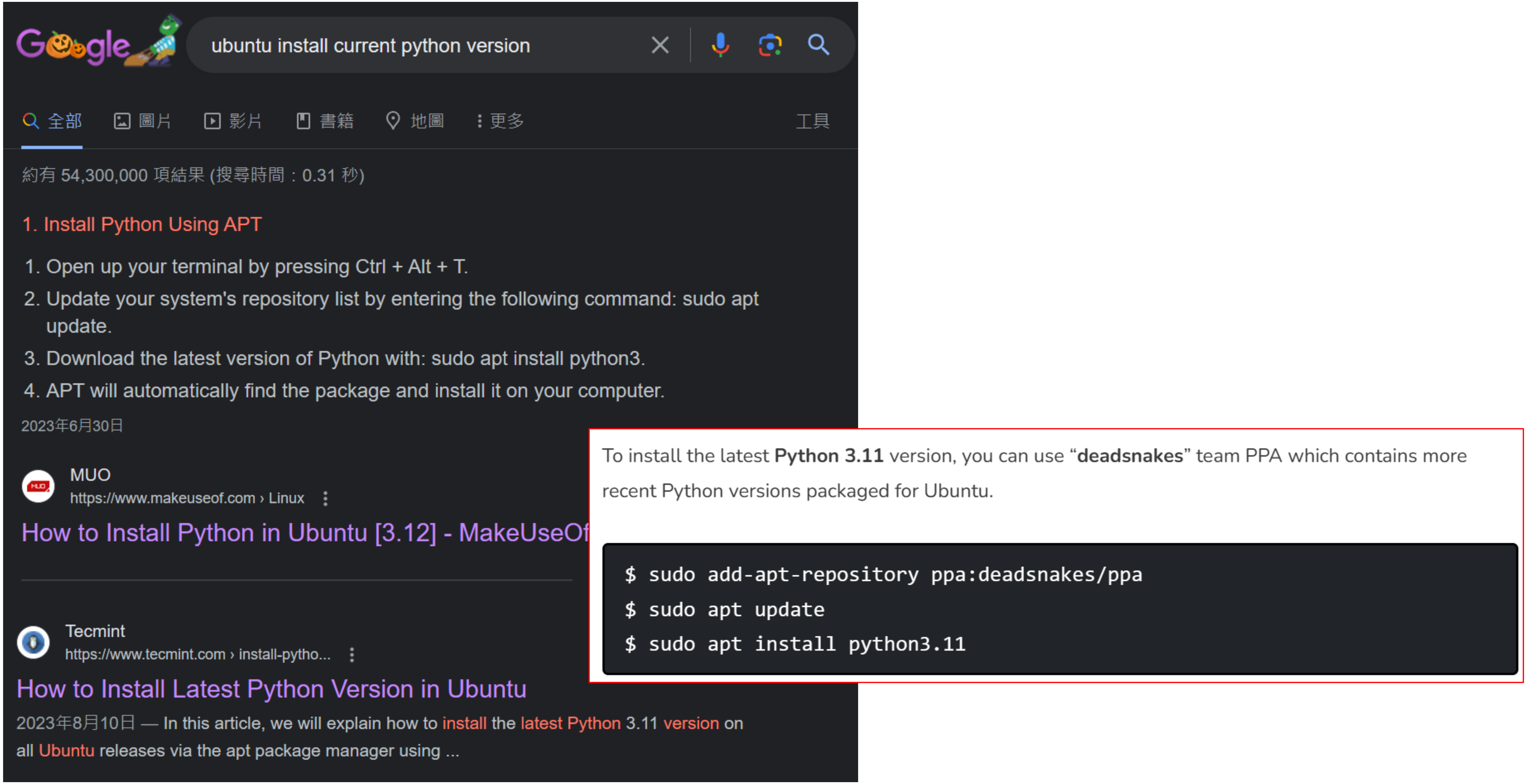
- In Ubuntu, system-wide software can be installed using “apt” or “apt-get” command
  - For example, you can type “**sudo** apt-get install vim” to install the vi editor, or “**sudo** apt-get install wget” to install the web crawler tool wget

One command to prepare Ubuntu for compilation tasks

```
sudo apt-get install build-essential
```

- This command installs essential packages (most but still not all of them) for compiling packages

- For example, you may find that your Python version is still quite outdated (should not happen if you are installing current version of Ubuntu). When this happens remember googling it. There are tons of answers available on the Internet.



約有 54,300,000 項結果 (搜尋時間：0.31 秒)

1. Install Python Using APT

1. Open up your terminal by pressing Ctrl + Alt + T.
2. Update your system's repository list by entering the following command: sudo apt update.
3. Download the latest version of Python with: sudo apt install python3.
4. APT will automatically find the package and install it on your computer.

2023年6月30日

MUO
https://www.makeuseof.com › Linux

How to Install Python in Ubuntu [3.12] - MakeUseOf

Tecmint
https://www.tecmint.com › install-pytho...

How to Install Latest Python Version in Ubuntu

2023年8月10日 — In this article, we will explain how to install the latest Python 3.11 version on all Ubuntu releases via the apt package manager using ...

To install the latest **Python 3.11** version, you can use “**deadsnakes**” team PPA which contains more recent Python versions packaged for Ubuntu.

```
$ sudo add-apt-repository ppa:deadsnakes/ppa
$ sudo apt update
$ sudo apt install python3.11
```

“Install” programs

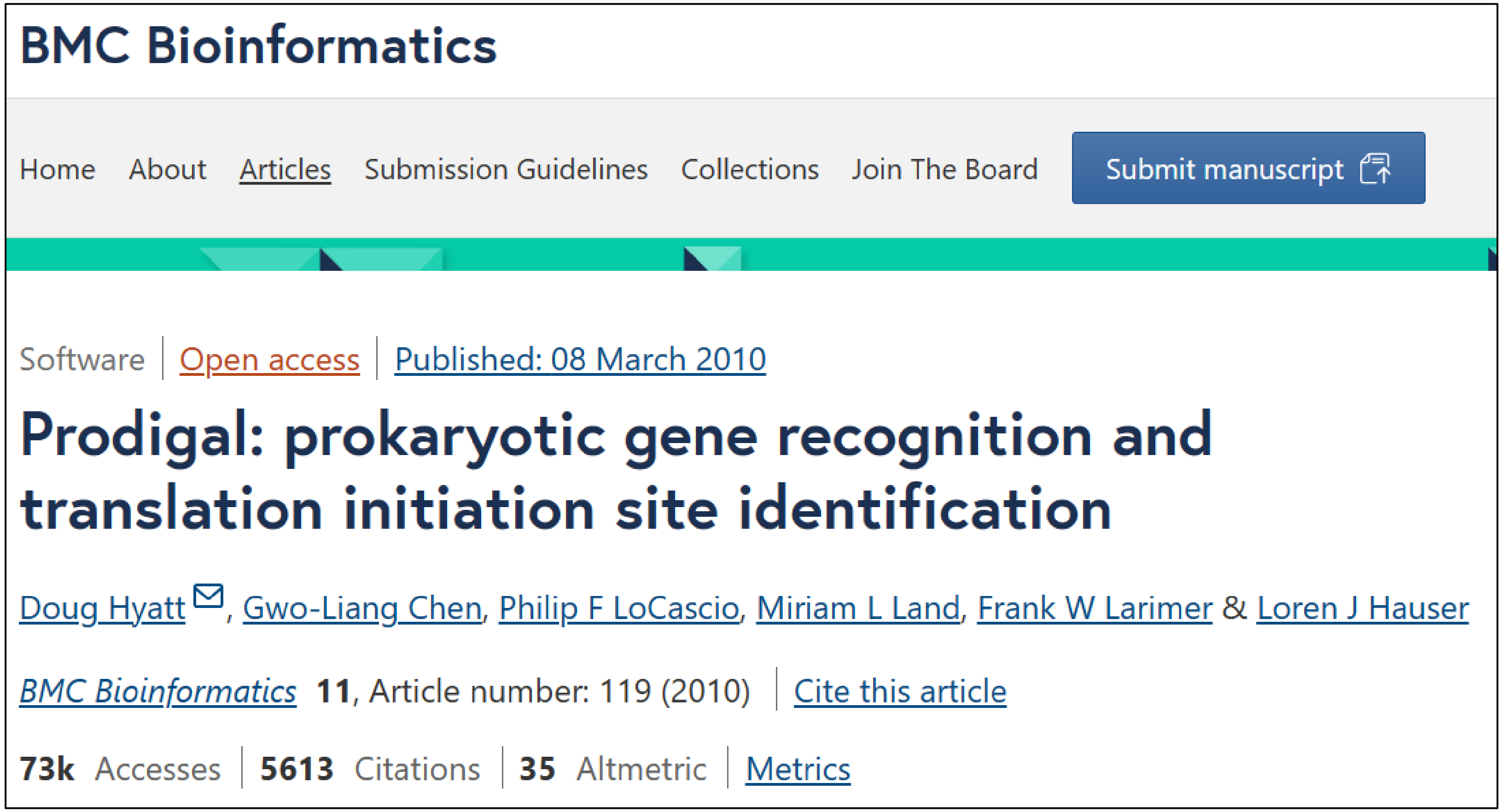
- In Linux system, installing programs is not the same as the Windows system
    - Especially if you are not the system supervisor

- Rather, it is our convention to “build” the program in our directory and just use it as normal

Build programs may be easy or difficult

- Most software packages has instruction on how to build the program
    - Or, you can also download the executable if you are sure that the OS and executable versions match

An easy example: prodigal



BMC Bioinformatics

Software | Open access | Published: 08 March 2010

Prodigal: prokaryotic gene recognition and translation initiation site identification

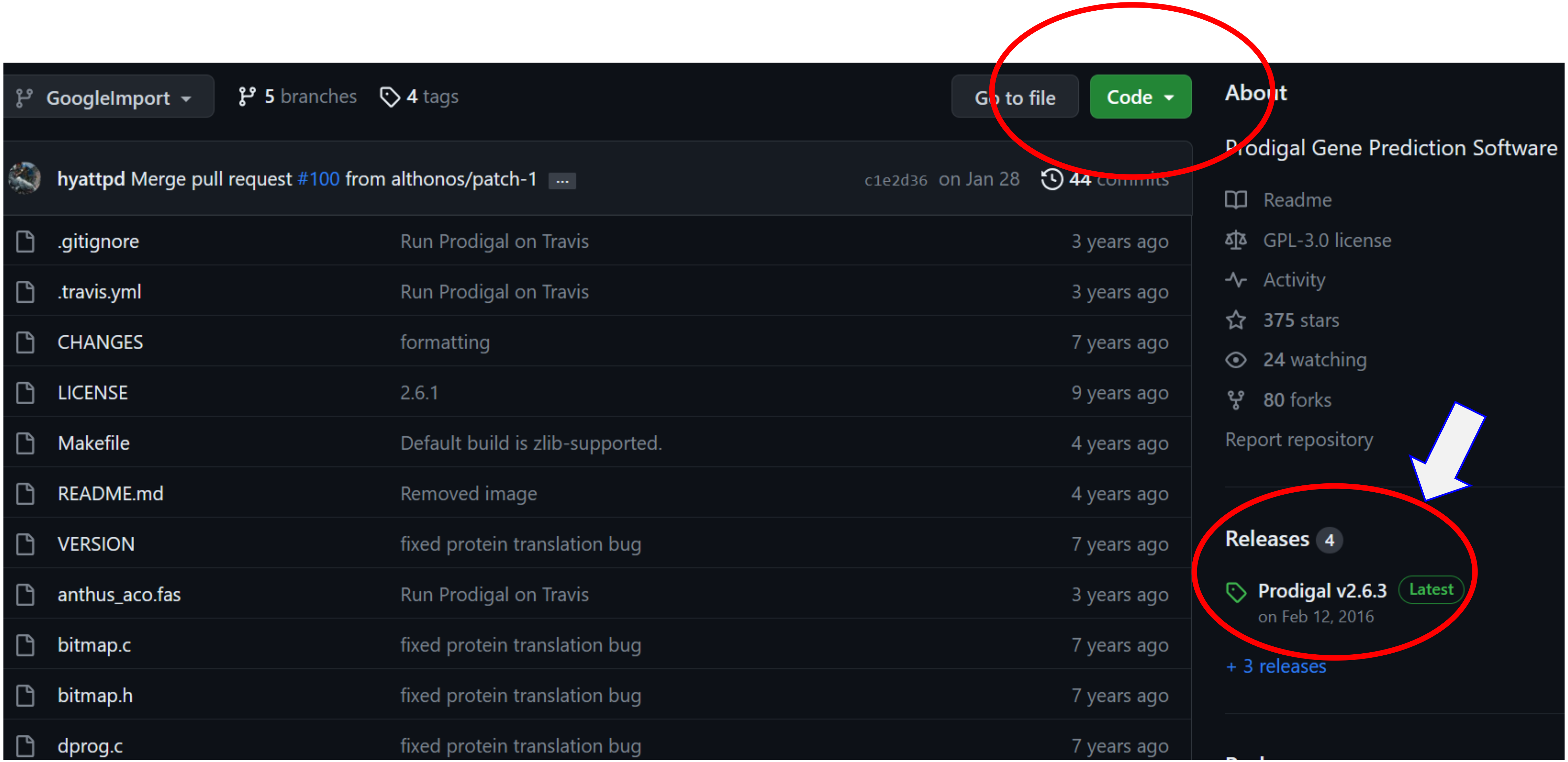
Doug Hyatt ✉, Gwo-Liang Chen, Philip F LoCascio, Miriam L Land, Frank W Larimer & Loren J Hauser

BMC Bioinformatics **11**, Article number: 119 (2010) | Cite this article

73k Accesses | **5613** Citations | **35** Altmetric | Metrics

Github

Can use “git clone” command to get the most current repo



GoogleImport ▾ 5 branches 4 tags

Go to file Code ▾

hyattpd Merge pull request #100 from althonos/patch-1 ... c1e2d36 on Jan 28 44 commits

