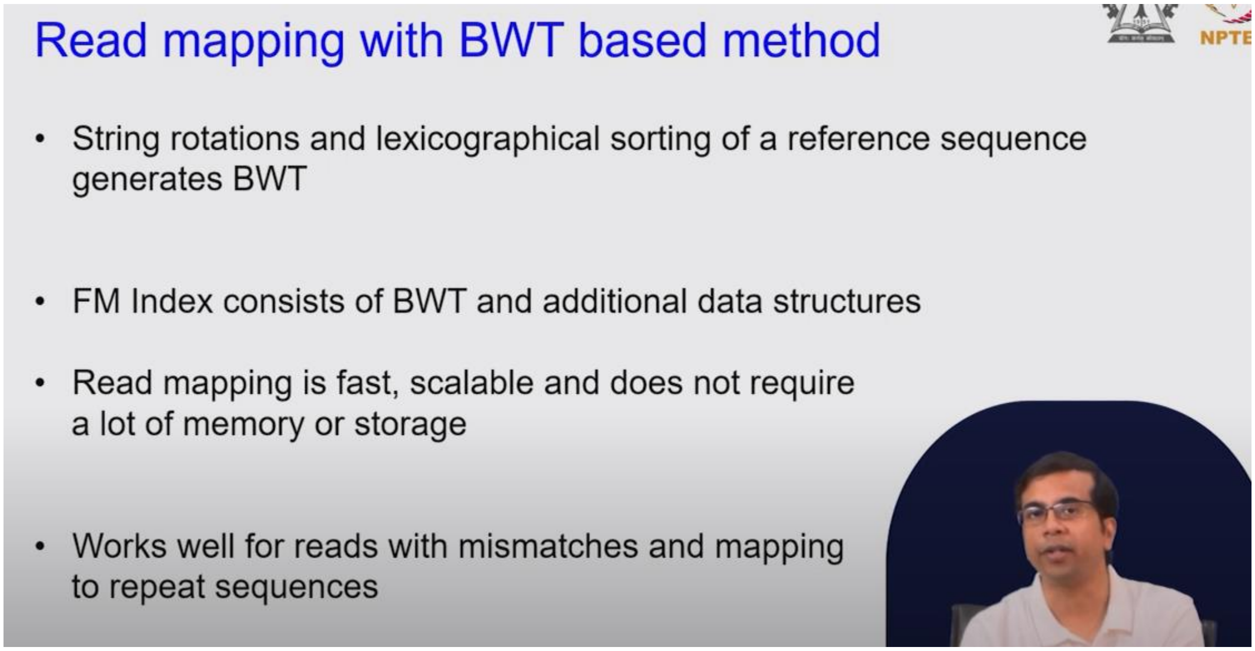
Next Generation Sequencing Technologies: Data Analysis and Applications

Bowtie2 tool

Dr. Riddhiman Dhar, **Department of Biotechnology**
Indian Institute of Technology, Kharagpur

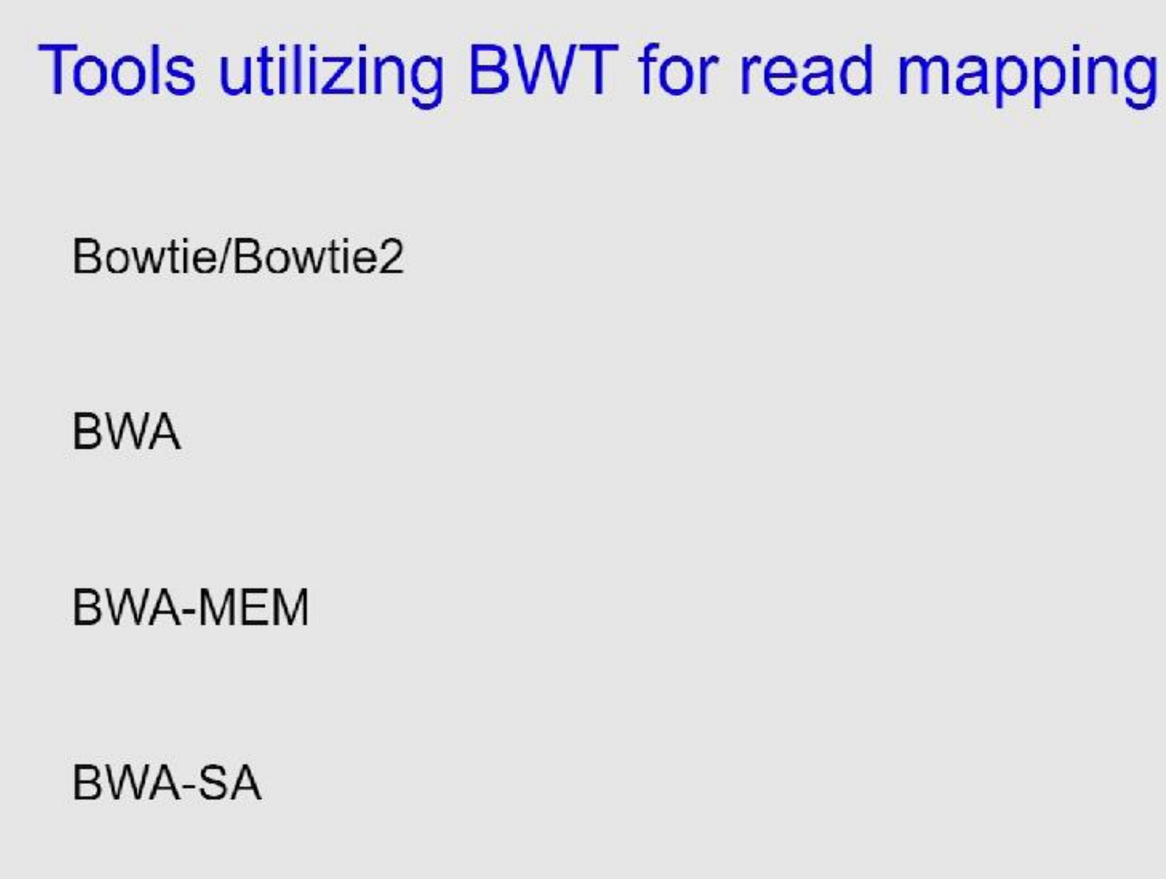
Good day, everyone. Welcome to the course on Next Generation Sequencing Technology Data Analysis and Applications. In the last few classes, we have discussed several mapping algorithms, and in the last class we looked at BWT-based mapping, and we have seen that this is the method that is probably the ideal method for us because it is fast, it does not take too much memory, and it can handle all sorts of reads, including reads with mismatches. It can also map reads to multiple positions or repeat sequences. So, what we are going to do today is use a tool right and look at a tool that utilizes this BWT, and we will do this hands-on right; we will do this read mapping in this class and the subsequent class, ok? So, let us start with that. The tool that we will discuss is called the Bowtie 2 tool, right? So, this will be the concept that we will be talking about, right? So, we will talk about the Bowtie 2 tool for read mapping. So, how does this work? So, we have talked about this, just to remind you, right? So, we have seen how we generate BWT through string rotations and hexa-geographic sorting. We have talked about FM index in the last class right, which consists of BWT and additional data structures. What we have seen with some examples is that read mapping is fast scalable and it does not require too much memory, and if you want to store the BWT you can also store it without too much space. This mapping also works very well for each with mismatches, and if we are mapping reads to repeat sequences, ok.



