RNA Sequencing

Building your own pipeline from scratch

#1

SUBSCRIBE

Overview

* Setup a Linux Environment
  * Setup and install
* Setup
  * Indexing a Reference Genome and Transcriptome
  * Sequence Read Archive (SRA)
* RNA-Seq pipeline

* Code and Scripts used today are available as gist:

https://gist.github.com/DannyArends/04d87f5590090dfe0dc6b42e5e1bbe15

Linux environment

* VirtualBox

  https://www.virtualbox.org/wiki/Downloads

* Debian

  https://www.debian.org/CD/netinst/
  https://cdimage.debian.org/debian-cd/current/amd64/iso-dvd/

Debian Quirk

* By default, the standard user doesn't have sudo rights

```
danny@debian:~$ su -
root@debian:~$ usermod -aG sudo danny
root@debian:~$ exit
```

* Change danny to your own username
* After, logout, then relogin, and sudo will work

Guest additions

* Extension pack for VirtualBox

VirtualBox 6.1.38 Oracle VM VirtualBox Extension Pack

- All supported platforms

* Setup the Guest Additions inside the Guest OS

```
danny@debian:~$ cd /media/cdrom0
danny@debian:/media/cdrom0$ sudo sh ./VBoxLinuxAdditions.run
```

The pipeline

* RNA-Seq pipeline
  * Read trimming & Adapter removal (trimmomatric)
  * Alignment (STAR)
  * Remove duplicates (picard)
  * INDEL realignment (gatk)
  * Base Recalibration (gatk)
  * Extract Read Counts (bedtools)
  * Compute RPKM (GenomicFeatures)
  * Normalization (preprocessCore)

Tools for inside the box

* R
  * GenomicFeatures
  * preprocessCore
* Trimmomatic
* STAR
* Picard tools
* HTSlib, BCFtools, samtools
* GATK
* bedtools

Download and install

* R - installed via apt (the Debian package manager)

```
danny@debian:~$ sudo apt install r-base
```

* If complains about the CD

```
danny@debian:~$ sudo nano /etc/apt/sources.list
```

* SSL, XML2, CURL are needed to install the GenomicFeatures & preprocessCore packages

```
danny@debian:~$ sudo apt install libssl-dev
danny@debian:~$ sudo apt install libxml2-dev
danny@debian:~$ sudo apt install libcurl4-openssl-dev
```

Download and install

* GenomicFeatures & preprocessCore

```
danny@debian:~$ sudo R

> if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

> BiocManager::install("GenomicFeatures")
> BiocManager::install("preprocessCore")
> q("no")
```

Trimmomatic

* To install Trimmomatic, we need git and ant

```
danny@debian:~$ sudo apt install git
danny@debian:~$ sudo apt install ant
```

* Make a folder to hold all the software tools

```
danny@debian:~$ mkdir software
danny@debian:~$ cd software
```

* Get a local copy of Trimmomatic

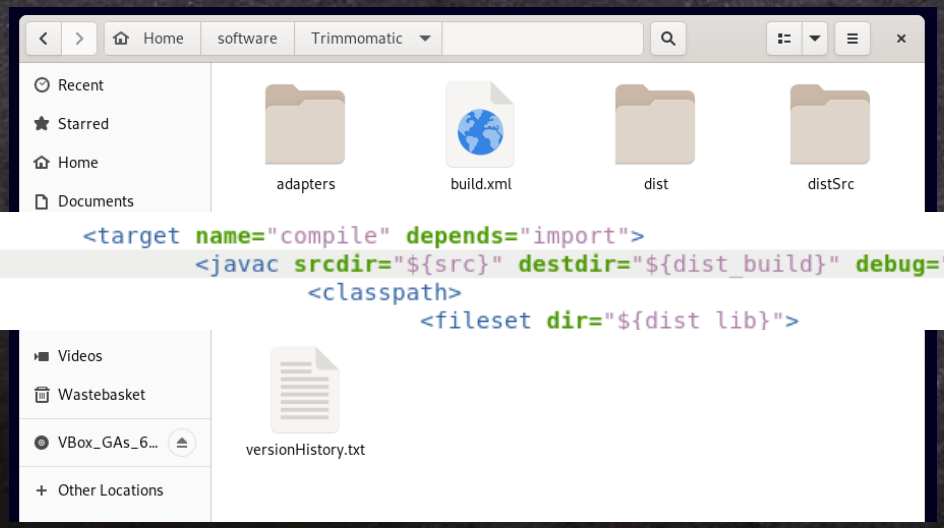
```
danny@debian:~/software$
  git clone https://github.com/usadellab/Trimmomatic.git
```

