RNA Sequencing

Building your own pipeline from scratch

#2

SUBSCRIBE

Overview

* Setting up the genome
* Downloading reads from SRA
* Create an alignment pipeline
  * Trimming
  * Alignment
  * QC
* Making basic pipeline in an R script

Software I forgot

* TABIX (part of htslib)

```
danny@debian:~$ cd bin
danny@debian:~/bin$ ln -s /home/danny/software/htslib/tabix tabix
```

* Some additional R libraries

```
danny@debian:~$ sudo R
> install.packages("ggplot2")
> install.packages("gplots")
> install.packages("gsalib")
> q("no")
```

* The IGV

```
danny@debian:~$ cd software
danny@debian:~/software$ wget https://.../2.14/IGV_Linux_2.14.1_WithJava.zip
danny@debian :~/software$ unzip IGV_Linux_2.14.1_WithJava.zip
```

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

Setting up a genome

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

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:~/genome$ R
```

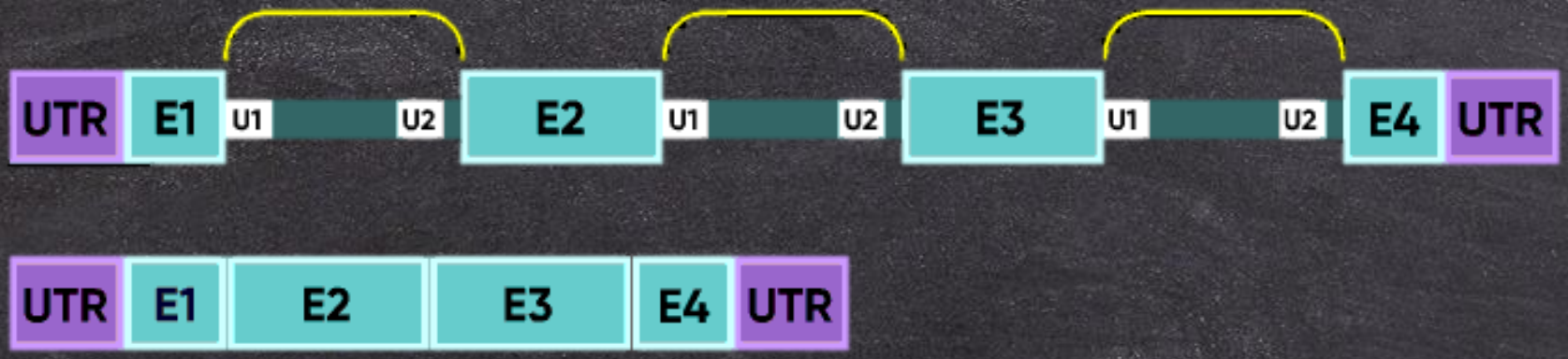
Create a Primary Assembly

- 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)
}
```

Transcriptome

* Needed for intron/exon boundary



Download the transcriptome

* ENSEMBL FTP
  * Get the GTF uri
  * 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	

Download the SNPs

* ENSEMBL FTP
  * Get the VCF uri
  * Download the known SNPs

```
danny@debian:~/genome$ wget <URL>
```

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	

Prepare Genome

- Preparing the genome requires building indices
  - Samtools, STAR, and picard need their own
- Tabix to index the SNPs

```
# Index the genome using samtools
samtools faidx Saccharomyces_cerevisiae.R64-1-1.dna.primary_assembly.fa.gz

# Generate genome/transcriptome index using STAR
STAR --runThreadN 2 --runMode genomeGenerate \
    --genomeDir ~/genome/STAR \
    --genomeSAindexNbases 10 \
    --sjdbGTFfile ~/genome/Saccharomyces_cerevisiae.R64-1-1.107.gtf \
    --genomeFastaFiles ~/genome/Saccharomyces_cerevisiae.R64-1-1.dna.primary_assembly.fa

# Get the reference SNPs and index using tabix
tabix saccharomyces_cerevisiae.vcf.gz

# Index the genome using picard
java -Xmx4g -jar /home/danny/software/picard/build/libs/picard.jar CreateSequenceDictionary \
    -R Saccharomyces_cerevisiae.R64-1-1.dna.primary_assembly.fa.gz
```