Read mapping with BWT based method

- String rotations and lexicographical sorting of a reference sequence generates BWT
- FM Index consists of BWT and additional data structures
- Read mapping is fast, scalable and does not require a lot of memory or storage
- Works well for reads with mismatches and mapping to repeat sequences

So, there are several tools that are available for this that utilize this BWT for mapping. So, you have Bowtie or Bowtie version 2, which is the Bowtie 2, and you have BWA and its variance, called BWA-MEM or BWA-SA.



Tools utilizing BWT for read mapping

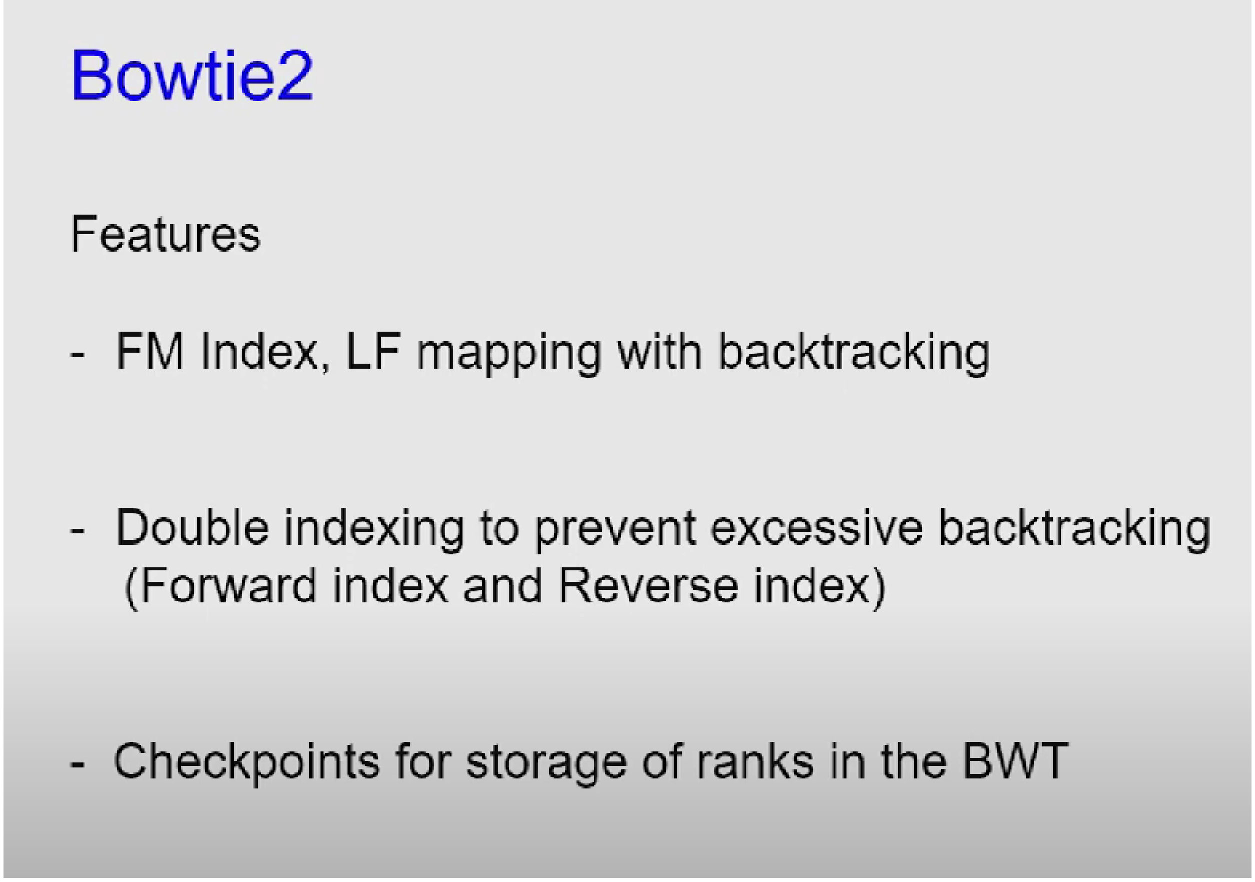
Bowtie/Bowtie2

BWA

BWA-MEM

BWA-SA

So, the tool that we will be discussing is Bowtie version 2, and this was published in 2009. This is the paper where they actually describe this method. So, what are the features of this tool, Bowtie 2? So, we will describe this and look at the tool a little bit, and then we will go and install it, set it up, and do the mapping ourselves. So, what are the features of this tool? So, it utilizes this FM index LF mapping with backtracking. So, which means it will allow you to map reads with mismatches right, and it also uses double indexing right to prevent excessive backtracking. So, it is called forward index and reverse index, right? So, you will