Trimmomatic

- We need to update the version of java
  - Debian has a newer version
  - Open build.xml
  - Go to line 34, and change

source="1.5" to source="1.6"

target="1.5" to target="1.6"




```
33      <target name="compile" depends="import">
34              <javac srcdir="${src}" destdir="${dist_build}" debug="true" source="1.6" target="1.6" includeantruntime="false">
35                      <classpath>
36                              <fileset dir="${dist lib}">
```

Trimmomatic

* Now we can compile it

```
danny@debian:~/software$ cd Trimmomatic
danny@debian:~/software/Trimmomatic$ ant
```

```
dist:
    [unjar] Expanding: /home/danny/software/Trimmomatic/dist/lib/jbzip2-0.9.1.jar into /home/dann
   [delete] Deleting directory /home/danny/software/Trimmomatic/dist/unpack/META-INF
   [delete] Deleting directory /home/danny/software/Trimmomatic/dist/unpack/demo
     [move] Moving 1 file to /home/danny/software/Trimmomatic/dist/unpack
     [move] Moving 1 file to /home/danny/software/Trimmomatic/dist/unpack
     [copy] Copying 1 file to /home/danny/software/Trimmomatic/dist/unpack
      [jar] Building jar: /home/danny/software/Trimmomatic/dist/jar/trimmomatic-0.40-rc1.jar
      [zip] Building zip: /home/danny/software/Trimmomatic/dist/Trimmomatic-0.40-rc1.zip

BUILD SUCCESSFUL
Total time: 0 seconds
```

STAR

* Spliced Transcripts Alignment to a Reference (STAR)

```
danny@debian:~/software$
   git clone https://github.com/alexdobin/STAR.git
danny@debian:~/software$ cd STAR/source
danny@debian:~/software/STAR/source$ make
```

* It ends with a warning, test it:

```
danny@debian:~/software/STAR/source$ ./STAR
```

```
danny@debian:~/software/STAR/source$ ./STAR
Usage: STAR  [options]... --genomeDir /path/to/genome/index/   --readFilesIn R1.fq R2.fq
Spliced Transcripts Alignment to a Reference (c) Alexander Dobin, 2009-2022

STAR version=2.7.10a_alpha_220818
STAR compilation time,server,dir=2022-10-08T13:11:03+01:00 :/home/danny/software/STAR/source
```

PICARD tools

* PICARD tools

```
danny@debian:~/software$
   git clone https://github.com/broadinstitute/picard.git
danny@debian:~/software$ cd picard
danny@debian:~/software/picard$ ./gradlew shadowJar
```

```
Note: Some input files use or override a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
7 warnings

BUILD SUCCESSFUL in 1m 13s
4 actionable tasks: 4 executed
```

htslib

* Install autotools

```
danny@debian:~/software$ sudo apt install autoconf
```

* Get local versions

```
danny@debian:~/software$
  git clone https://github.com/samtools/htslib.git
danny@debian:~/software$
  git clone https://github.com/samtools/samtools.git
danny@debian:~/software$
  git clone https://github.com/samtools/bcftools.git
```

htslib

* Download the requirements

```
danny@debian:~/software$ cd htslib
danny@debian:~/software/htslib$
   git submodule update --init --recursive
```