.gitignore	Run Prodigal on Travis	3 years ago
.travis.yml	Run Prodigal on Travis	3 years ago
CHANGES	formatting	7 years ago
LICENSE	2.6.1	9 years ago
Makefile	Default build is zlib-supported.	4 years ago
README.md	Removed image	4 years ago
VERSION	fixed protein translation bug	7 years ago
anthus_aco.fas	Run Prodigal on Travis	3 years ago
bitmap.c	fixed protein translation bug	7 years ago
bitmap.h	fixed protein translation bug	7 years ago
dprog.c	fixed protein translation bug	7 years ago

About

Prodigal Gene Prediction Software

Readme

GPL-3.0 license

Activity

375 stars

24 watching

80 forks

Report repository

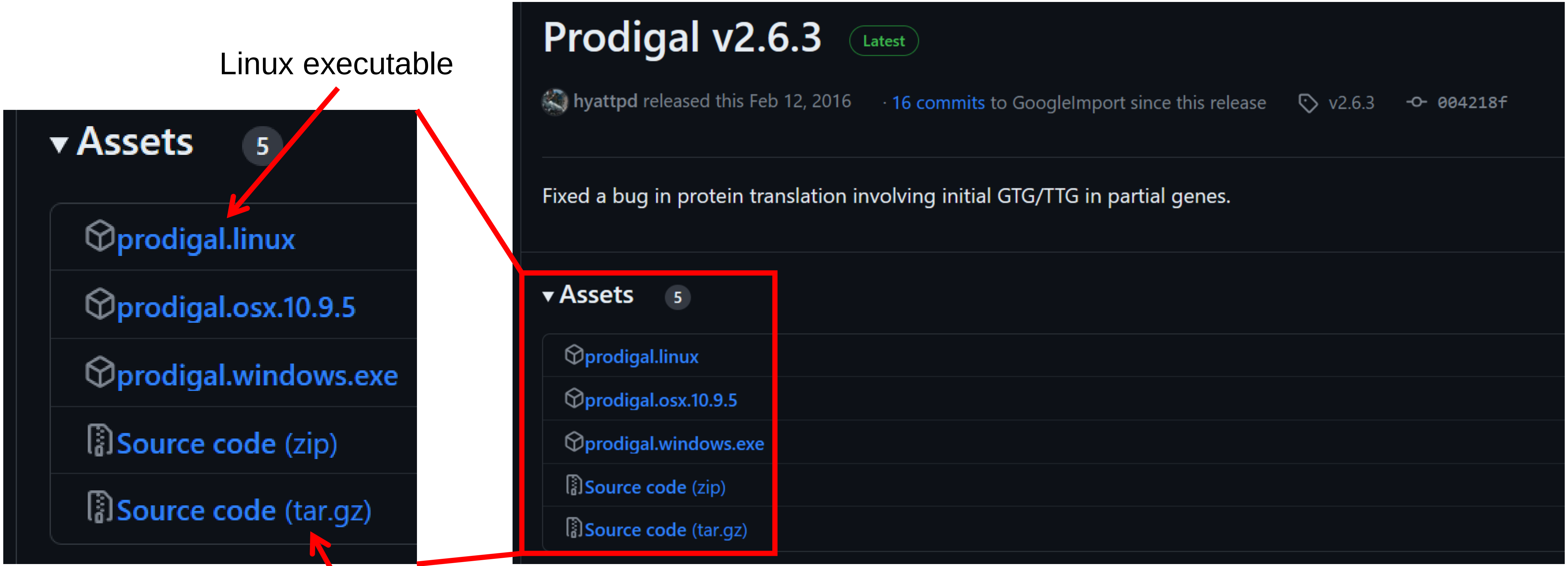
Releases 4

Prodigal v2.6.3 Latest
on Feb 12, 2016

+ 3 releases

Github releases

Linux executable



Prodigal v2.6.3 Latest

hyattpd released this Feb 12, 2016 · 16 commits to GoogleImport since this release v2.6.3 004218f

Fixed a bug in protein translation involving initial GTG/TTG in partial genes.

▾ Assets 5

- prodigal.linux
- prodigal.osx.10.9.5
- prodigal.windows.exe
- Source code (zip)
- Source code (tar.gz)

▾ Assets 5

- prodigal.linux
- prodigal.osx.10.9.5
- prodigal.windows.exe
- Source code (zip)
- Source code (tar.gz)

Source code

Method 1: get the executable (easy)

- Here are the commands to get the prodigal linux executable

```
$ cd (your preferred place for placing the executable)
$ wget https://github.com/hyattpd/Prodigal/releases/download/v2.6.3/prodigal.linux
$ mv prodigal.lilnux prodigal # rename it to prodigal
$ chmod 0755 prodigal.linux # add execution permission
```

Method 2: get the executable (also easy…sort of)

- Here are the commands to “make” the prodigal linux executable

```
$ wget https://github.com/hyattpd/Prodigal/archive/refs/tags/v2.6.3.tar.gz
$ tar zxvf v2.6.3.tar.gz
$ cd Prodigal-2.6.3/
$ make
```

Outcome

- From both method 1 and method 2 you will see that an executable file “prodigal”
- To run the program, simply type “prodigal” or “./prodigal” at the program directory

This “./” means that I want to run a program from my current directory

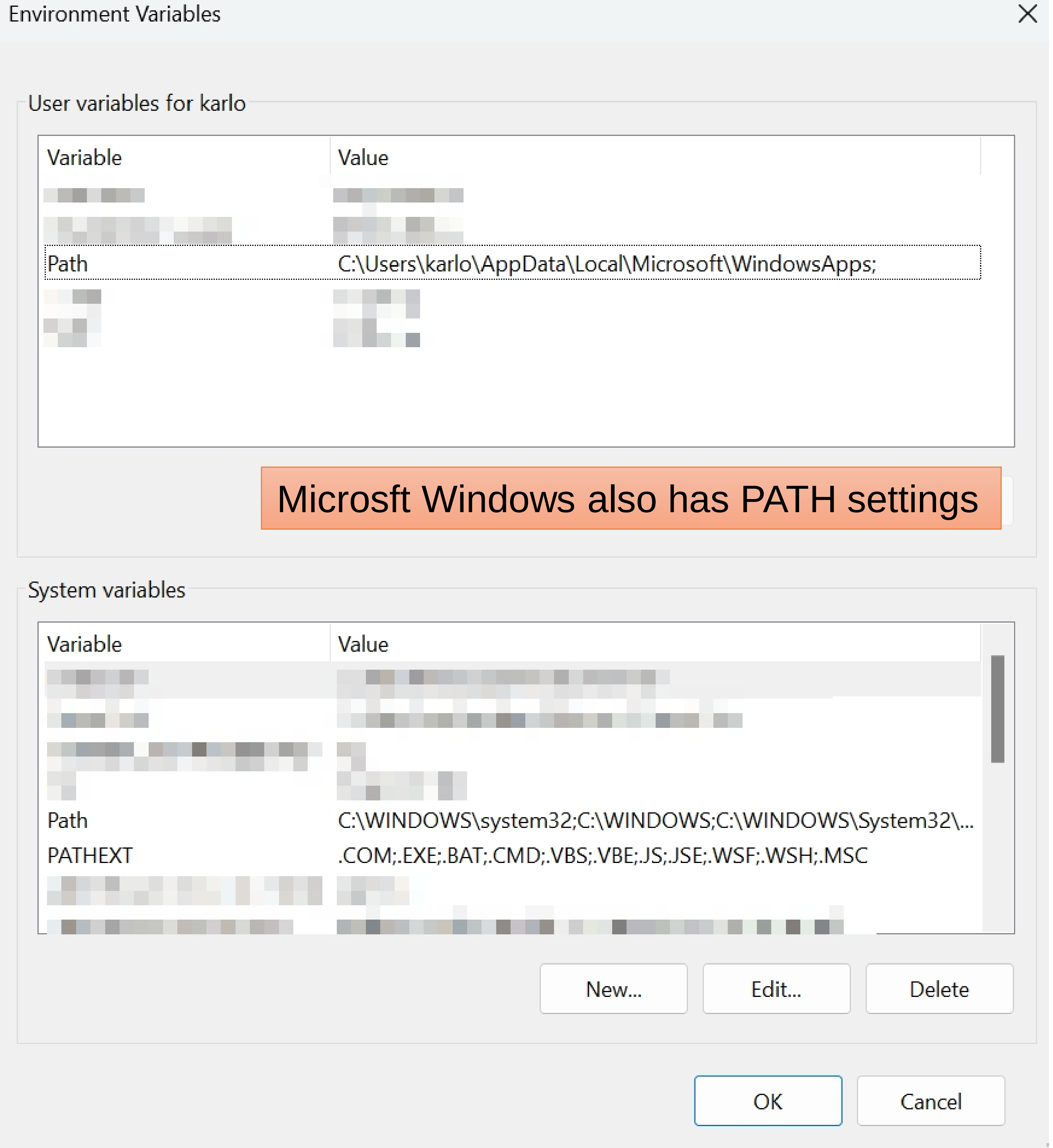
Installing program?

- Usually, if you have root privilege, you can install the program into the system by typing “sudo make install”
    - The executable will be copied into “/usr/bin” or “/usr/local/bin”, depending on the make settings

- However, if you do not have root permission, we can still make the program runnable from everywhere in the system by adding something into system **PATH**

System path

- Just think about this: how to run program EVERYWHERE in the system?
- The system must have kept a PATH such that it can find programs in specific places.



Environment Variables

User variables for karlo

Variable	Value
[redacted]	[redacted]
[redacted]	[redacted]
Path	C:\Users\karlo\AppData\Local\Microsoft\WindowsApps;
[redacted]	[redacted]

Microsft Windows also has PATH settings

System variables

Variable	Value
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
Path	C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
[redacted]	[redacted]
[redacted]	[redacted]

Add path into system PATH

- Edit the file “**.bash_aliases**”, which may or may not exist in the beginning

Note the “dot” here

- Add the directory consisting of our program
- For example, assuming I have a program at “/home/yuwei/bin/abc” and another at “/home/yuwwu/bin/xyz”, then I can add the following line into .bash_aliases file

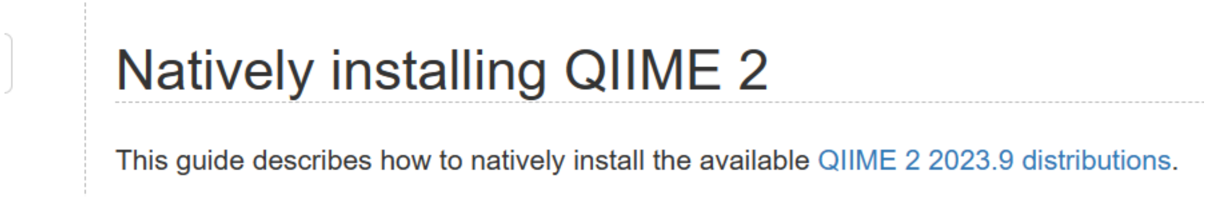
```
export PATH=$PATH:/home/yuwei/bin/abc;/home/yuwei/bin/xyz
```

- Then logout and login again to activate the PATH setting

Other software usage options - 1

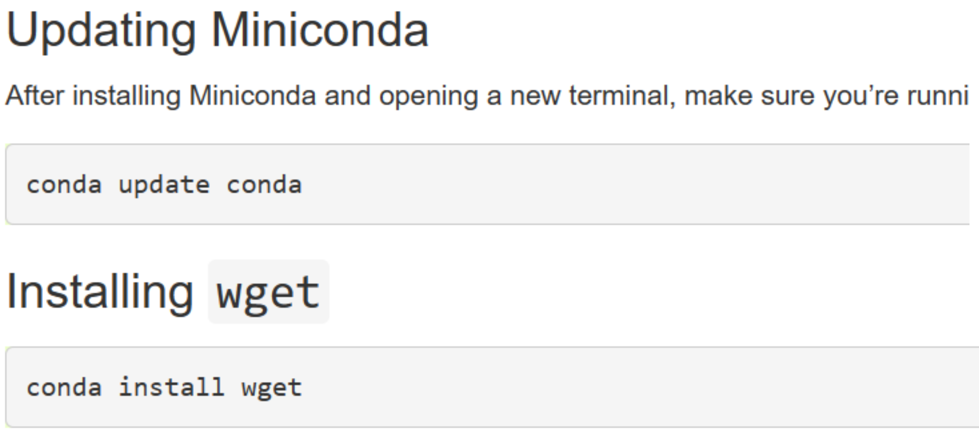
- Anaconda or miniconda (conda) – follow developer’s instruction on how to install packages (e.g. qiime2)

qiime2docs https://docs.qiime2.org/2023.9/install/native/



Natively installing QIIME 2

This guide describes how to natively install the available QIIME 2 2023.9 distributions.