see these index files when you actually go and do the hands-on, and there are of course, checkpoints for storing ranks in the BW. So, we discussed this in the last class. So, if you want to store all the ranks, that will take too much space in the memory, or if you want to store them on the hard drive, that will take too much storage.



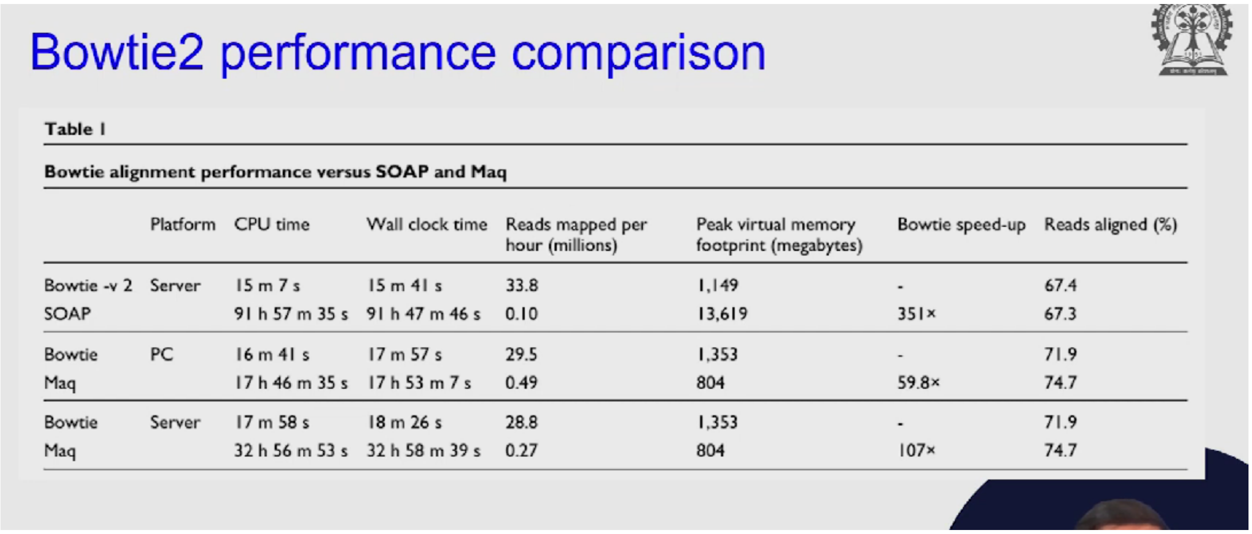
Bowtie2

Features

- FM Index, LF mapping with backtracking
- Double indexing to prevent excessive backtracking (Forward index and Reverse index)
- Checkpoints for storage of ranks in the BWT

So, there is no need to optimize this, and you can decide how often you store these ranks, so this method also utilizes these checkpoints. So, if you compare the performance of Bowtie 2, you can see this.

So, published by the authors themselves right against two tools, one is SOAP and another is MAC, So, again, we have discussed the other methods. So, SOAP uses hash table-based methods, and MAC utilizes something like creating an index from the reads and then searching through the references. So, what you will see here, right on the left, are the names of the tools. So, here in the first set, there is a comparison between Bowtie 2 version 2 versus the SOAP, and the platform is the server, and you can see the CPU time or wall clock time.