* Reconfigure and compile htslib

```
danny@debian:~/software/htslib$ autoreconf -i
danny@debian:~/software/htslib$ ./configure
danny@debian:~/software/htslib$ make
```

samtools

* Configure and compile samtools

```
danny@debian:~/software$ cd samtools
danny@debian:~/software/samtools$ autoheader
danny@debian:~/software/samtools$ autoconf -Wno-syntax
danny@debian:~/software/samtools$ ./configure
danny@debian:~/software/samtools$ make
```

bcftools

* Compile samtools

```
danny@debian:~/software$ cd bcftools
danny@debian:~/software/bcftools$ autoheader
danny@debian:~/software/bcftools$ autoconf -Wno-syntax
danny@debian:~/software/bcftools$ ./configure
danny@debian:~/software/bcftools$ make
```

GATK

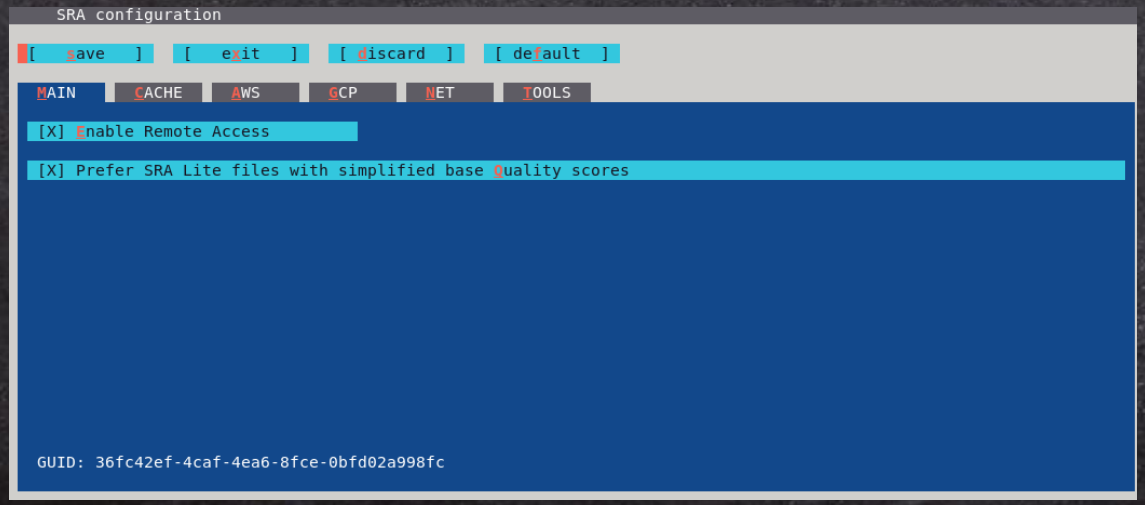
* Get local version of GATK

```
danny@debian:~/software$
  wget https://github.com/broadinstitute/gatk/releases/download/4.2.6.1/gatk-4.2.6.1.zip
danny@debian:~/software$ unzip gatk-4.2.6.1.zip
```

SRA toolkit

* Get local version of SRA

```
danny@debian:~/software$
   wget https://ftp-trace.ncbi.nlm.nih.gov/sra/sdk/3.0.0/sratoolkit.3.0.0-centos_linux64-cloud.tar.gz
danny@debian:~/software$ mkdir sratoolkit
danny@debian:~/software$
   tar -xzf sratoolkit.3.0.0-centos_linux64-cloud.tar.gz -C sratoolkit
danny@debian:~/software$
./sratoolkit/usr/local/ncbi/sra-tools/bin/vdb-config -interactive
```



So almost there

* Make a local bin folder

```
danny@debian:~$ mkdir bin
danny@debian:~$ cd bin
```