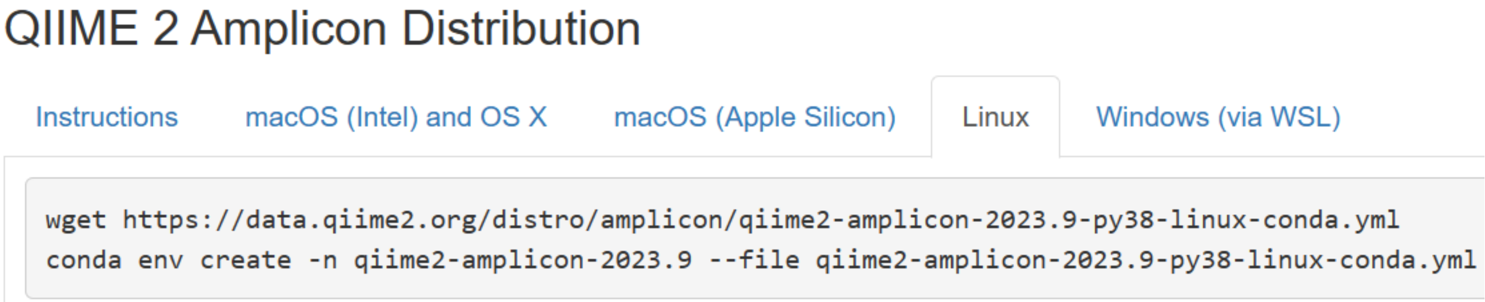
Updating Miniconda

After installing Miniconda and opening a new terminal, make sure you’re runni

```
conda update conda
```

Installing wget

```
conda install wget
```

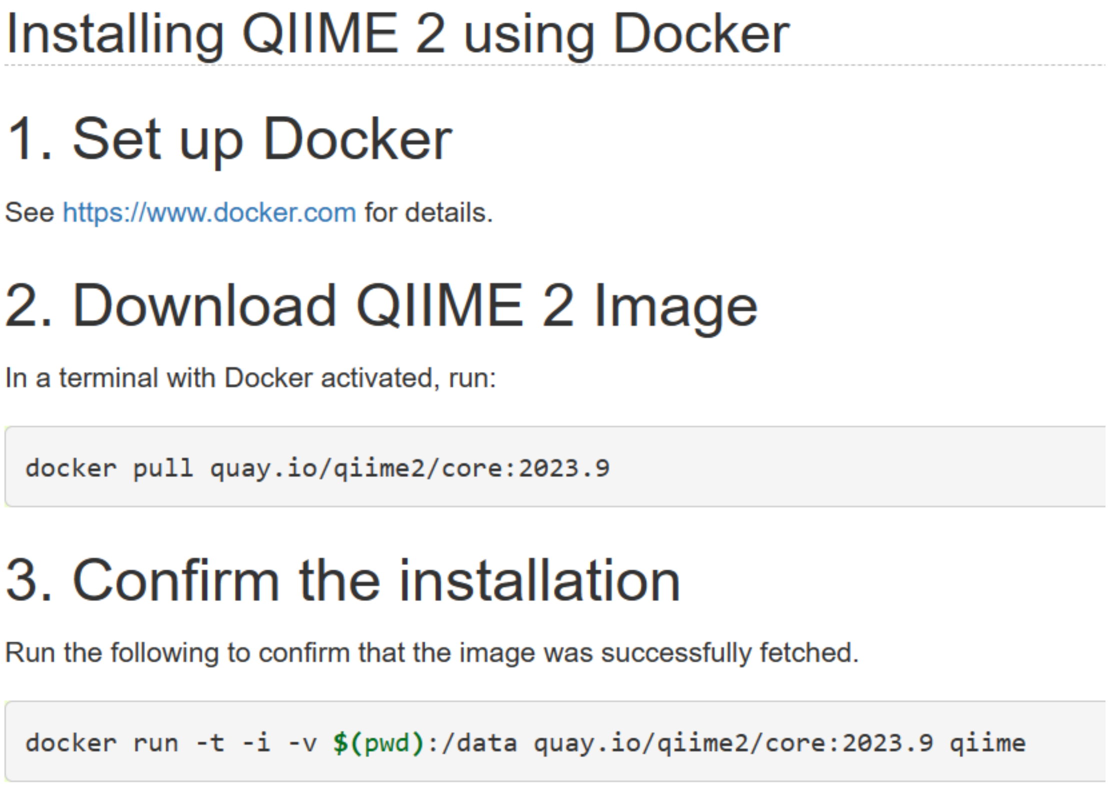


QIIME 2 Amplicon Distribution

```
wget https://data.qiime2.org/distro/amplicon/qiime2-amplicon-2023.9-py38-linux-conda.yml
conda env create -n qiime2-amplicon-2023.9 --file qiime2-amplicon-2023.9-py38-linux-conda.yml
```

Other software usage options - 2

- Docker – developers install everything into a linux image that you can use directly. Again take qiime2 as an example.



Installing QIIME 2 using Docker

1. Set up Docker

See https://www.docker.com for details.

2. Download QIIME 2 Image

In a terminal with Docker activated, run:

```
docker pull quay.io/qiime2/core:2023.9
```

3. Confirm the installation

Run the following to confirm that the image was successfully fetched.

```
docker run -t -i -v $(pwd):/data quay.io/qiime2/core:2023.9 qiime
```

Docker is very convenient. However, since it allows everyone using docker to have system-level permission, system administration does not like this approach.

Other software usage options - 3

- Singularity – save docker image as a file and run the file as if you are using docker.
- Very safe system-wide!

Why bother docker or singularity?

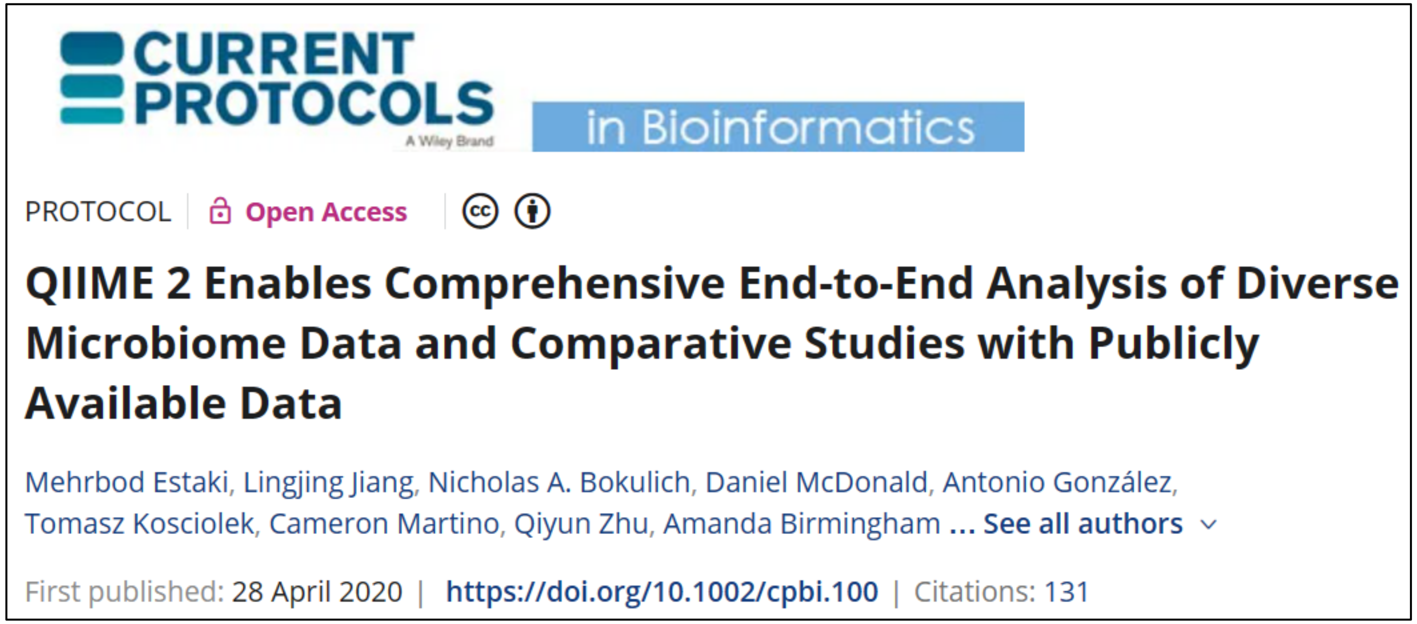
- Because installing software may sometimes be very complicated.
- For example, a software package, “RepeatMasker”, requires 6-10 other packages also installed in the system. Or another package, “CheckM”, requires at least 3 other packages as well as Python packages with specific versions.

- So having an image can be very helpful

Back to microbiome

- I won’t talk too much about 16S-based analysis, as the guidelines outlined on qiime2 and their Current Protocols paper is already very good

Tutorials



PROTOCOL | Open Access

QIIME 2 Enables Comprehensive End-to-End Analysis of Diverse Microbiome Data and Comparative Studies with Publicly Available Data

Mehrbod Estaki, Lingjing Jiang, Nicholas A. Bokulich, Daniel McDonald, Antonio González, Tomasz Kosciolek, Cameron Martino, Qiyun Zhu, Amanda Birmingham ... See all authors

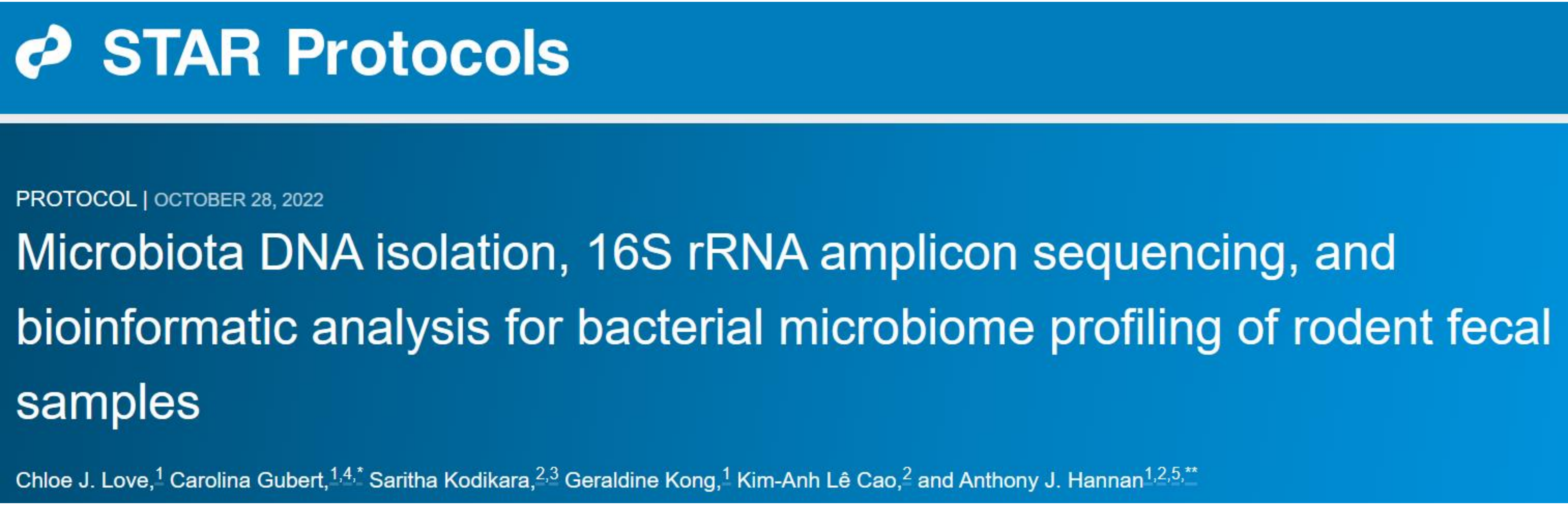
First published: 28 April 2020 | https://doi.org/10.1002/cpbi.100 | Citations: 131

How to improve your PCA plot?

- I only want to talk about one thing:

- “What to do if your PCA or NMDS plots are not looking good?”