Bowtie2 performance comparison

Table 1

Bowtie alignment performance versus SOAP and Maq

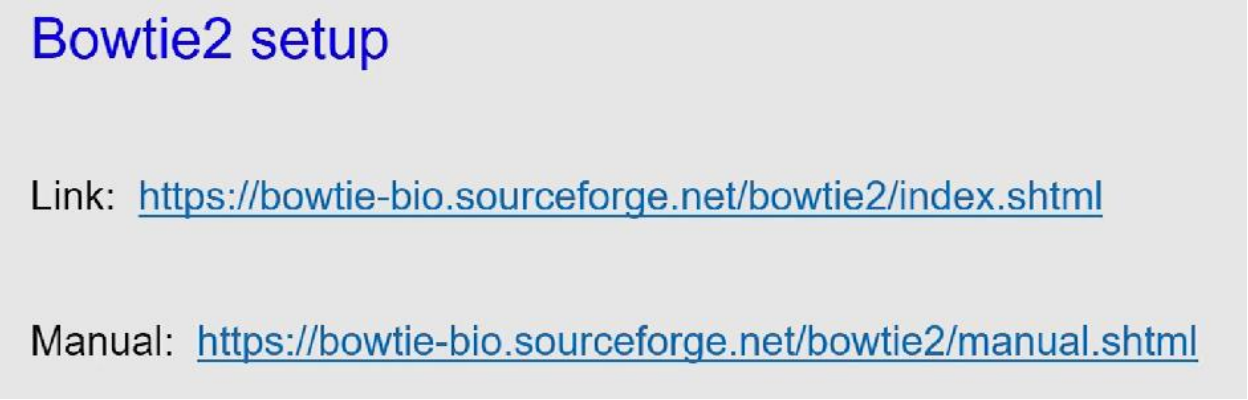
	Platform	CPU time	Wall clock time	Reads mapped per hour (millions)	Peak virtual memory footprint (megabytes)	Bowtie speed-up	Reads aligned (%)
Bowtie -v 2	Server	15 m 7 s	15 m 41 s	33.8	1,149	-	67.4
SOAP		91 h 57 m 35 s	91 h 47 m 46 s	0.10	13,619	351×	67.3
Bowtie	PC	16 m 41 s	17 m 57 s	29.5	1,353	-	71.9
Maq		17 h 46 m 35 s	17 h 53 m 7 s	0.49	804	59.8×	74.7
Bowtie	Server	17 m 58 s	18 m 26 s	28.8	1,353	-	71.9
Maq		32 h 56 m 53 s	32 h 58 m 39 s	0.27	804	107×	74.7

So, the time taken to do a certain number of mappings right and you can see Bowtie 2 is significantly faster right much much faster here right. So, you can see it took about 15 minutes to map those reads, whereas SOAP actually took 91 hours, ok? So, you can see the difference right, and there you can also see this column here, which is mapped per hour in millions. So, Bowtie 2 could actually map 33.8 million reads per hour, whereas SOAP actually mapped only 0.

1 million reads per hour. So, there is a substantial difference in speed. Also, you can see the virtual memory footprint in megabytes. So, how much memory is it taking? Bowtie 2 took about 1.

2 GB, right? So, it is 1.15 MB, right? So, sorry, 1150 MB. So, it is about 1.2 GB, whereas, for SOAP, it is much higher. So, it is around 14 GB, right? So, that is a significant difference, right? That means you can run Bowtie 2 on your desktop, whereas you cannot run the other program on your desktop computer. So, you can see the speed up right. So, Bowtie 2 was much faster, and there was also no difference in the percentage of reads that were aligned right. So, that means the performance is not the output of the results; there is almost no difference, but this method, Bowtie 2, is much faster, and it also takes much less memory. So, similarly, there are other comparisons in Bowtie Mac, right? So, you can see in PC and server versions, and in all these comparisons, you can see these differences. So, there is a substantial difference in speed between Bowtie and Mac, and you can also see the memory usage. So, memory usage on Mac is slightly lower, but there is a tradeoff, right? So, you actually utilize more time; you need more time, right? So, both of these are manageable, right? So, 800 MB and 1.4 GB are not too much of a difference, and you can run

them on the desktop computer. Whereas, whereas Bowtie excels, it is actually much faster, utilizing slightly more RAM. So, again, the performance was not all that different. So, Bowtie could map about 73 percent of the text, whereas both Macs could actually map about 75 percent of the text. So, all right. So, what we are going to do now is we are going to look at Bowtie 2 right. We will set this up right, and we will start looking at the functions, look at the options, and we will also run them to actually generate our own mapping. So, let us go ahead with that right.