* Symlink the required tools

```
danny@debian:~/bin$ ln -s /home/danny/software/STAR/source/STAR STAR
danny@debian:~/bin$ ln -s /home/danny/software/htslib/bgzip bgzip
danny@debian:~/bin$ ln -s /home/danny/software/samtools/samtools samtools
danny@debian:~/bin$ ln -s /home/danny/software/bcftools/bcftools bcftools
danny@debian:~/bin$
ln -s /home/danny/software/sratoolkit/usr/local/ncbi/sra-tools/bin/fasterq-dump fasterq-dump
```

Update your path

* Update your bash file

```
danny@debian:~$ nano ~/.bashrc
```

* Add the following at the end:

```
export PATH="$HOME/bin:$PATH"
```

* Press ctrl+o followed by ctrl+x to save the file
* Exit the terminal, and reopen

```
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

export PATH="$HOME/bin:$PATH"
```

The new terminal

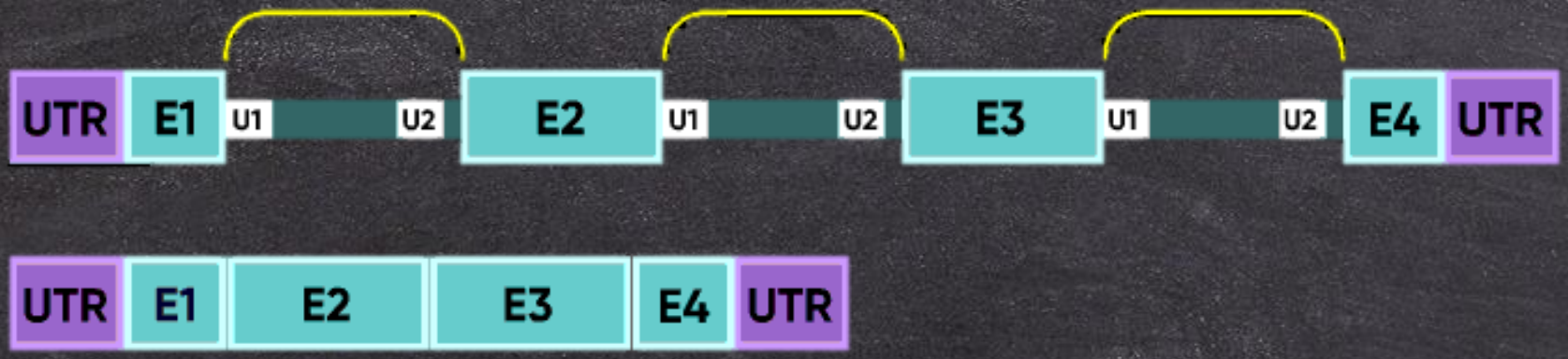
* Can directly execute
  * samtools — Manipulate SAM/BAM files
  * bgzip — Blocked GZ compression
  * STAR — Alignment of reads
  * bcftools — Manipulate VCF files
  * fasterq-dump — Get reads from SRA
* We also have
  * trimmomatic.jar — Read & Adapter trimming
  * gatk-package-4.2.6.1-local.jar — Everything GATK
  * picard.jar — Various SAM/BAM tools

Reference Genome

* Needed to align reads against
* Needs to be indexed for fast alignment
* Comes in different flavors
  * primary_assembly versus toplevel
  * DNA, SM, HM masking

Transcriptome

* Needed for intron/exon boundary



Setting up a genome

- *Saccharomyces cerevisiae*
  - 12 Mb genome
  - 16 chromosomes
- First eukaryotic sequenced
  - 1996
- Reference: S288C

Mogana Das Murtey and Patchamuthu Ramasamy, CC BY-SA 3.0, via *Wikimedia Commons*