An example paper



STAR Protocols

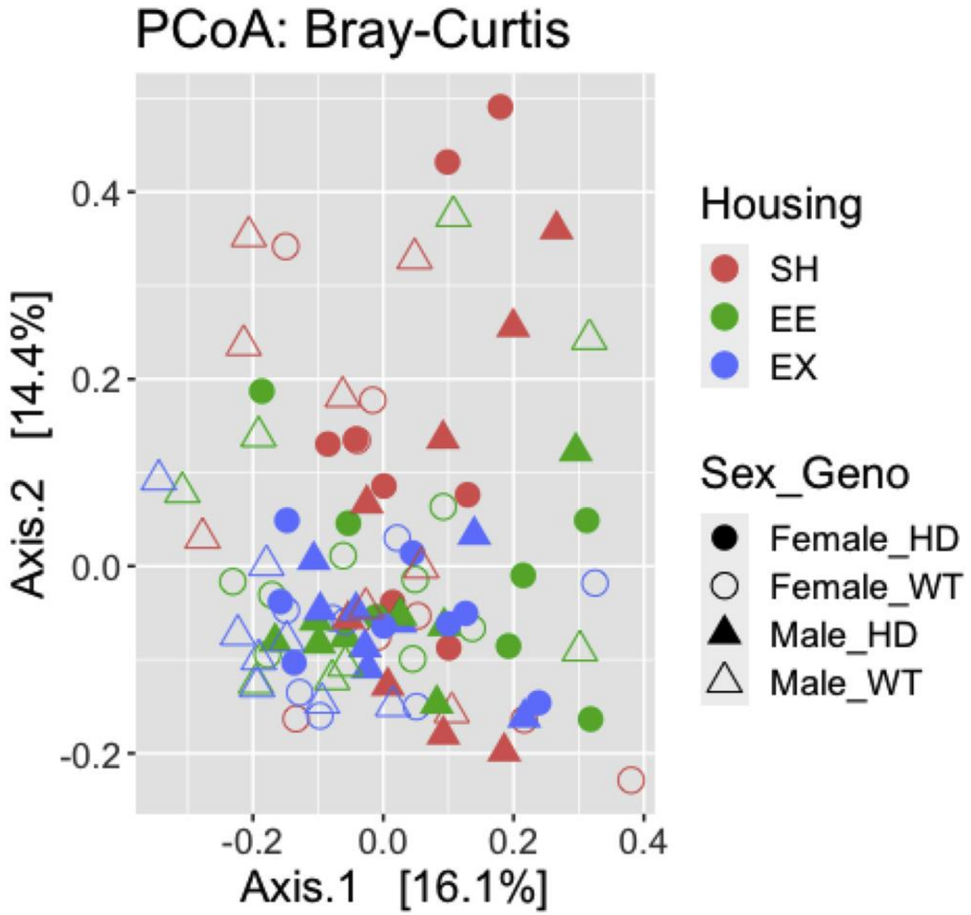
PROTOCOL | OCTOBER 28, 2022

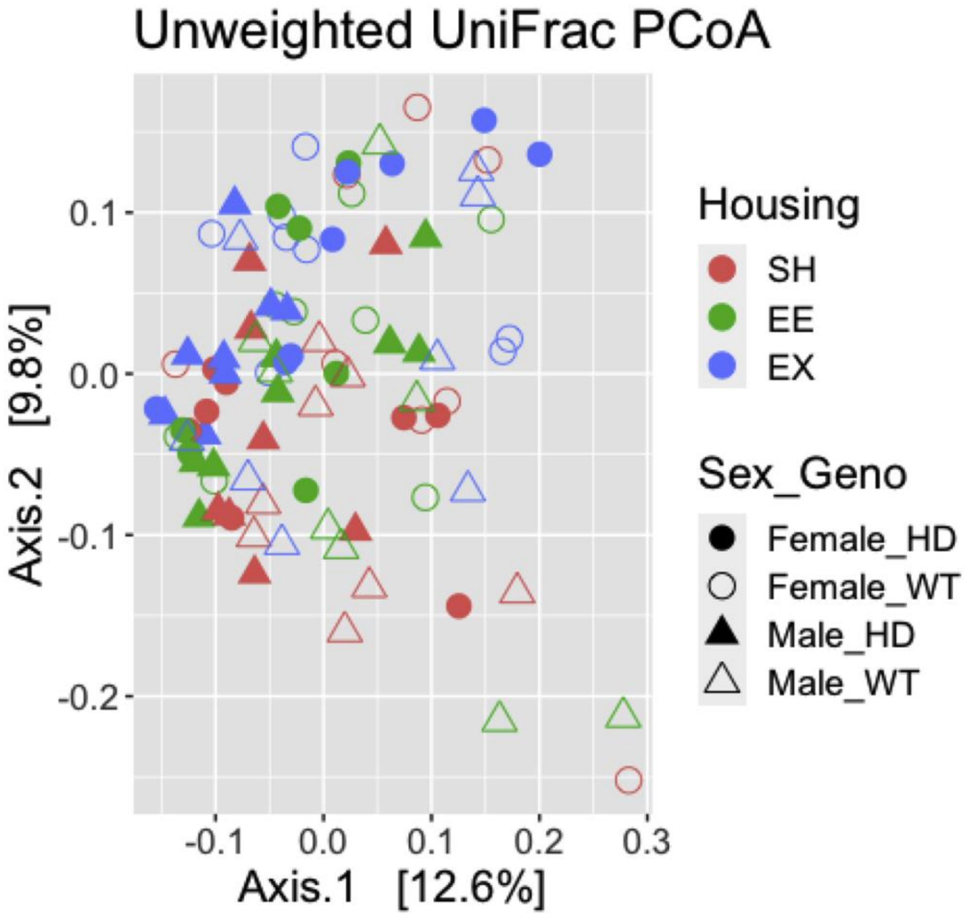
Microbiota DNA isolation, 16S rRNA amplicon sequencing, and bioinformatic analysis for bacterial microbiome profiling of rodent fecal samples

Chloe J. Love,[1] Carolina Gubert,[1,4,*] Saritha Kodikara,[2,3] Geraldine Kong,[1] Kim-Anh Lê Cao,[2] and Anthony J. Hannan[1,2,5,**]

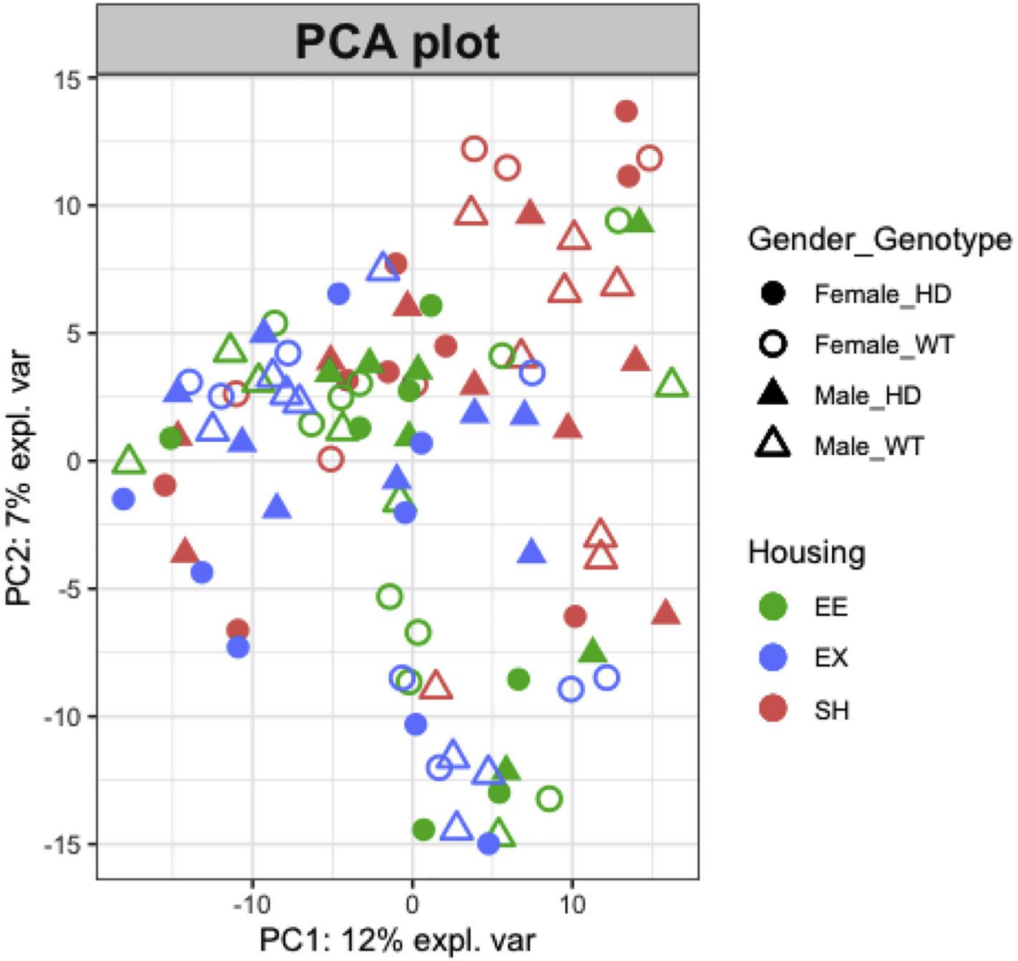
This paper is about the exact bioinformatics steps for analyzing rodent fecal microbiome. But it also has steps for “improving” PCA plots

PCA plots made with all taxonomic units (ASV)