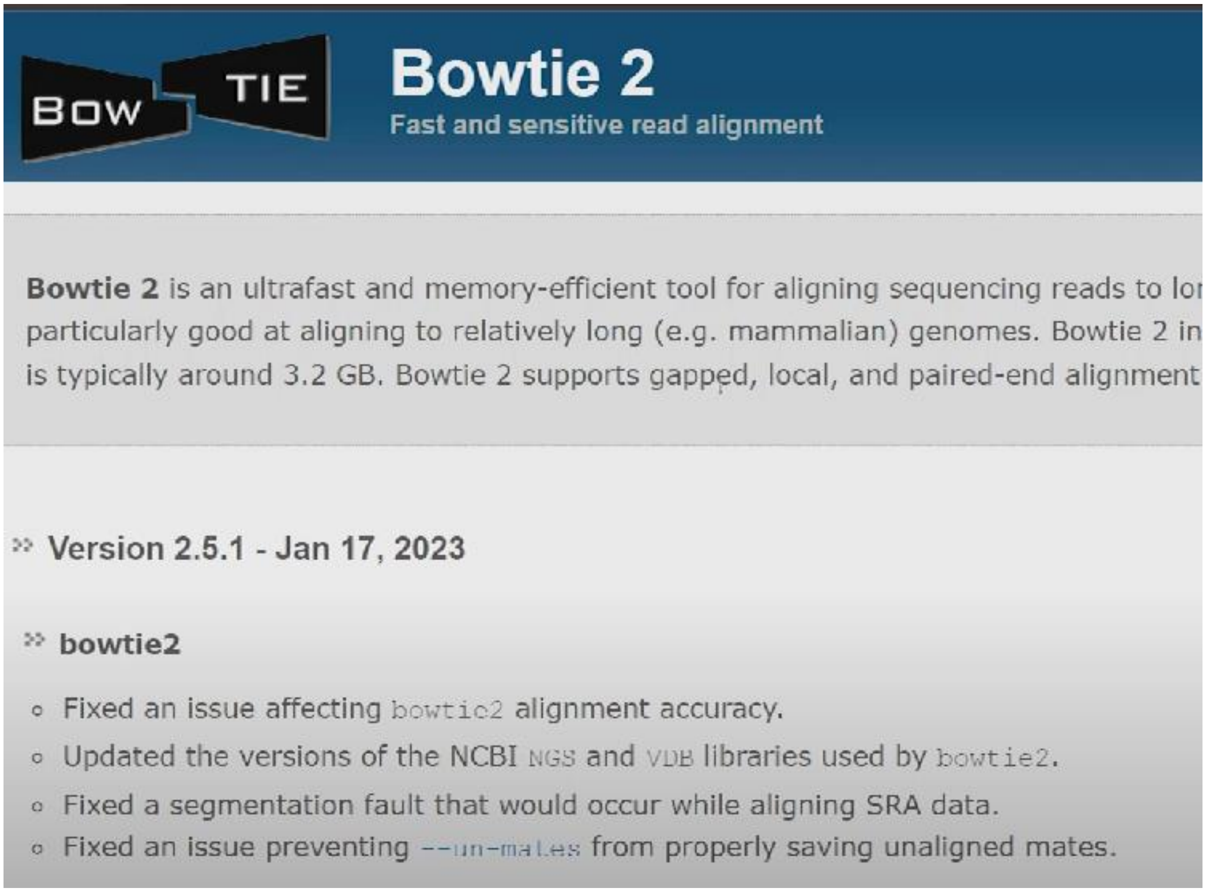


Bowtie2 setup

Link: https://bowtie-bio.sourceforge.net/bowtie2/index.shtml

Manual: https://bowtie-bio.sourceforge.net/bowtie2/manual.shtml

So, here is the link from where you can download the Bowtie 2 tool. It is a free tool that you can download and install, and here is a detailed manual right giving all the instructions that are required for running this tool. So, what we are going to do is now we go to this link and we will download and set this up for our system, and then we will run Bowtie 2 to map our reads. So, let us go there right and we will talk about all the options that are there. So, if you go there right,. So, if you go to this link right here is Bowtie 2 you can see right and it gives you some details like some descriptions of our Bowtie 2 right and what you will find here if you come down here right it is



Bowtie 2

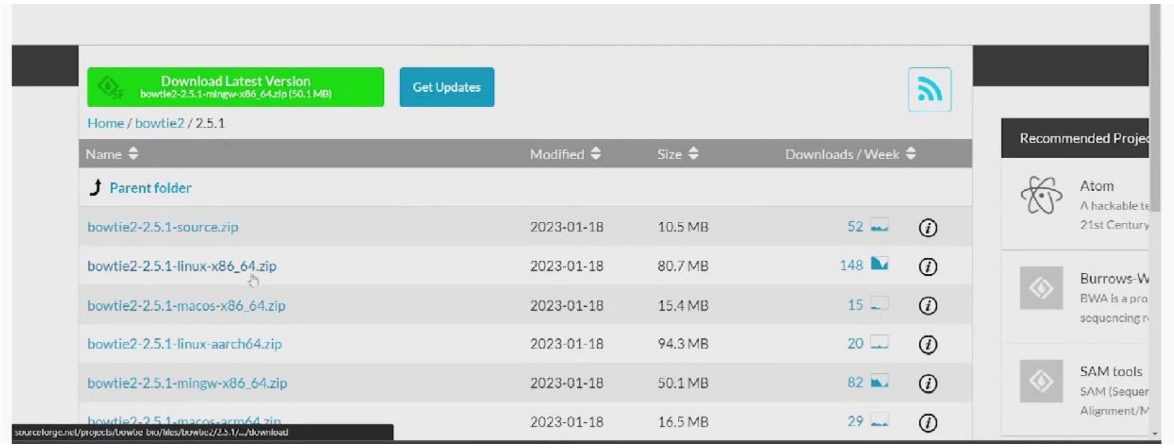
Fast and sensitive read alignment

Bowtie 2 is an ultrafast and memory-efficient tool for aligning sequencing reads to lo
particularly good at aligning to relatively long (e.g. mammalian) genomes. Bowtie 2 in
is typically around 3.2 GB. Bowtie 2 supports gapped, local, and paired-end alignment

Version 2.5.1 - Jan 17, 2023

bowtie2

- Fixed an issue affecting `bowtie2` alignment accuracy.
- Updated the versions of the NCBI `NGS` and `VDB` libraries used by `bowtie2`.
- Fixed a segmentation fault that would occur while aligning SRA data.
- Fixed an issue preventing `--un-mates` from properly saving unaligned mates.