Ready to automate

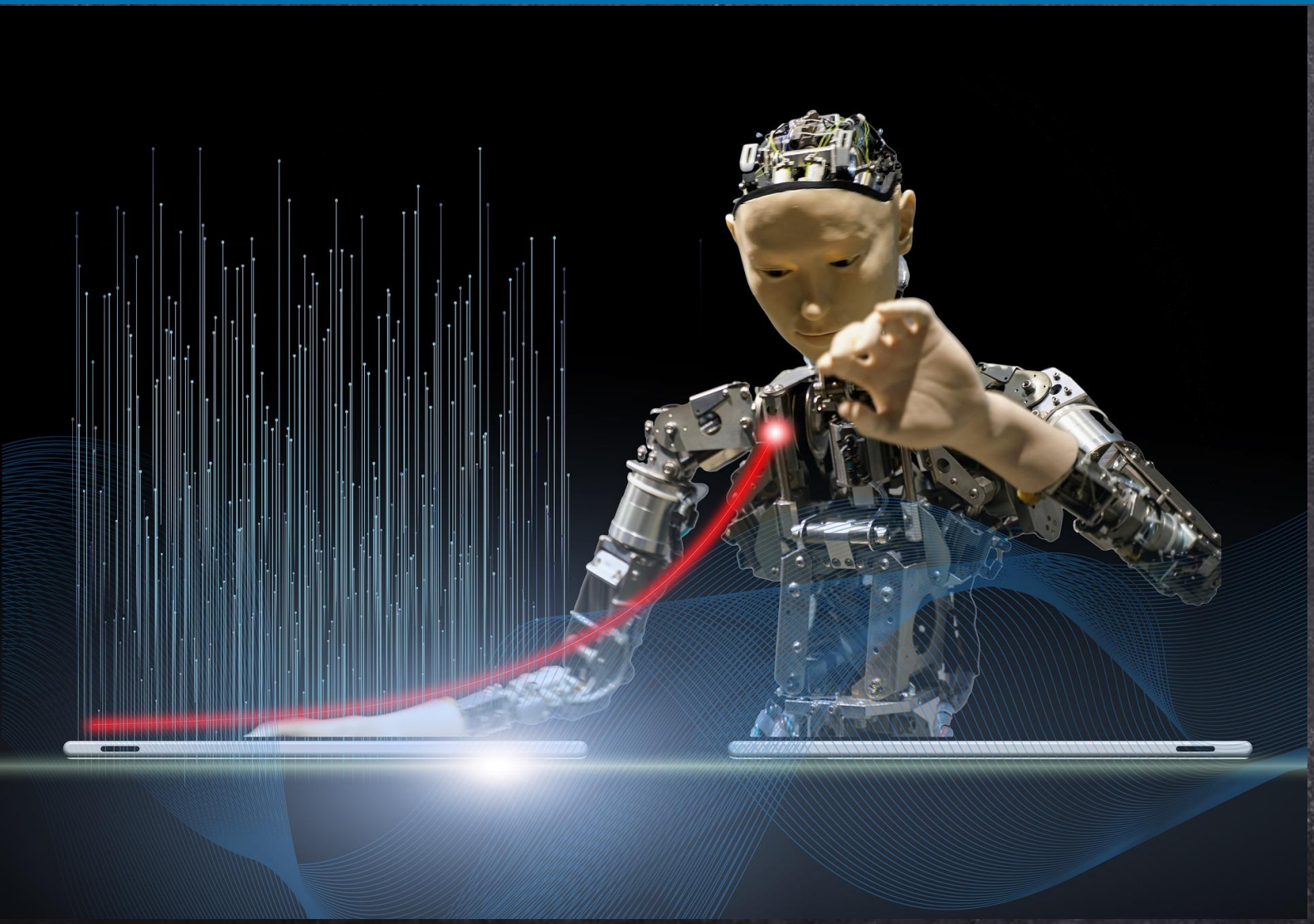


Image by Gerd Altmann from Pixabay

Get some reads SRA

* https://www.ncbi.nlm.nih.gov/sra/?term=S288C
  * Tick: Public, RNA, Paired, Illumina
* Download fastq (I took a small datasets):

```
danny@debian:~/$ mkdir -p data/raw

# 3 control samples
danny@debian:~/data/raw$ fasterq-dump -p --split-files SRR13978643
danny@debian:~/data/raw$ fasterq-dump -p --split-files SRR13978644
danny@debian:~/data/raw$ fasterq-dump -p --split-files SRR13978645

# 3 samples in SPRC medium
danny@debian:~/data/raw$ fasterq-dump -p --split-files SRR13978640
danny@debian:~/data/raw$ fasterq-dump -p --split-files SRR13978641
danny@debian:~/data/raw$ fasterq-dump -p --split-files SRR13978642
```