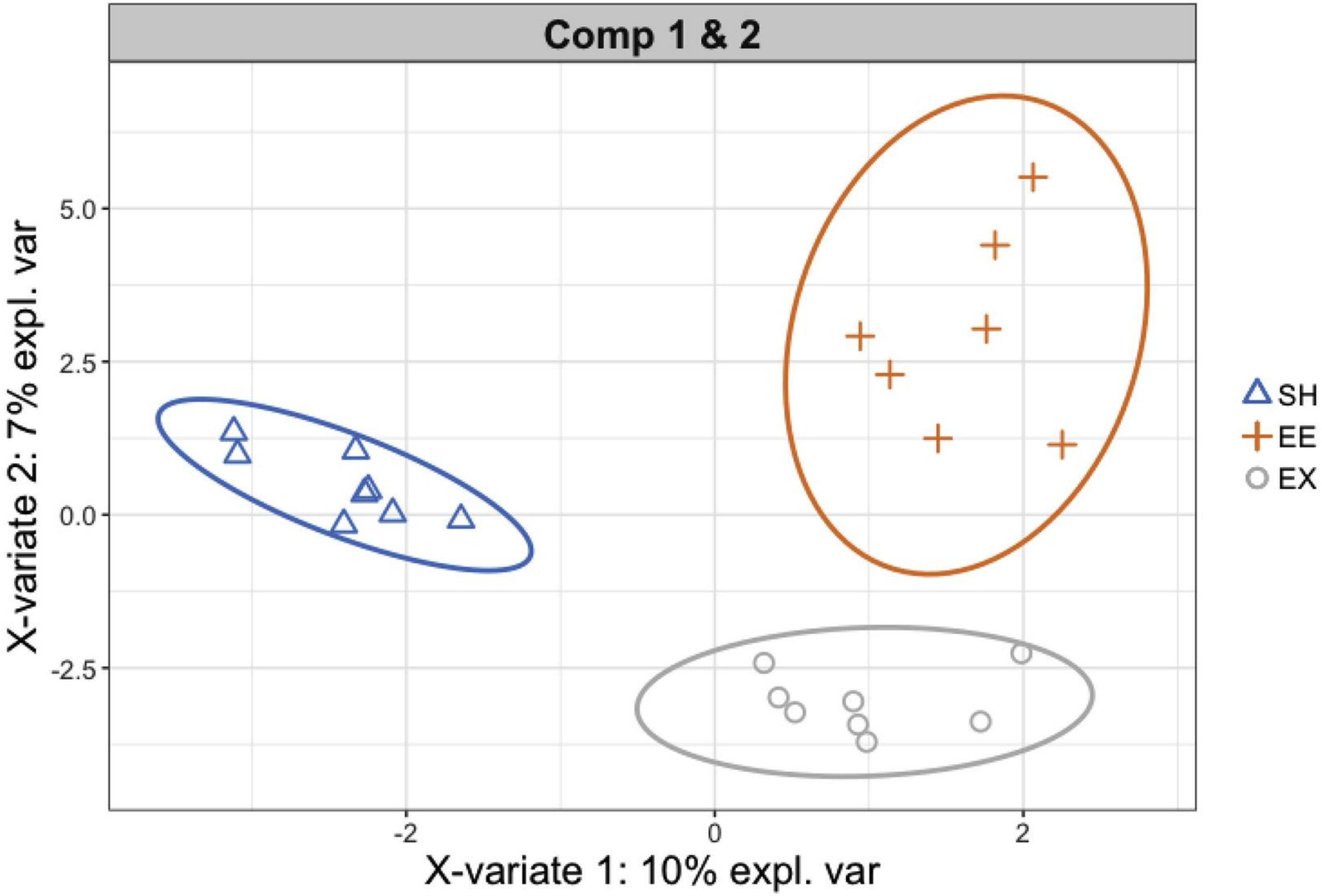




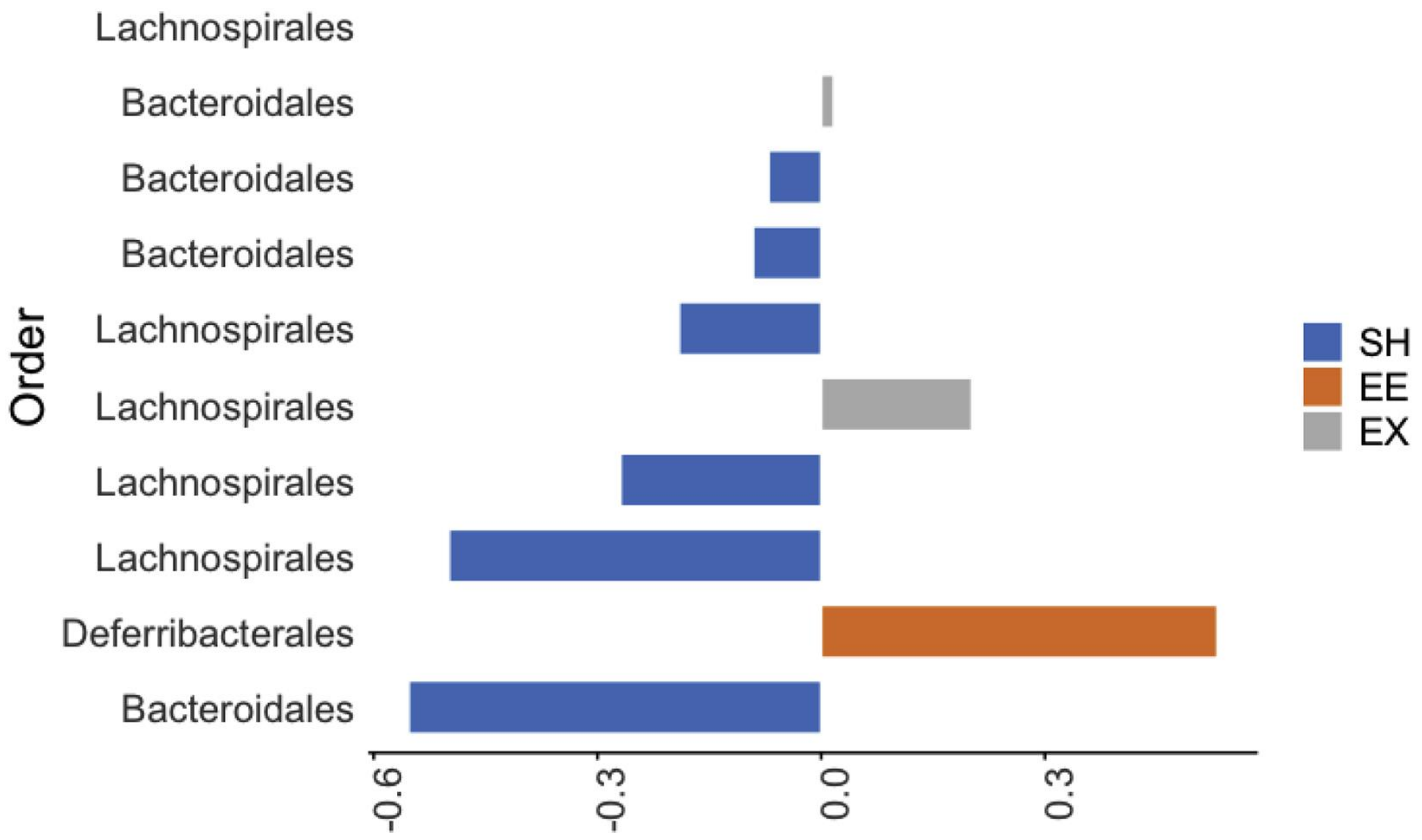
PCA plots made with ASV > 0.01%



PCA after selecting only highly-relevant ASVs

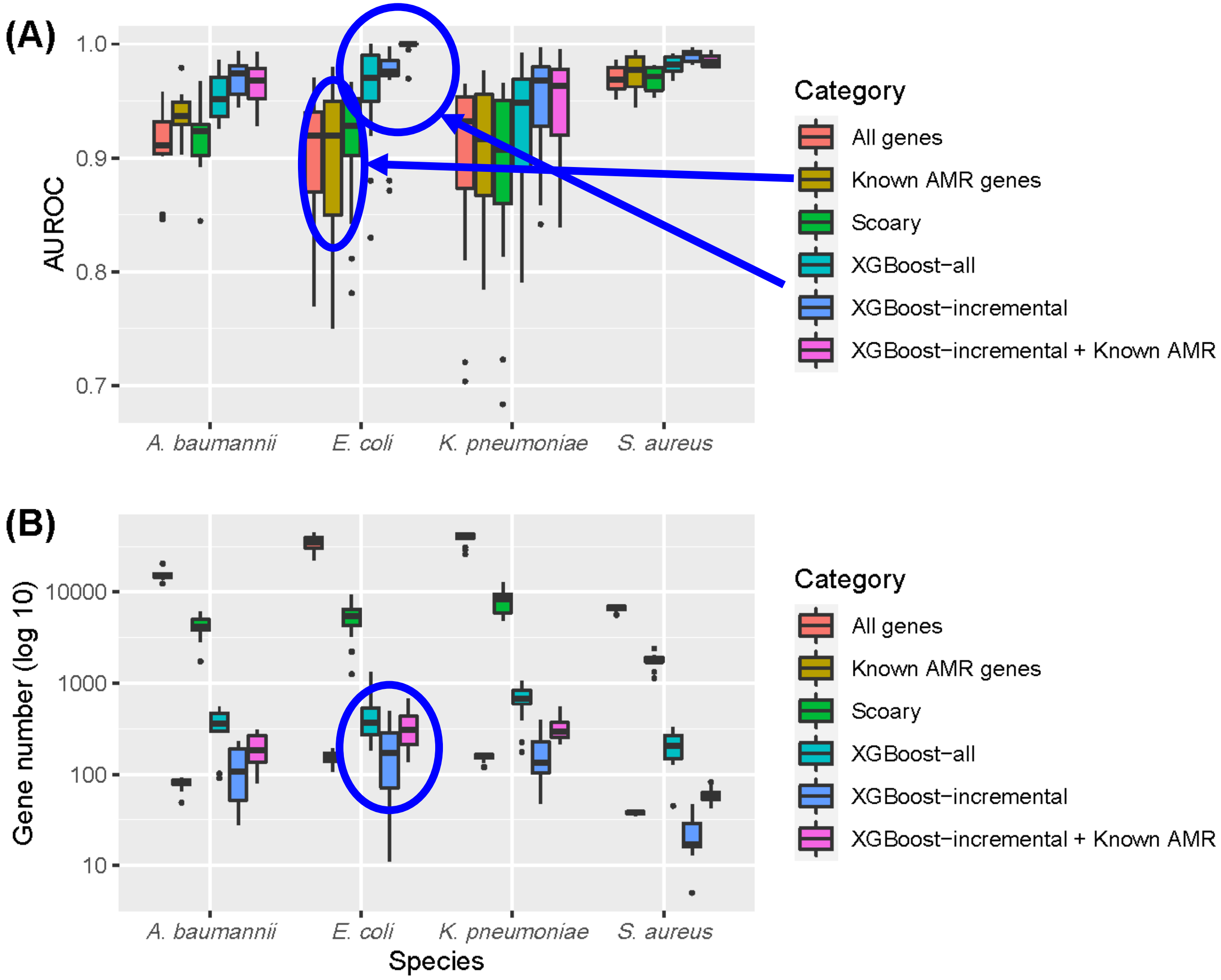


Relevant orders and relevance scores



This is called “feature selection”

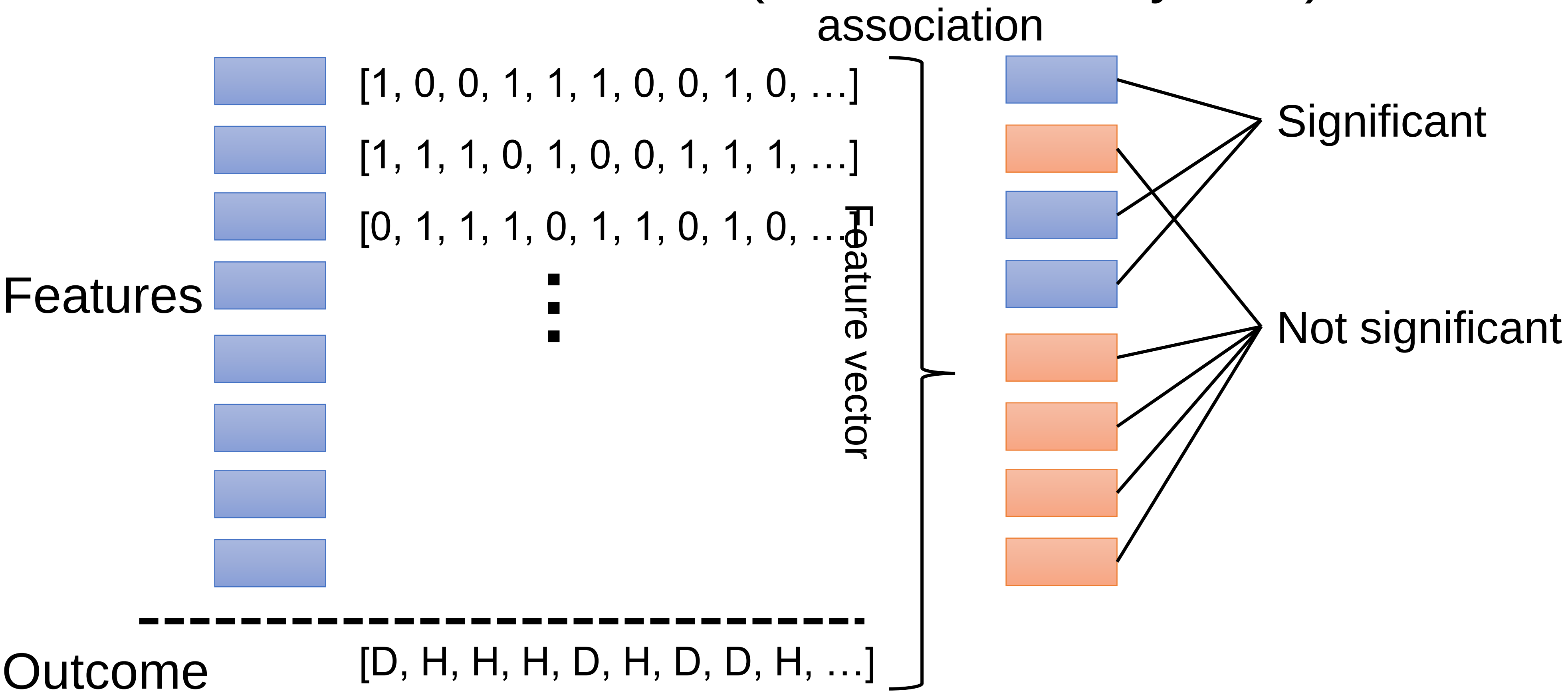
- Commonly used in the machine learning world to find crucial features that best predict the outcomes
- For example, in our work on antimicrobial resistances using the presence/absence patterns of genes, we found that selecting a hundred or so genes achieves much better prediction outcome than the entire collection of tens of thousands of genes



(Yang 2022 BMC Bioinformatics)

A simple rationale for feature selection

- Looking for features (e.g. genes) that are significantly associated with final outcome (disease/healthy, etc.)



Feature selection…in other words

- You may also think of feature selection as the process of “noise removal”

- The prediction, clustering, and classification performances will likely be improved after feature selection

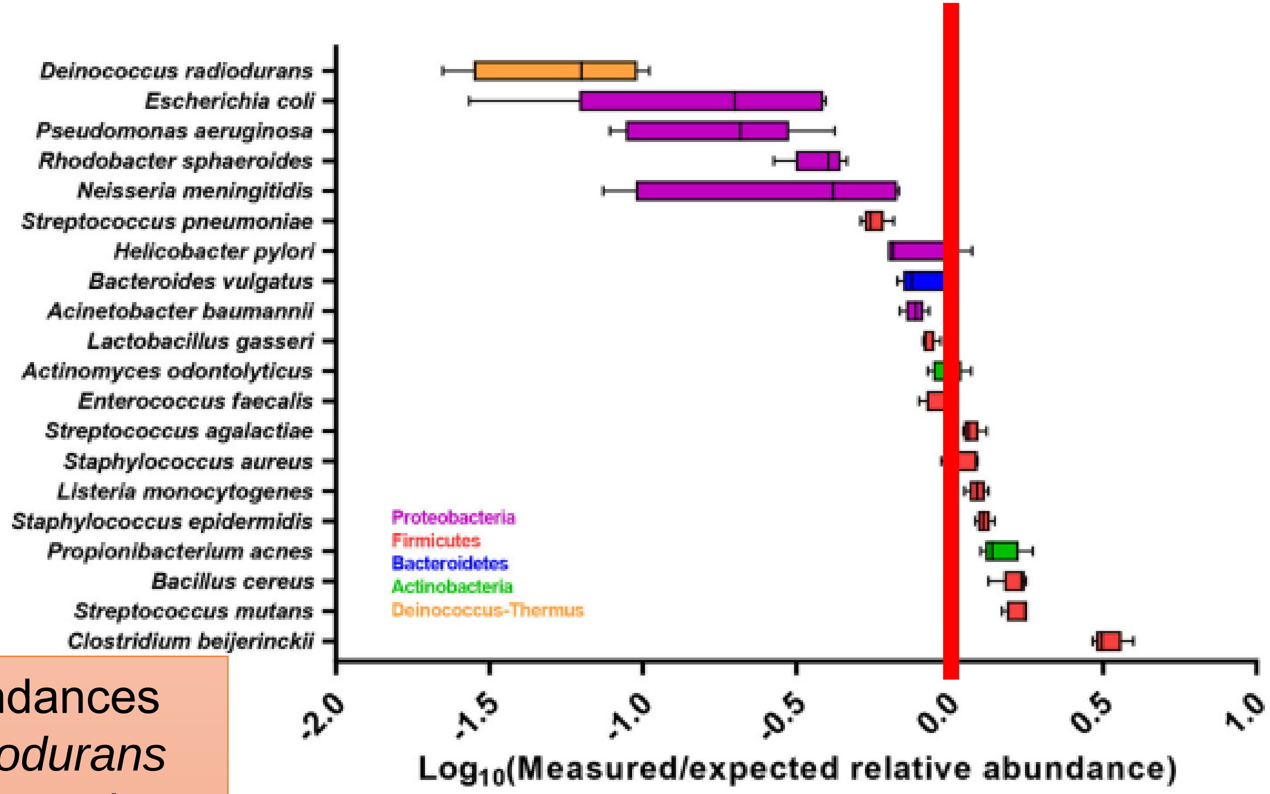
Pros and Cons for 16S-based analysis

- Pros
  - **Inexpensive**
  - Easy to **analyze** and **quantify** (by comparing sequences against those in *databases*)

- Cons
  - **Primer** may miss some 16S genes from unknown bacteria
  - The **amplification** process may create biases
  - Have no way to understand the **functional potentials** of the organisms

PCR amplification “bias”

- The majority of estimated organism abundances deviate from the actual abundances by orders of magnitude

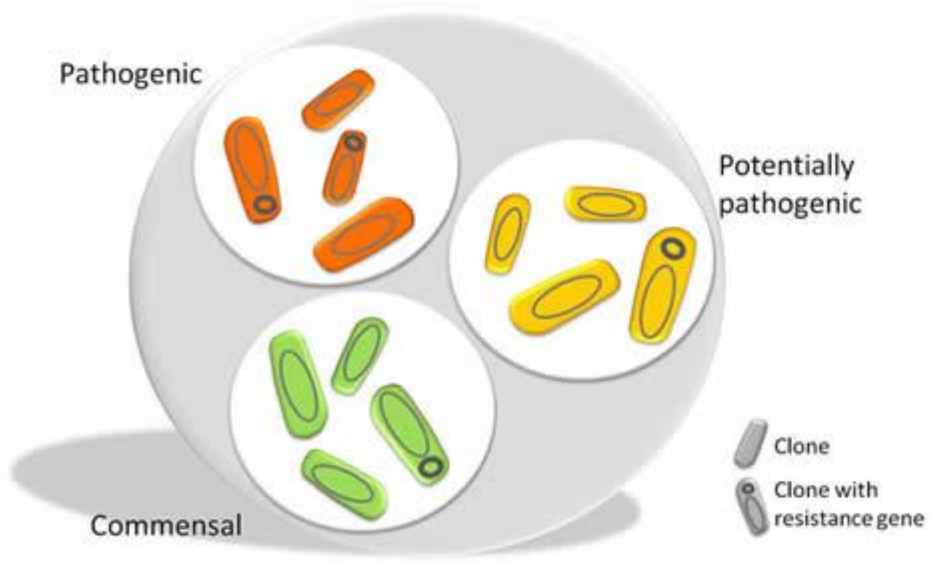


“Generally, the determined relative abundances of *Proteobacteria* and *Deinococcus radiodurans* were underestimated, whereas those of species within Firmicutes (especially*. beijerinckii*) were mostly overestimated compared with the expected community composition of 5% for each species.”

https://www.frontiersin.org/articles/10.3389/fmicb.2017.01934/full

Also 16S sequences cannot be used to identify “strain variation”