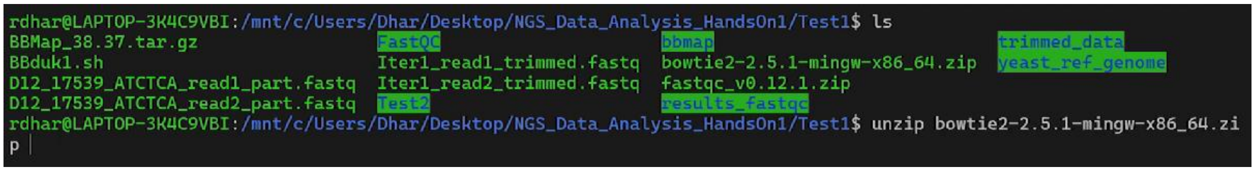


Home / bowtie2 / 2.5.1

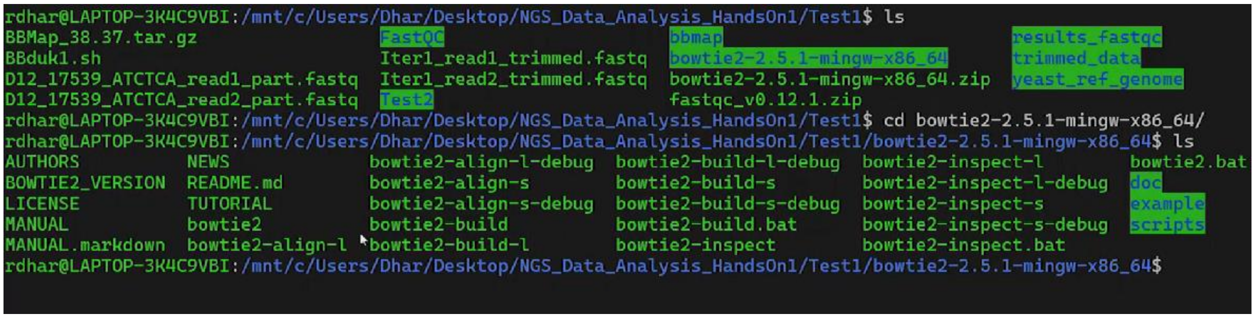
Name	Modified	Size	Downloads / Week
Parent folder			
bowtie2-2.5.1-source.zip	2023-01-18	10.5 MB	52
bowtie2-2.5.1-linux-x86_64.zip	2023-01-18	80.7 MB	148
bowtie2-2.5.1-macos-x86_64.zip	2023-01-18	15.4 MB	15
bowtie2-2.5.1-linux-aarch64.zip	2023-01-18	94.3 MB	20
bowtie2-2.5.1-mingw-x86_64.zip	2023-01-18	50.1 MB	82
bowtie2-2.5.1-macos-arm64.zip	2023-01-18	16.5 MB	29

sourceforge.net/projects/bowtie-bio/files/bowtie2/2.5.1/.../download

Bowtie 2 version 2.5.1 right and if you go to this link right you can download the latest version which is a zip version or you can also look for your specific system specific version right for example, if you are running Mac you will probably want to install this one right. So, you can download the version appropriate for your system, and then you can work with that system. So, what I have done is I have downloaded the latest version of this one, and I have stored it right; otherwise, it will take a bit of time. So, I will now go into that folder right where I have actually stored this downloaded version. So, here you see this right. So, this is the Bowtie 2 latest version, and this is the zip folder, right? So, what will we do now? We want to unzip, right? So, hopefully you remember these commands right that you have discussed, and we are in that folder where we are doing all these operations, ok? So, first, let us look at this.

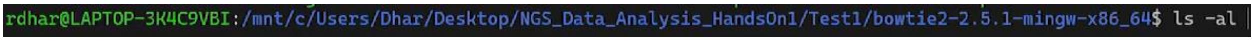


```
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$ ls
BBMap_38.37.tar.gz                 FastQC                     bbmap                           trimmed_data
BBduk1.sh                          Iter1_read1_trimmed.fastq  bowtie2-2.5.1-mingw-x86_64.zip  yeast_ref_genome
D12_17539_ATCTCA_read1_part.fastq  Iter1_read2_trimmed.fastq  fastqc_v0.12.1.zip
D12_17539_ATCTCA_read2_part.fastq  Test2                      results_fastqc
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$ unzip bowtie2-2.5.1-mingw-x86_64.zi
p
```



```
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$ ls
BBMap_38.37.tar.gz                 FastQC                     bbmap                           results_fastqc
BBduk1.sh                          Iter1_read1_trimmed.fastq  bowtie2-2.5.1-mingw-x86_64      trimmed_data
D12_17539_ATCTCA_read1_part.fastq  Iter1_read2_trimmed.fastq  bowtie2-2.5.1-mingw-x86_64.zip  yeast_ref_genome
D12_17539_ATCTCA_read2_part.fastq  Test2                      fastqc_v0.12.1.zip
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$ cd bowtie2-2.5.1-mingw-x86_64/
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1/bowtie2-2.5.1-mingw-x86_64$ ls
AUTHORS          NEWS               bowtie2-align-l-debug  bowtie2-build-l-debug  bowtie2-inspect-l        bowtie2.bat
BOWTIE2_VERSION  README.md          bowtie2-align-s        bowtie2-build-s        bowtie2-inspect-l-debug  doc
LICENSE          TUTORIAL           bowtie2-align-s-debug  bowtie2-build-s-debug  bowtie2-inspect-s        example
MANUAL           bowtie2            bowtie2-build          bowtie2-build.bat      bowtie2-inspect-s-debug  scripts
MANUAL.markdown  bowtie2-align-l    bowtie2-build-l        bowtie2-inspect        bowtie2-inspect.bat
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1/bowtie2-2.5.1-mingw-x86_64$
```

So, here it is. So, with LS, we see this Bowtie program in the zip format, and to unzip it, we simply write unzip bowtie dot zip and write the name of the file. So, this will create this unzip version. So, you can simply clear the screen and then again look at this, and you will see that there is a folder now with this Bowtie 2 right. So, this Bowtie 2 folder contains the programs that we are going to use, ok? So, we can go in there and see the programs, and here are the programs that we can use for our mapping, ok? So, the setting up, as you have seen, is very easy; there are no complications. You just download the zip file, unzip it, and in your system, you might have to change permission, but here I think the permission is there.