Ensembl

- *Saccharomyces cerevisiae*
  - Only has toplevel available

Saccharomyces_cerevisiae.R64-1-1.dna.chromosome.XII.fa.gz	2022-05-12 12:06	324K
Saccharomyces_cerevisiae.R64-1-1.dna.chromosome.XIII.fa.gz	2022-05-12 12:06	281K
Saccharomyces_cerevisiae.R64-1-1.dna.chromosome.XIV.fa.gz	2022-05-12 12:06	239K
Saccharomyces_cerevisiae.R64-1-1.dna.chromosome.XV.fa.gz	2022-05-12 12:06	333K
Saccharomyces_cerevisiae.R64-1-1.dna.chromosome.XVI.fa.gz	2022-05-12 12:06	289K
Saccharomyces_cerevisiae.R64-1-1.dna.toplevel.fa.gz	2022-05-12 12:06	3.6M
Saccharomyces_cerevisiae.R64-1-1.dna_rm.chromosome.I.fa.gz	2022-05-12 12:06	67K
Saccharomyces_cerevisiae.R64-1-1.dna_rm.chromosome.II.fa.gz	2022-05-12 12:06	240K

- Create our own primary_assembly using R

Create our own primary_assembly

* Download the individual chromosomes
* Unpack them into 1 big chromosome
* Re-pack the chromosome using bgzip

* Let's start by making a folder for the reference data

```
danny@debian:~$ mkdir genome
danny@debian:~$ cd genome
```

* Start R

```
danny@debian:~$ R
```

Create a Primary Assembly

- Download
- Extract & Merge
- Compress
- Delete Chrs

```r
#
# Download Saccharomyces Cerevisiae genome
# copyright (c) 2022 - Danny Arends
#

uri <- "ftp.ensembl.org/pub/release-107/fasta/saccharomyces_cerevisiae/dna/"
base <- "Saccharomyces_cerevisiae.R64-1-1.dna.chromosome."
chrs <- c(as.character(as.roman(seq(1:16))), "Mito")

# Download
for(chr in chrs){
  fname <- paste0(base, chr, ".fa.gz")

  # Download command
  cmd <- paste0("wget ", uri, fname)
  #cat(cmd, "\n")
  system(cmd)
}

# Create an empty the file
cat("", file = "Saccharomyces_cerevisiae.R64-1-1.dna.primary_assembly.fa")
for(chr in chrs){
  fname <- paste0(base, chr, ".fa.gz")
  # Extract and merge into a fast file
  cmd <- paste0("zcat ", fname, " >> Saccharomyces_cerevisiae.R64-1-1.dna.primary_assembly.fa")
  #cat(cmd, "\n")
  system(cmd)
}

# Compress the fasta file using bgzip (keep original)
cmd <- paste0("bgzip -k Saccharomyces_cerevisiae.R64-1-1.dna.primary_assembly.fa")
#cat(cmd, "\n")
system(cmd)

# Delete the chromosomes
for(chr in chrs){
  fname <- paste0(base, chr, ".fa.gz")
  # Extract and merge into a fast file
  cmd <- paste0("rm ", fname)
  #cat(cmd, "\n")
  system(cmd)
}
```

Download the transcriptome

* Go to ensemble and get the GTF url
* Download the transcriptome

```
danny@debian:~/genome$ wget <URL>
danny@debian:~/genome$ gunzip Saccharomyces_cerevisiae.R64-1-1.107.gtf.gz
```