- Some bacterial strains have diverse functions
  - Commensal and pathogenic *E. coli*
  - Drug-resistant or susceptible *Klebsiella pneumonia*



http://www.ecl-lab.com/en/ecoli/index.asp

Shotgun metagenome

- Instead of getting and amplifying just one gene (16S rRNA) among the microbial population, the shotgun metagenome seeks to sequence EVERYTHING

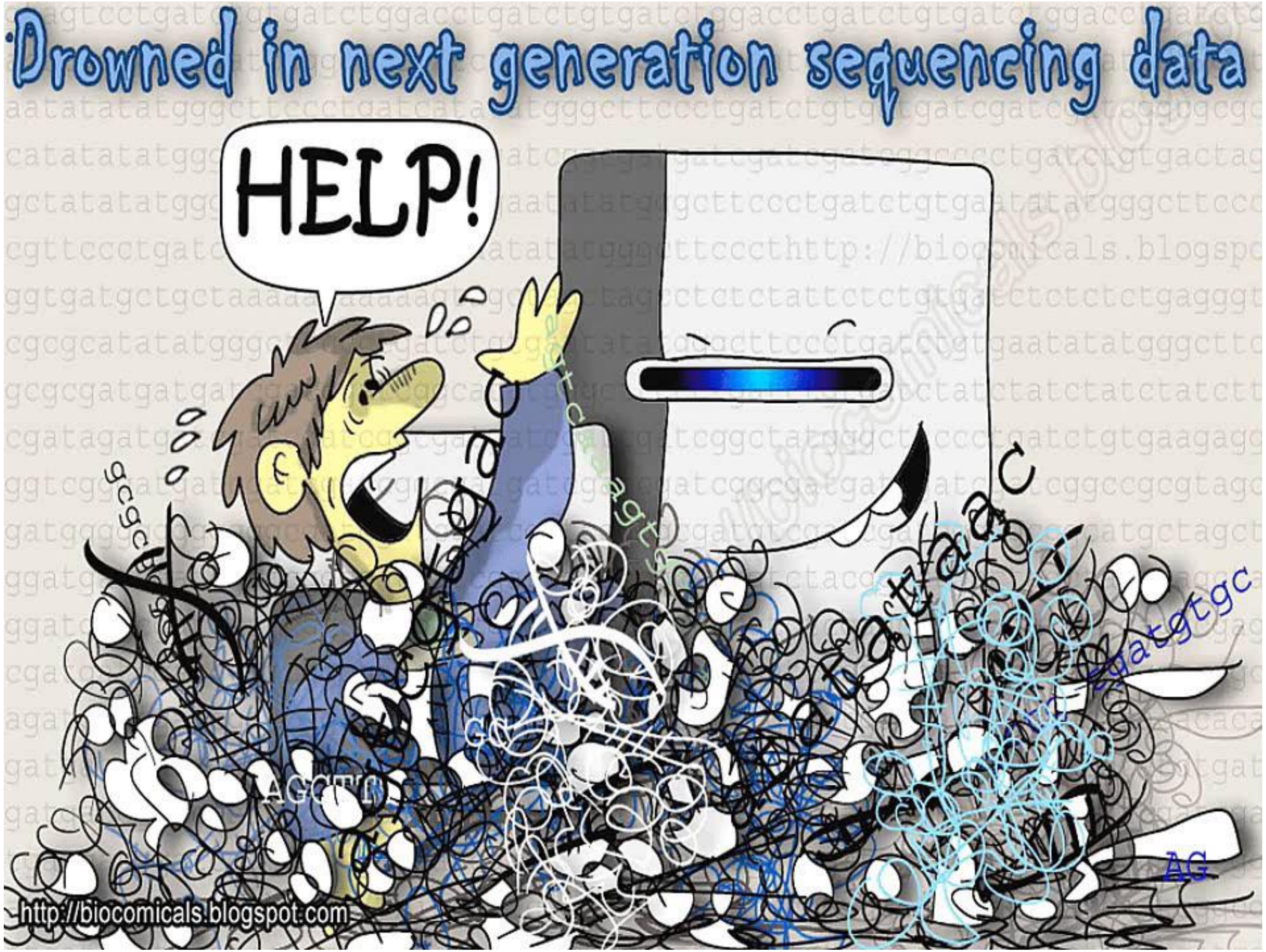
The good, the bad, and the ugly

We have everything!

It’s TOO complicated!

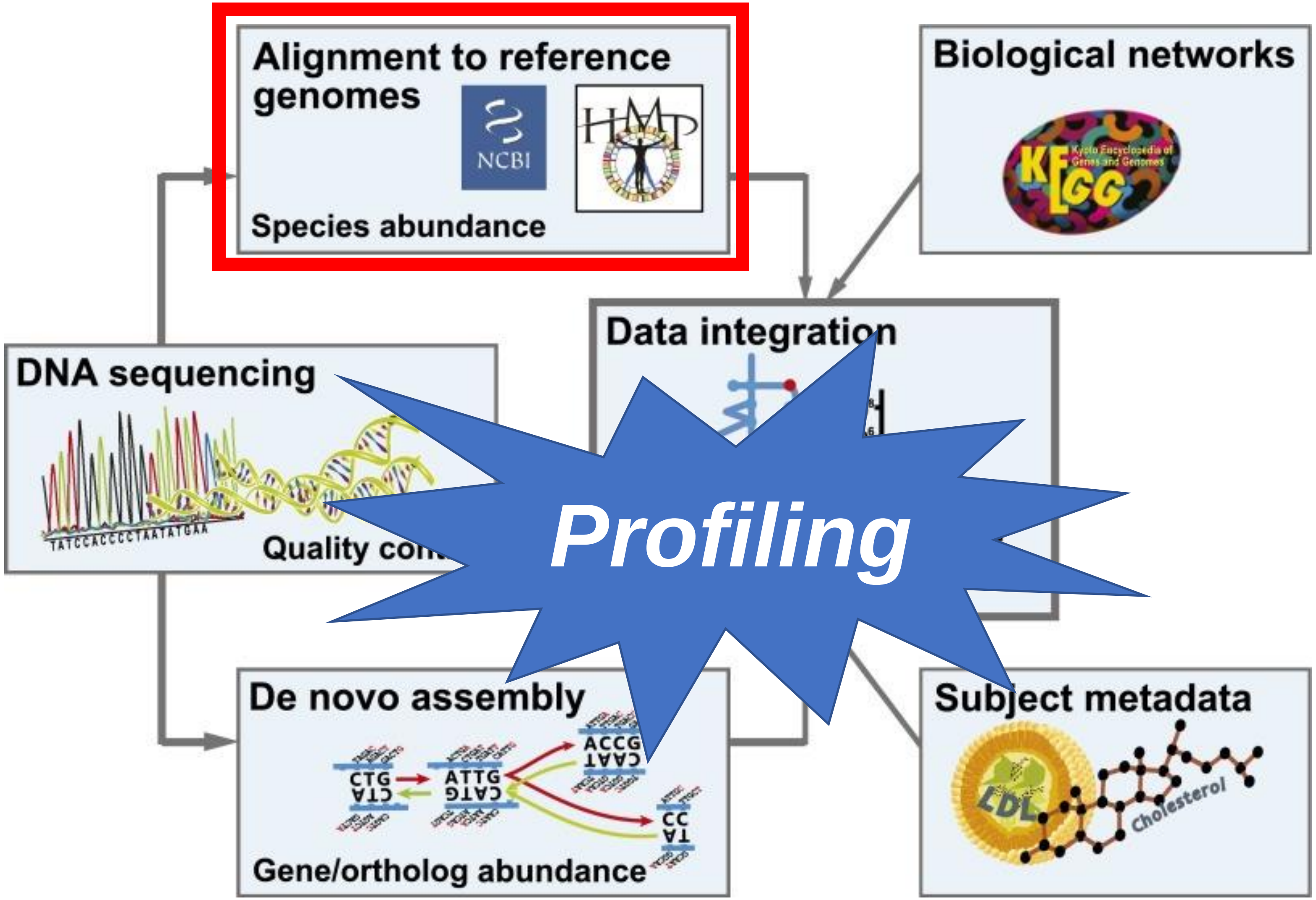
PHASE TWO : INTERPRETATION
I THINK I FOUND A CORNER PIECE
3 BILLION PIECES
GENOME

http://biobeans.blogspot.com/2012/11/bioinformatics-genome-assembly.html
https://www.123rf.com/photo_16591603_cute-angel-cartoon.html
https://www.vectorstock.com/royalty-free-vector/cartoon-demon-vector-2806309


Drowned in next generation sequencing data
HELP!
http://biocomicals.blogspot.com
AG

https://www.youtube.com/watch?v=Nap6AJb31Kc

Common data processing methods



https://www.researchgate.net/publication/257072074_Assessing_the_Human_Gut_Microbiota_in_Metabolic_Diseases/figures?lo=1&utm_source=google&utm_medium=organic

Database!

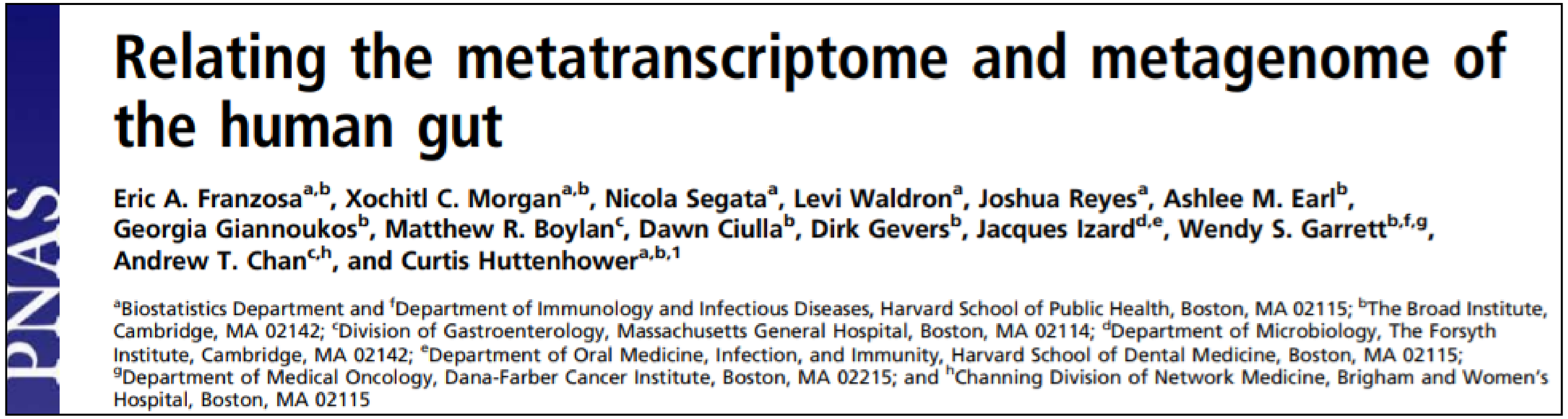
Key to the success of this workflow is…

https://0x30.io/databases-per-use-type/

Database advantages

There are four apparent advantages for using a database-based method:

1. **Fast**
2. **Easy-to-handle**
3. **Making things comparable**
4. **Model training**

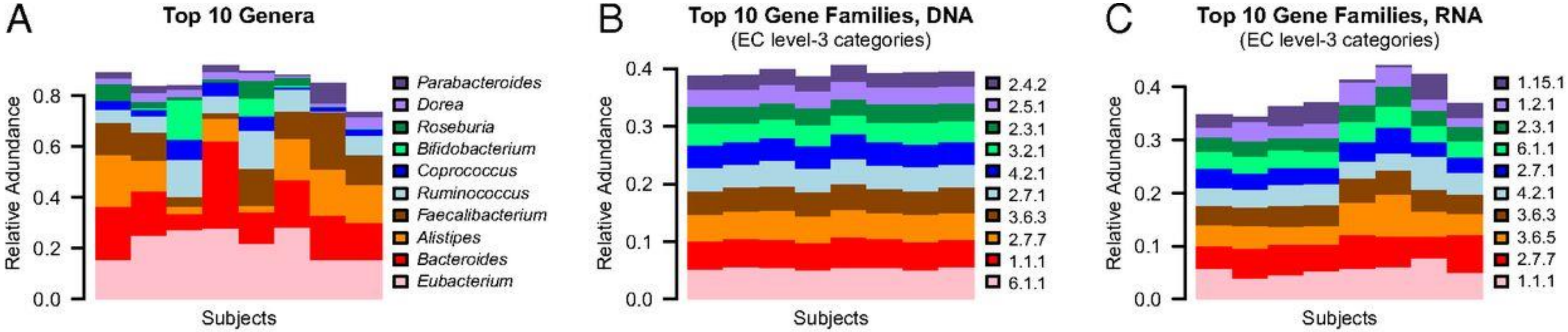


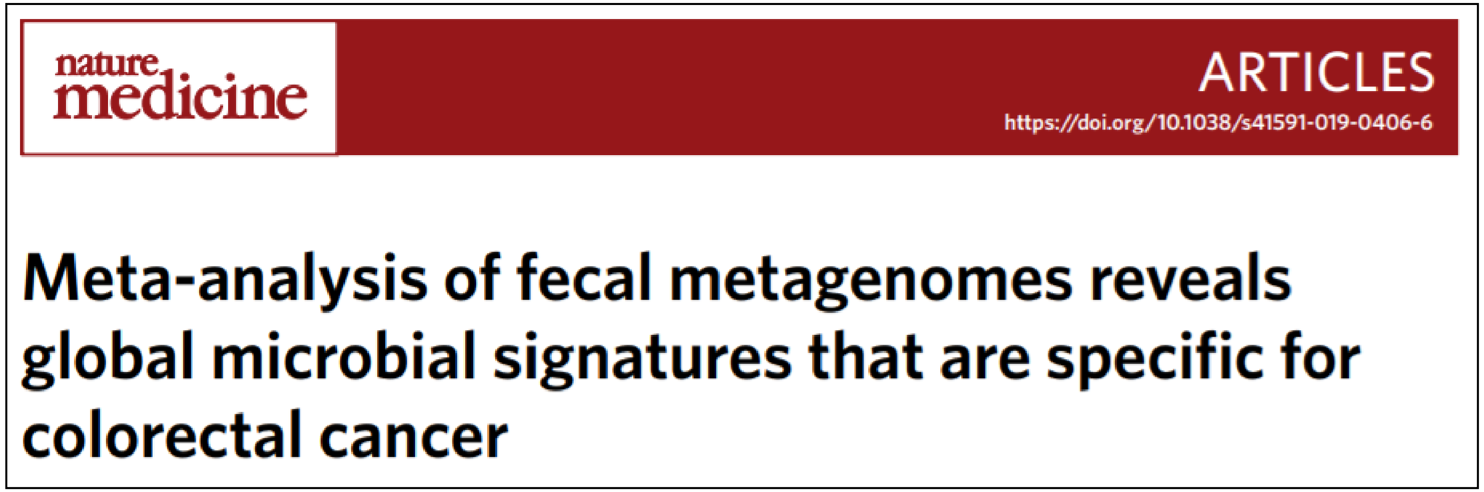
Relating the metatranscriptome and metagenome of the human gut