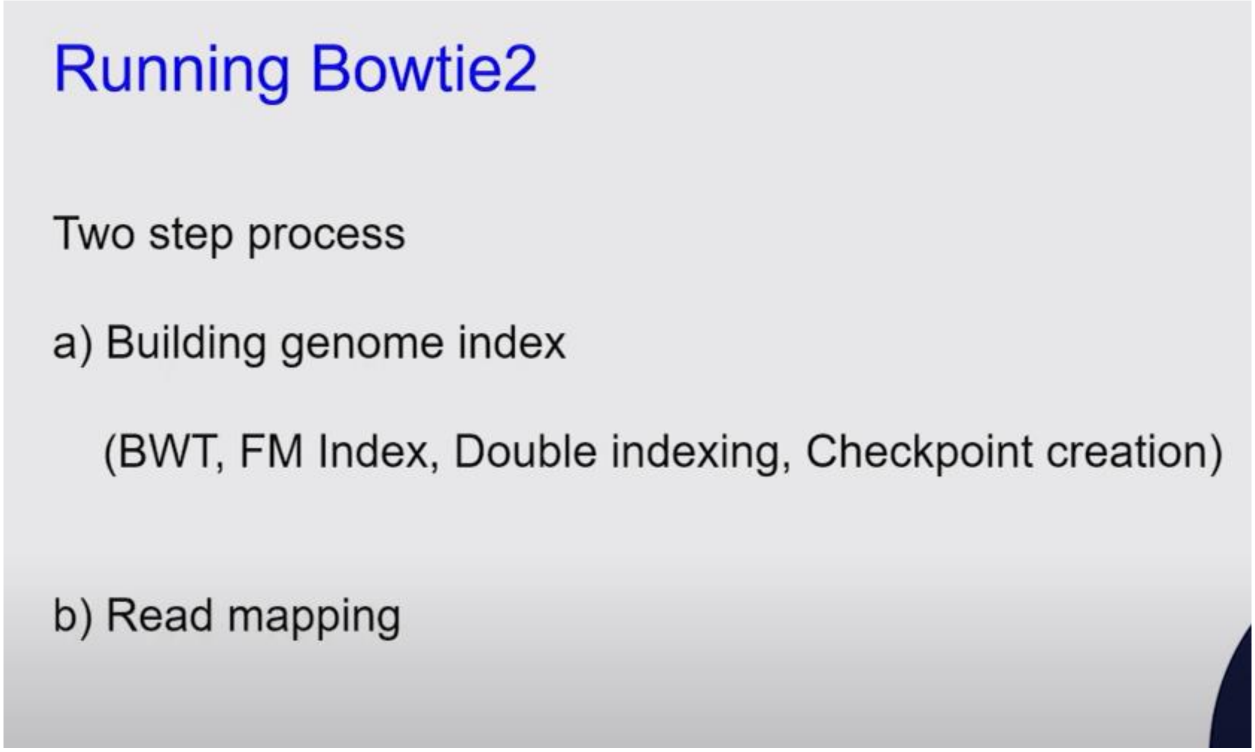


```
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1/bowtie2-2.5.1-mingw-x86_64$ ls -al
```

So, we can also check the permission right by saying ls minus al right. So, for these programs, especially Bowtie, Bowtie 2 L, bowtie-align-l, bowtie-2-align-s, etcetera, these are the executables that will run when you want to map read. So, you want to make sure that these files have this execution permission right. So, here in this system, we have this execution permission X, but for all of them in your system, it might not be there.

So, you want to make sure that you have this execution permission, ok? If you have that, and that is it, we are actually ready to go, ok? We can utilize this and run the mapping, ok? So, we will come out right, and we will be in the test right. So, now one of the things we will discuss is how we actually run this course. So, there are different ways to do this, but I will go back and show you some of these steps, ok? So, how do you actually run this? So, we will also look at the manual right once we actually start running these commands ok? So, it appears that this is a two-step process, right? So, you would have to first build something called a genome index, right? What happens in genome indexing is actually this burrow-wheeler transform, this fm index creation, double indexing, and checkpoint creation. So, all these things are happening when you run this genome indexing part ok.

So, that is the first step ok you would have to do this and once you have done this for a genome right you can use that for all sorts of mapping right you do not have to create that index again and again ok. So, once you have created that index you can simply store it in your hard drive and you can use this whenever you want your mapping reads from that organism ok. So, this is the first step ok and there is a different command to do this and in the next step once you have built this index you you do the read mapping ok.