Get some reads SRA

* https://www.ncbi.nlm.nih.gov/sra/?term=S288C
  * Tick: Public, RNA, Paired, Illumina
* Download fastq (I took a small datasets):

```
danny@debian:~/$ mkdir -p data/raw

# 3 control samples
danny@debian:~/da[illegible]aw$ fasterq-dump -p --split[illegible]es SRR13978643
danny@debian:~/da[illegible]terq-dump -p [illegible]s SRR13978644
danny@debian:~/da[illegible]es SRR13978645

# 3 samples in SPRC m[illegible]
danny@debian:~/d[illegible] dump [illegible] SRR13978640
danny@debian:~/da[illegible]asterq-dump -p --s[illegible]s SRR13978641
danny@debian:~/da[illegible]raw$ fasterq-dump -p --split-[illegible]es SRR13978642
```

Use R to execute

```
execute <- function(x, outputfile = NA, intern = FALSE, quitOnError = FALSE){
  if (!is.na(outputfile) && file.exists(outputfile)) {
    cat("Output for step exists, skipping this step\n");
    invisible("")
  }
  cat("----", x, "\n"); res <- system(x, intern = intern); cat(">>>>", res[1], "\n")
  if(res[1] >= 1){
    cat("Error external process did not finish\n\n");
    if(quitOnError) q("no")
  }
}
```

* Example using fasterq-dump:

```
# If we always want to execute the command
execute("fasterq-dump -p --split-files SRR13978643")

# If we only want to execute the command if the file does NOT exist !
execute("fasterq-dump -p --split-files SRR13978643", "SRR13978643_1.fastq")
```

Building up a script

* Static variables

```
input.dir <- "/home/danny/data/raw"
input.base <- "SRR13978643" # This one will come from the command line
output.dir <- paste0("/home/danny/data/output/", input.base, ".aln")
genome.path <- "/home/danny/genome/STAR"
ref.fa.gz <- "<path to reference>"
ref.snps <- "<path to SNPs>"
```

Create in/out folders

```
# Create a folder for the input files
if(!file.exists(input.dir)){
  dir.create(input.dir, recursive = TRUE)
}

# Create a folder for the output files
if(!file.exists(output.dir)){
  dir.create(output.dir, recursive = TRUE)
}
```

Image by Gerd Altmann from Pixabay

SRA download

* Download the FASTQ file

```
# STEP 0 - SRA Download and Compress
setwd(input.dir)

execute(paste0("fasterq-dump -p --split-files ", input.base),
        paste0(input.base, "_1.fastq"))

execute(paste0("bgzip ", input.base, "_1.fastq"),
        paste0(input.base, "_1.fastq.gz"))
execute(paste0("bgzip ", input.base, "_2.fastq"),
        paste0(input.base, "_2.fastq.gz"))
```

Read Trimming

- 2 input files
  - FASTQ_1 and FASTQ_2
- 4 output files
  - Paired End _1 and _2
  - Unpaired _1 and _2

- Adapters
  - TruSeq3-PE-2
- Trimming Options
  - Leading, Trailing, Sliding window, Minimum length

Trimmomatic

- Trimmomatic
  - Input and output files
  - Path to Trimmomatic
  - Executable call
  - Options
  - Final command
  - Execute

```r
# STEP 1 - READ Trimming
trim.files  <- c(
                paste0(input.dir, "/", input.base,"_1.fastq.gz"),
                paste0(input.dir, "/", input.base,"_2.fastq.gz"),
                paste0(output.dir, "/", input.base,"_1.P.fastq.gz"),
                paste0(output.dir, "/", input.base,"_1.U.fastq.gz"),
                paste0(output.dir, "/", input.base,"_2.P.fastq.gz"),
                paste0(output.dir, "/", input.base,"_2.U.fastq.gz")
              )
trim.path <- "/home/danny/software/Trimmomatic"
trim.exec <- paste0("java -jar ", trim.path, "/dist/jar/trimmomatic-0.40-rc1.jar")
trim.opts <- paste0("ILLUMINACLIP:",trim.path,"/adapters/TruSeq3-PE-2.fa:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36")
trim.cmd  <- paste0(trim.exec, " PE ", paste0(trim.files, collapse=" "), " ", trim.opts)
execute(trim.cmd, trim.files[3])
```