Eric A. Franzosa[a,b], Xochitl C. Morgan[a,b], Nicola Segata[a], Levi Waldron[a], Joshua Reyes[a], Ashlee M. Earl[b], Georgia Giannoukos[b], Matthew R. Boylan[c], Dawn Ciulla[b], Dirk Gevers[b], Jacques Izard[d,e], Wendy S. Garrett[b,f,g], Andrew T. Chan[c,h], and Curtis Huttenhower[a,b,1]

[a]Biostatistics Department and [f]Department of Immunology and Infectious Diseases, Harvard School of Public Health, Boston, MA 02115; [b]The Broad Institute, Cambridge, MA 02142; [c]Division of Gastroenterology, Massachusetts General Hospital, Boston, MA 02114; [d]Department of Microbiology, The Forsyth Institute, Cambridge, MA 02142; [e]Department of Oral Medicine, Infection, and Immunity, Harvard School of Dental Medicine, Boston, MA 02115; [g]Department of Medical Oncology, Dana-Farber Cancer Institute, Boston, MA 02215; and [h]Channing Division of Network Medicine, Brigham and Women's Hospital, Boston, MA 02115

Metaphlan, a marker-gene-based profiling tool, was used to to cross-compare samples





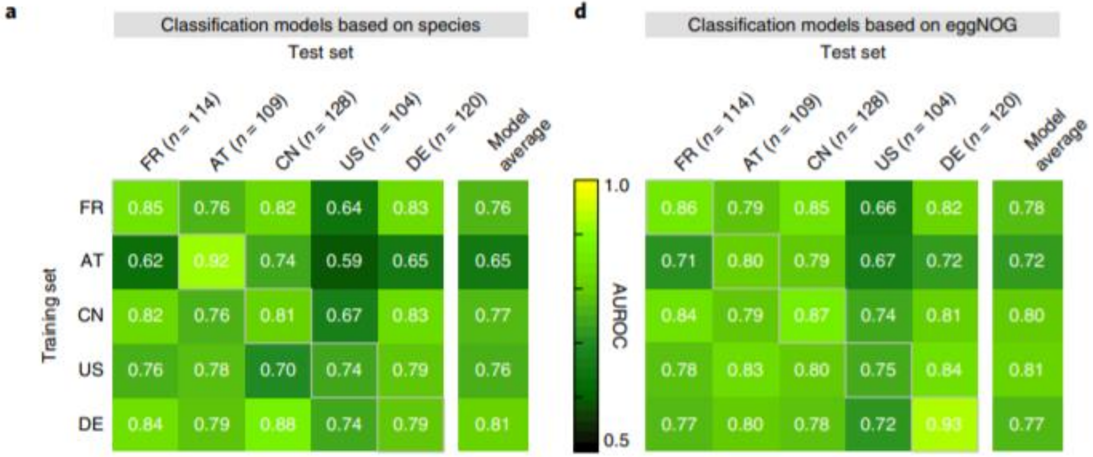
nature medicine

ARTICLES

https://doi.org/10.1038/s41591-019-0406-6

Meta-analysis of fecal metagenomes reveals global microbial signatures that are specific for colorectal cancer

mOTU, another marker gene-based profiling tool, was utilized in comparing different studies.



Classification models based on species

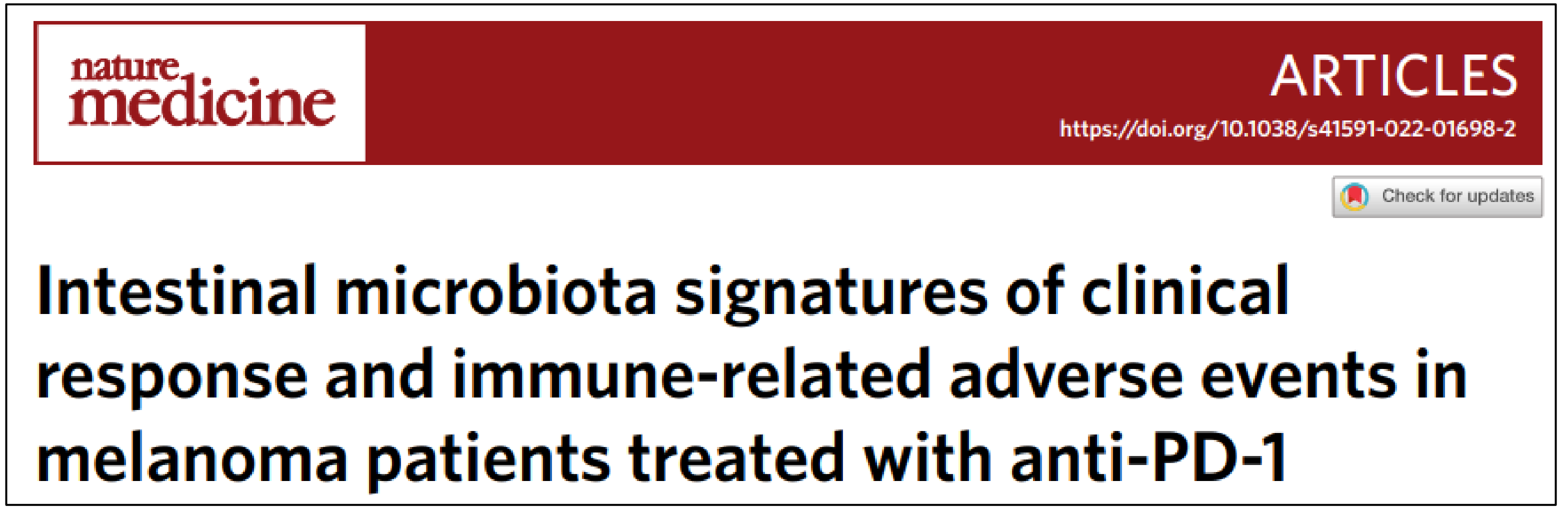
Test set

Training set	FR (*n* = 114)	AT (*n* = 109)	CN (*n* = 128)	US (*n* = 104)	DE (*n* = 120)	Model average
FR	0.85	0.76	0.82	0.64	0.83	0.76
AT	0.62	0.92	0.74	0.59	0.65	0.65
CN	0.82	0.76	0.81	0.67	0.83	0.77
US	0.76	0.78	0.70	0.74	0.79	0.76
DE	0.84	0.79	0.88	0.74	0.79	0.81

Classification models based on eggNOG

Test set

Training set	FR (*n* = 114)	AT (*n* = 109)	CN (*n* = 128)	US (*n* = 104)	DE (*n* = 120)	Model average
FR	0.86	0.79	0.85	0.66	0.82	0.78
AT	0.71	0.80	0.79	0.67	0.72	0.72
CN	0.84	0.79	0.87	0.74	0.81	0.80
US	0.78	0.83	0.80	0.75	0.84	0.81
DE	0.77	0.80	0.78	0.72	0.93	0.77



nature medicine

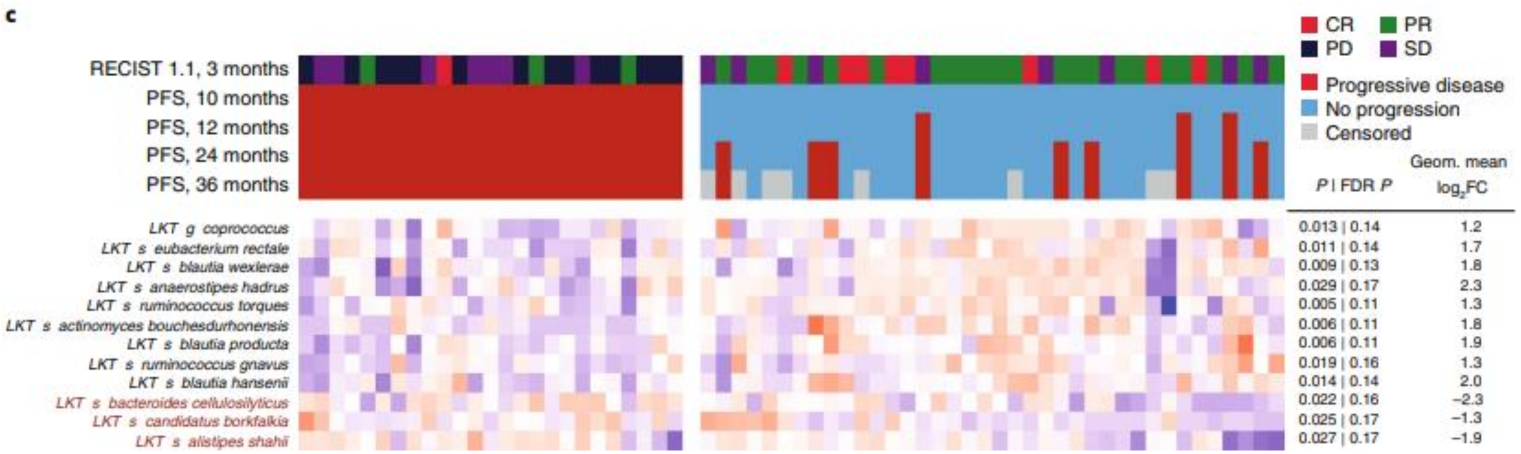
ARTICLES

https://doi.org/10.1038/s41591-022-01698-2

Check for updates

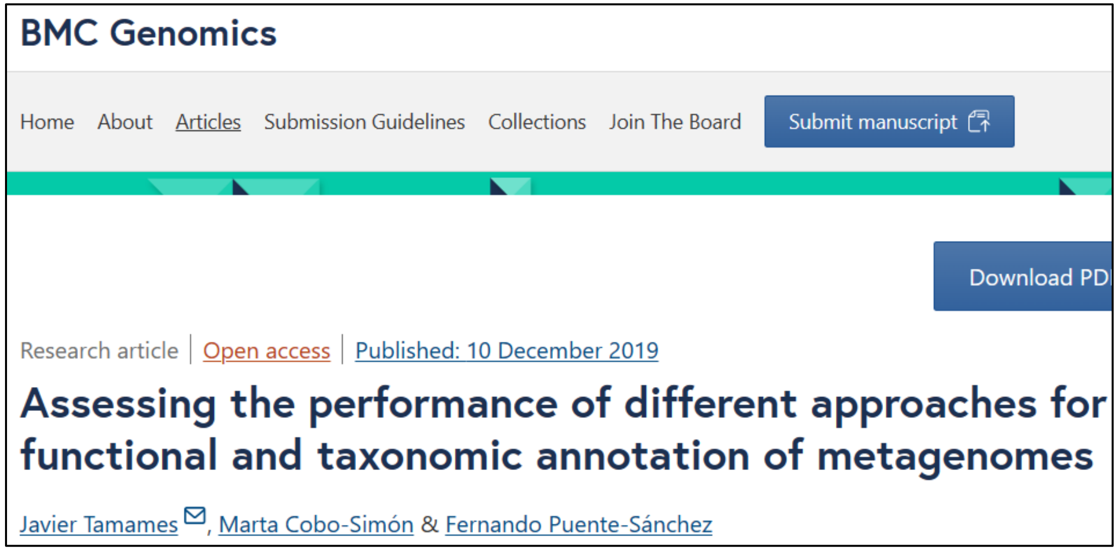
Intestinal microbiota signatures of clinical response and immune-related adverse events in melanoma patients treated with anti-PD-1

Kraken2, a reference genome-based k-mer profiling tool, was to compare genomic abundances between different conditions



Reads mapping rate

- Since marker genes only accounted for a small portion of the entire metagenome, only a small portion of reads are mapped to references in marker gene-based methods



BMC Genomics

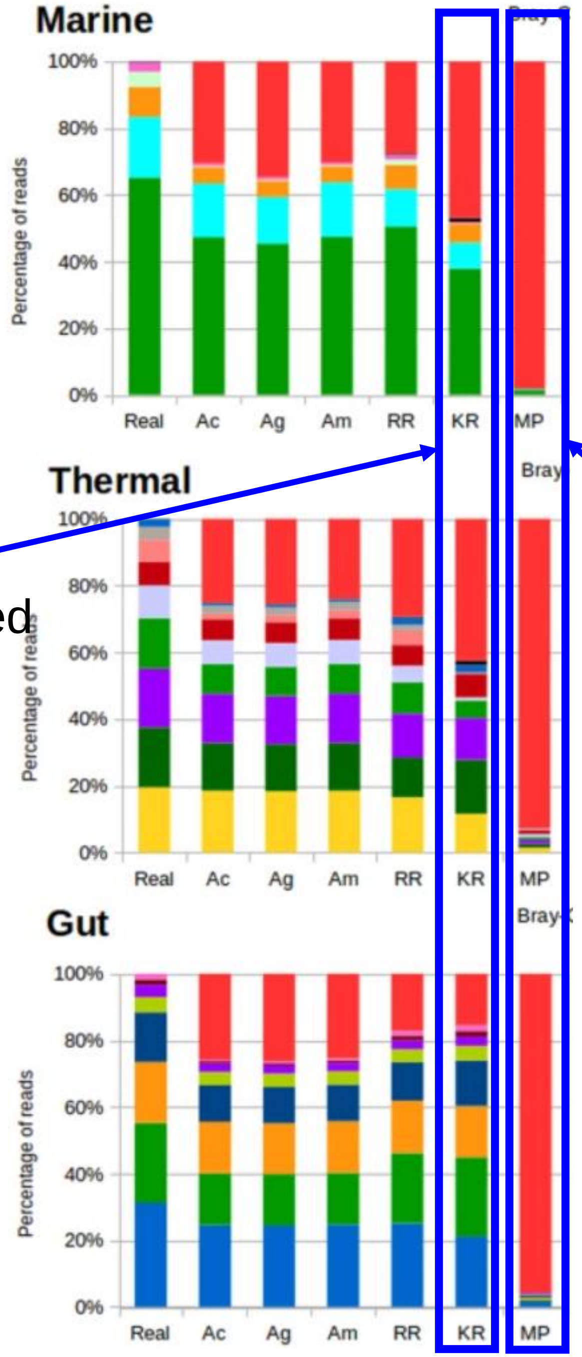
Home About Articles Submission Guidelines Collections Join The Board Submit manuscript

Download PD

Research article | Open access | Published: 10 December 2019

Assessing the performance of different approaches for functional and taxonomic annotation of metagenomes

Javier Tamames, Marta Cobo-Simón & Fernando Puente-Sánchez



Reference genome-based method (Kraken2)

Marker gene-based method (Metaphlan2)

Which tool to use?