Index of /pub/release-107/gtf/saccharomyces_cerevisiae

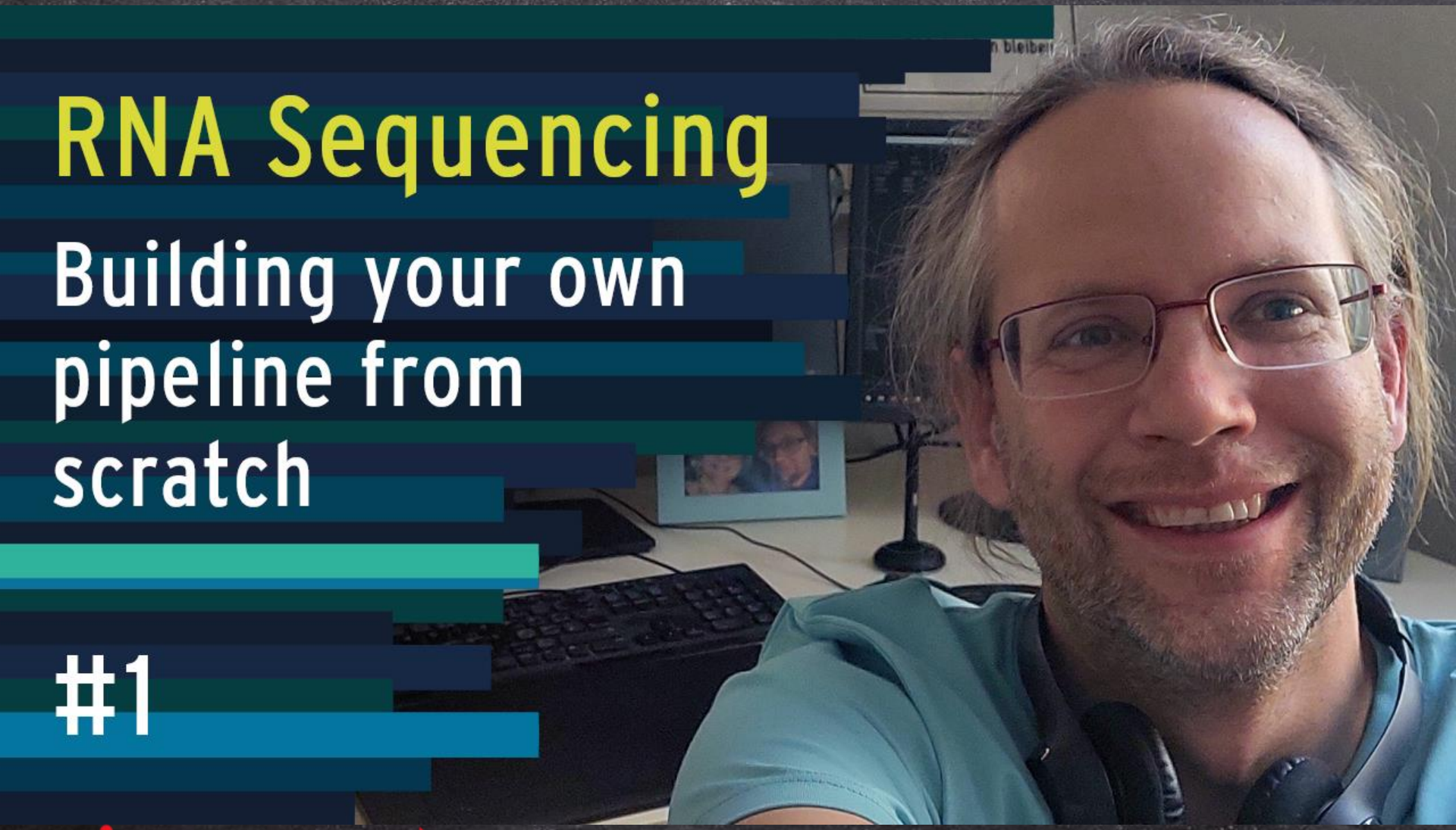
Name	Last modified	Size	Description
Parent Directory		-	
CHECKSUMS	2022-05-23 06:33	140	
README	2022-05-14 16:10	9.2K	
Saccharomyces_cerevisiae.R64-1-1.107.abinitio.gtf.gz	2022-05-14 16:10	116	
Saccharomyces_cerevisiae.R64-1-1.107.gtf.gz	2022-05-14 16:10	572K	

Now we are all set

- Next time we'll go through the next steps:
  - The first RNA alignment (in detail)
  - Extracting RPKM values
  - Testing differential expression
  - Building a flexible pipeline with R scripts
  - Adding automated QC to the pipeline

Image by Gerd Altmann from Pixabay

Thanks for watching



SUBSCRIBE