Let’s build the rest

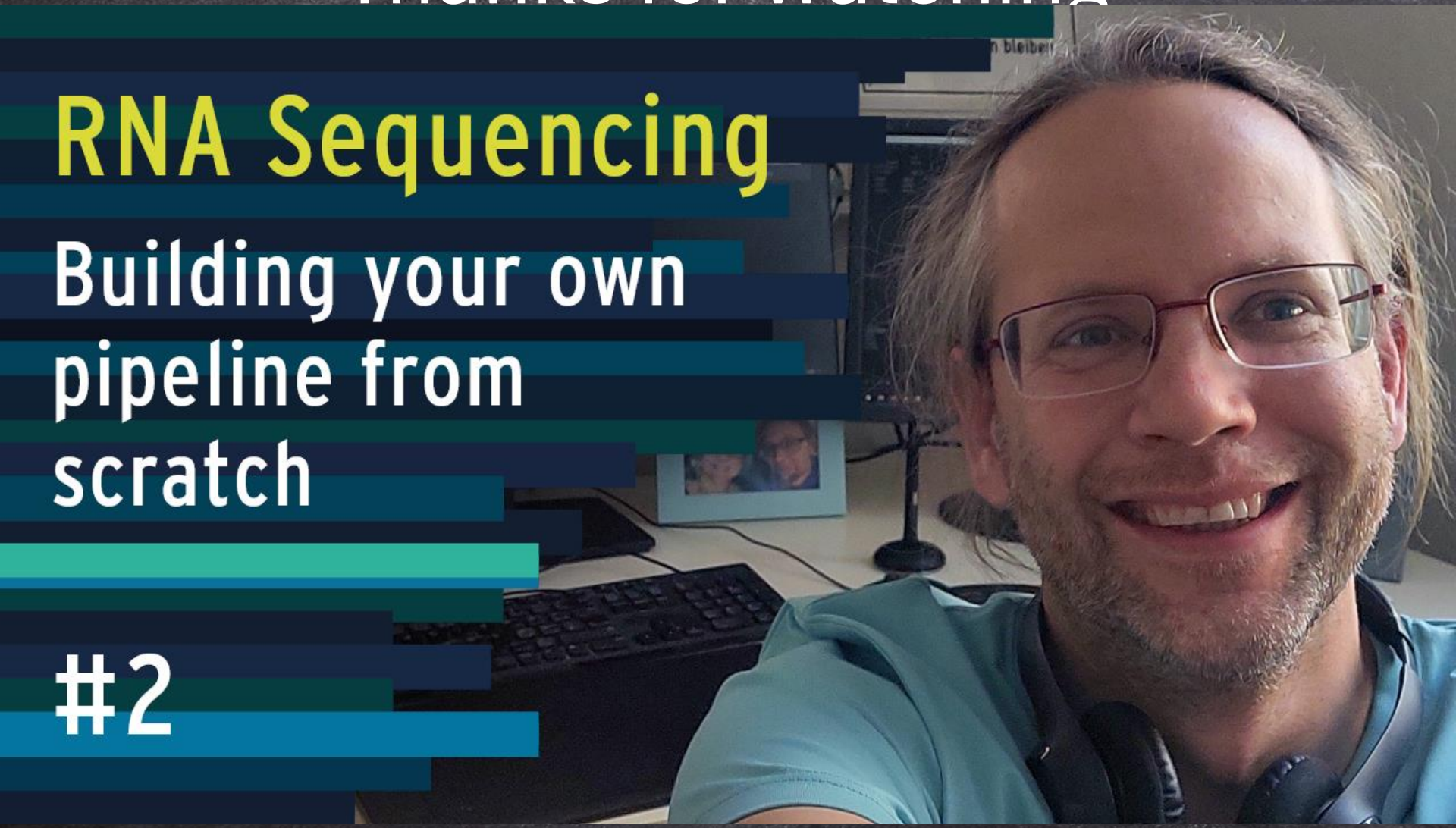
* Alignment via STAR
* Coverage statistics
* Removing duplicate reads
* Add readgroup and run, library, and name
* GATK base recalibratrion
  * Based on known SNPs

Next time

- The the next steps
  - 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