- Depends on the purpose
- If your purpose is to get a glimpse on the distribution of microbial species, then all tools should be ok regardless of small differences
  - Hard to say which one works better beforehand
- However, if your purpose if to assign reads to different taxonomic ranks, then you should consider whole-genome-database-based approaches (such as Kraken2) instead of marker gene-based tools (e.g. Metaphlan2)

Genome reconstruction

- Metagenome assembly -> binning -> quality checking
- All using readily-available tools

Metagenome assembly

- Reads quality control (trimming)
  - Trimmomatic
  - bbduk (BBTools published by JGI)
- Assembly
  - metaSPAdes
  - MEGAHIT

And yes I recommend that trimming tools should be run before submitting metagenomes into profiling tools

Trimmomatic

- A java program
- Just download the jar file and the provide TruSeq3 adaptor file (also come along with the package).
- Usage (similar to what was described on its website:

```
java -jar trimmomatic-0.39.jar PE input_forward.fq.gz
input_reverse.fq.gz output_forward_paired.fq.gz
output_forward_unpaired.fq.gz
output_reverse_paired.fq.gz
output_reverse_unpaired.fq.gz ILLUMINACLIP:TruSeq3-
PE.fa:2:30:10:2:True LEADING:10 TRAILING:10 MINLEN:36
```

Specify the minimum quality to be kept at the head and tail

I don’t have Java?

- Try download and install java (on Ubuntu)

With sudo

Command:

```
sudo apt-get install default-jre
```

Without sudo

1. Download Java Runtime Environment (jre) tar.gz file from java.com
2. Untar (tar –zxvf) the file
3. Set the path into system PATH to run it everywhere on the machine

Running Trimmomatic

- Since I do not like to type in everything every time I run Trimmomatic, I just composed a bash script

run_trim_pair.sh

```
cp /home/yuwei/bin/Trimmomatic-0.39/TruSeq3-PE.fa .

java -jar /home/yuwei/bin/Trimmomatic-0.39/trimmomatic-0.39.jar PE -
phred33 $1 $2 $1.trimmed $1.filtered $2.trimmed $2.filtered
ILLUMINACLIP:TruSeq3-PE.fa:2:30:10:2:True LEADING:20
TRAILING:20 MINLEN:36

rm TruSeq3-PE.fa
```

And I run it via the following command

```
$ ~/bin/Trimmomatic-0.39/run_trim_pair.sh (pair1.fq.gz) (pair2.fq.gz)
```

bbduk

- Similarly, bbduk can also be used for trimming purpose

- My script to run bbduk is as follows (in the same logic as I setup Trimmomatic)

```
/home/yuwwu/bin/bbmap/bbduk.sh in1=$1 in2=$2
ref=/home/yuwwu/bin/bbmap/resources/adapters.fa
out1=$1.bbduk_trimmed.fq out2=$2.bbduk_trimmed.fq
stats=$1.stats.txt k=23 ktrim=r mink=11 hdist=1 tpe tbo
qtrim=rl trimq=20 maq=20
```

Assembly

- Tools include metaSPAdes or MEGAHIT can be used

- metaSPAdes
  - SPAdes with “-meta” mode. (note this is contradictory to “-careful” mode)
  - Can assemble longer scaffolds but may encounter memory insufficiency problem

- MEGAHIT
  - Designed specifically for memory saving purpose
  - Assembled scaffolds are shorter (somewhat) than metaSPAdes but (usually) do not have memory problem even facing large dataset

Assembly quality?

- Usually very difficult to evaluate, as the contigs/scaffolds are very short compared to single genome projects

- What you can do is to compare **between** different assemblies, say which one yields better (longer) scaffolds, etc.

- However, in complicated metagenomes, MEGAHIT is usually the only option for assembly

Assembly statistics

- We can measure the quality using N50 as metric
  - a very commonly used metric for assessing genome assembly statistics
  - N50 is defined as “the length of contig from which 50% of the bases are in it and shorter contigs”

- Imagine we got 7 contigs with lengths as
  - 1, 1, 3, 5, 8, 12, 20 ➔ sort it in descending order as 20, 12, 8, 5, 3, 1, 1
- The total length is
  - 20+12+8+5+3+1+1 = 50
- N50 is “halfway” to the summed total length
  - 20 (not yet halfway)
  - 20+12 = 32 (halfway reached)

N50 = 12

And L50 is the number of contigs within N50.

In this case the **L50** is 3

There are also N90 and L90

- N90: “the length of contig from which 90% of the bases are in it and shorter contigs”
- Back to our example
  - 1, 1, 3, 5, 8, 12, 20 ➔ sort it in descending order as 20, 12, 8, 5, 3, 1, 1
- The total length is
  - 20+12+8+5+3+1+1 = 50
- N90 is 90% to the summed total length (50*0.9=45)
  - 20 (not yet 90%)
  - 20+12 = 32 (not yet 90%)
  - 20+12+8 = 40 (not yet 90%)
  - 20+12+8+5 = 45 (90% reached)

N90 = 5

And L90 is the number of contigs within N90.

In this case the **L90** is 4

Binning

- Dozens of software to choose from
  - MaxBin
  - MetaBAT
  - CONCOCT
  - …
- And there are also tools that “merge” the results together
  - DASTools
  - MetaWrap

- Just follow the instruction unless you feel like some settings are better for you

Short contigs?

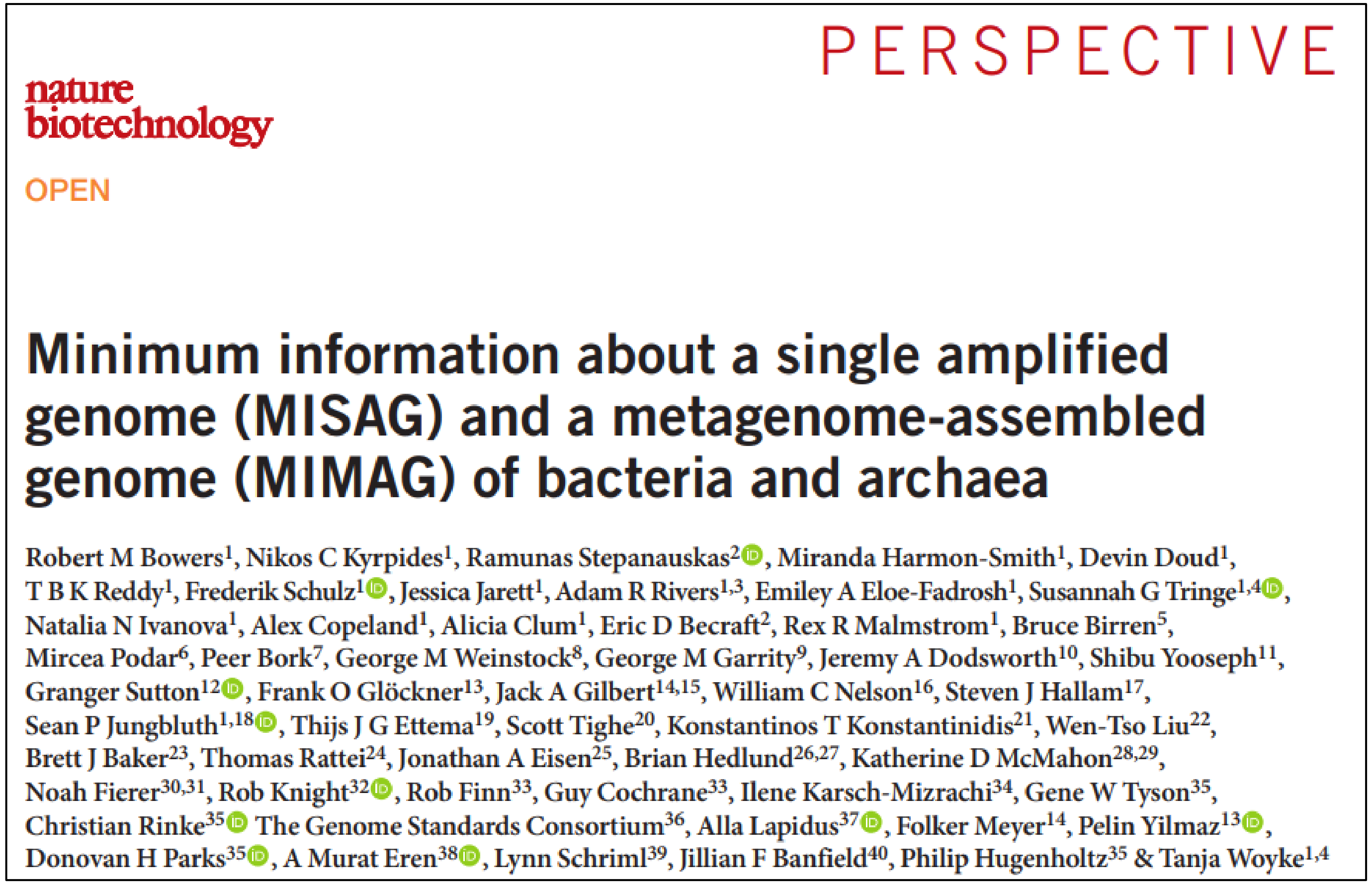
- Due to the difficulty of retrieving genomic signals from short contigs (usually indicates contigs < 1000 or 2000 bps), binning tools usually have limitation on the contig lengths
  - In other words, shorter contigs will NOT be binned
  - As of today no one has a solution yet. The only hope is to assemble better metagenomes.

- Third generation sequencing such as PacBio or Nanopore may resolve this problem

Genome quality checking

- CheckM may be the only option right now.
- Very recently CheckM2 was released. But there are two repositories exist at the same time
  - Have not tried checkM2 but should be very similar on most genomes, as CheckM2 aims to resolve genomes with much reduced sizes (e.g. DPANN, which has a smaller genome and many members are episymbionts)

How good is “good”?



nature biotechnology

PERSPECTIVE

OPEN

Minimum information about a single amplified genome (MISAG) and a metagenome-assembled genome (MIMAG) of bacteria and archaea

Robert M Bowers[1], Nikos C Kyrpides[1], Ramunas Stepanauskas[2], Miranda Harmon-Smith[1], Devin Doud[1], T B K Reddy[1], Frederik Schulz[1], Jessica Jarett[1], Adam R Rivers[1,3], Emiley A Eloe-Fadrosh[1], Susannah G Tringe[1,4], Natalia N Ivanova[1], Alex Copeland[1], Alicia Clum[1], Eric D Becraft[2], Rex R Malmstrom[1], Bruce Birren[5], Mircea Podar[6], Peer Bork[7], George M Weinstock[8], George M Garrity[9], Jeremy A Dodsworth[10], Shibu Yooseph[11], Granger Sutton[12], Frank O Glöckner[13], Jack A Gilbert[14,15], William C Nelson[16], Steven J Hallam[17], Sean P Jungbluth[1,18], Thijs J G Ettema[19], Scott Tighe[20], Konstantinos T Konstantinidis[21], Wen-Tso Liu[22], Brett J Baker[23], Thomas Rattei[24], Jonathan A Eisen[25], Brian Hedlund[26,27], Katherine D McMahon[28,29], Noah Fierer[30,31], Rob Knight[32], Rob Finn[33], Guy Cochrane[33], Ilene Karsch-Mizrachi[34], Gene W Tyson[35], Christian Rinke[35] The Genome Standards Consortium[36], Alla Lapidus[37], Folker Meyer[14], Pelin Yilmaz[13], Donovan H Parks[35], A Murat Eren[38], Lynn Schriml[39], Jillian F Banfield[40], Philip Hugenholtz[35] & Tanja Woyke[1,4]

Quality	Description
Finished	Single contiguous sequence
High-quality draft	Completeness > 90% Contamination < 5% Has 23S, 16S, and 5S rRNA Has at least 18 tRNA
Medium-quality draft	Completeness > 50% Contamination < 10%
Low-quality draft	Completeness < 50% Contamination < 10%

Summary

- I hope this workshop provided the basic information for people who want to dive into the metagenomic analysis world
- Very often you may encounter problems in this process, for example some programs cannot be compiled very smoothly or you do not know how to run a program. Remember: Google is your friend.