Running Bowtie2

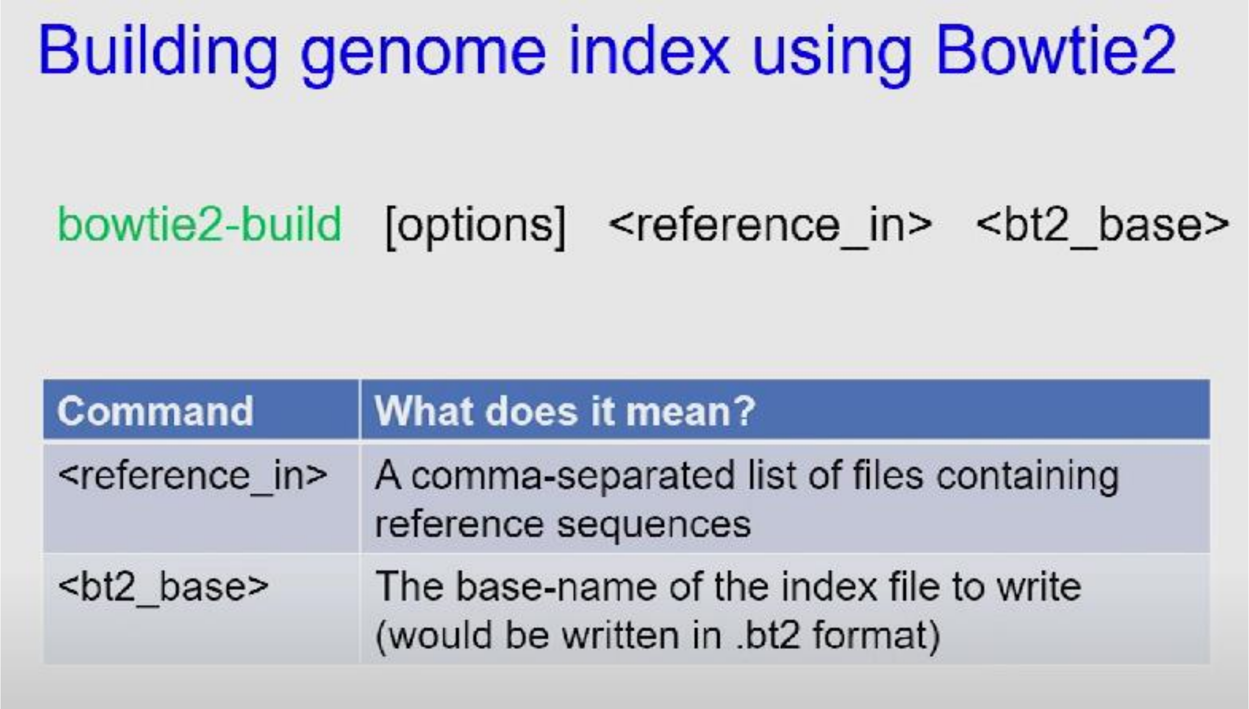
Two step process

a) Building genome index

(BWT, FM Index, Double indexing, Checkpoint creation)

b) Read mapping

So, there are two different commands which we will see ah in a moment we will also use those commands for our purpose. So, the first one is, ah, the first step right is the building of the genome index using Bowtie 2.

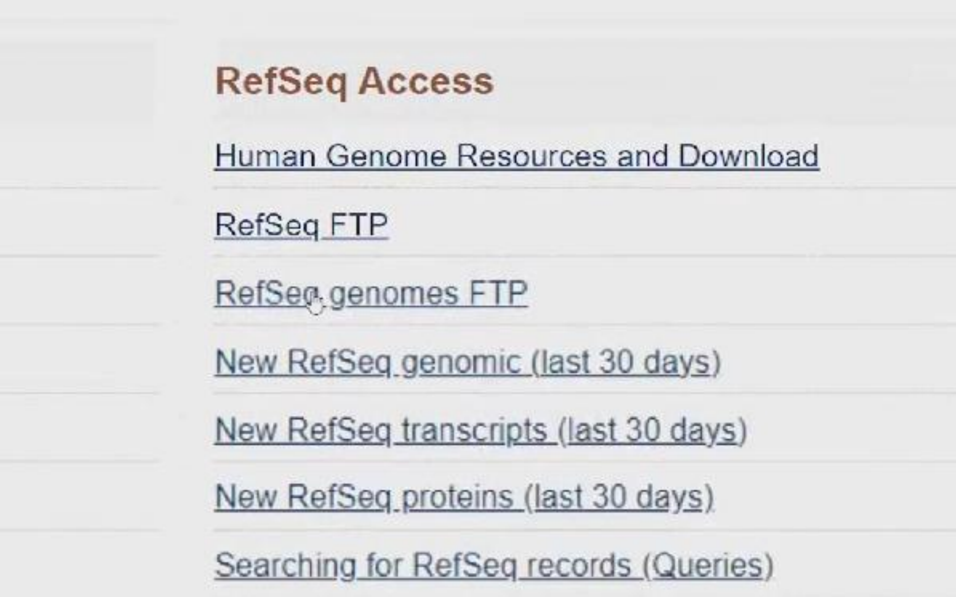


Building genome index using Bowtie2

```
bowtie2-build   [options]   <reference_in>   <bt2_base>
```

Command	What does it mean?
<reference_in>	A comma-separated list of files containing reference sequences
<bt2_base>	The base-name of the index file to write (would be written in .bt2 format)

The command is bowtie2 build ok. So, this is a specific command to build these genome indexes ok. So, this is what I have. I am showing you how you should write this command in the terminal or how you should run this command in the terminal. So, this is the Bowtie2 build, followed by certain options that might be useful in certain cases, followed by this reference. This is the list of reference genome sequences, right? It could be a single AH reference genome or it could be multiple AH reference genomes, right? So, maybe you have sequenced one reference genome along with certain other fragments, and then you want to use this AH multiple reference genome to build the index. And then you have BT 2 base, which means you want to give the base file name for the index file. So, what it will do right when you run this Bowtie2 build is that it will create this index file. So, containing all this information if you have indexes, etcetera, So, this BT-2 base file will then be useful for the mapping process, right? So, you want to specify the name that will be used again for the mapping process, ok? So, what are the options, and what is the explanation? So, as I mentioned, reference ah is a comma-separated list of files containing reference sequences. So, it could be just one single file or it could be multiple files, which should be comma-separated right. So, there should be a comma between these files, ok? And then you have this bt2 base, which is the base name of the index file, which will be written in this dot bt2 format, right? So, you will see when we actually run this, we will find this out, and we will see these files created after we run this command, ok? Now, the question is, how do you find the reference sequences? So, you might be wondering, OK, with reference sequence, how do you find this reference sequence? So, there are places where you can find reference genome data, and that is the store in this NCBI link, which is a good resource for finding this reference genome data. So, we will have a look at where you can actually find this reference genome data very quickly. So, if you go to this link here in the refsec, it is called the refsec, right? So, these are the reference sequences, and you can simply go to refsec ftp or refsec genomes ftp right?



RefSeq Access

Human Genome Resources and Download

RefSeq FTP

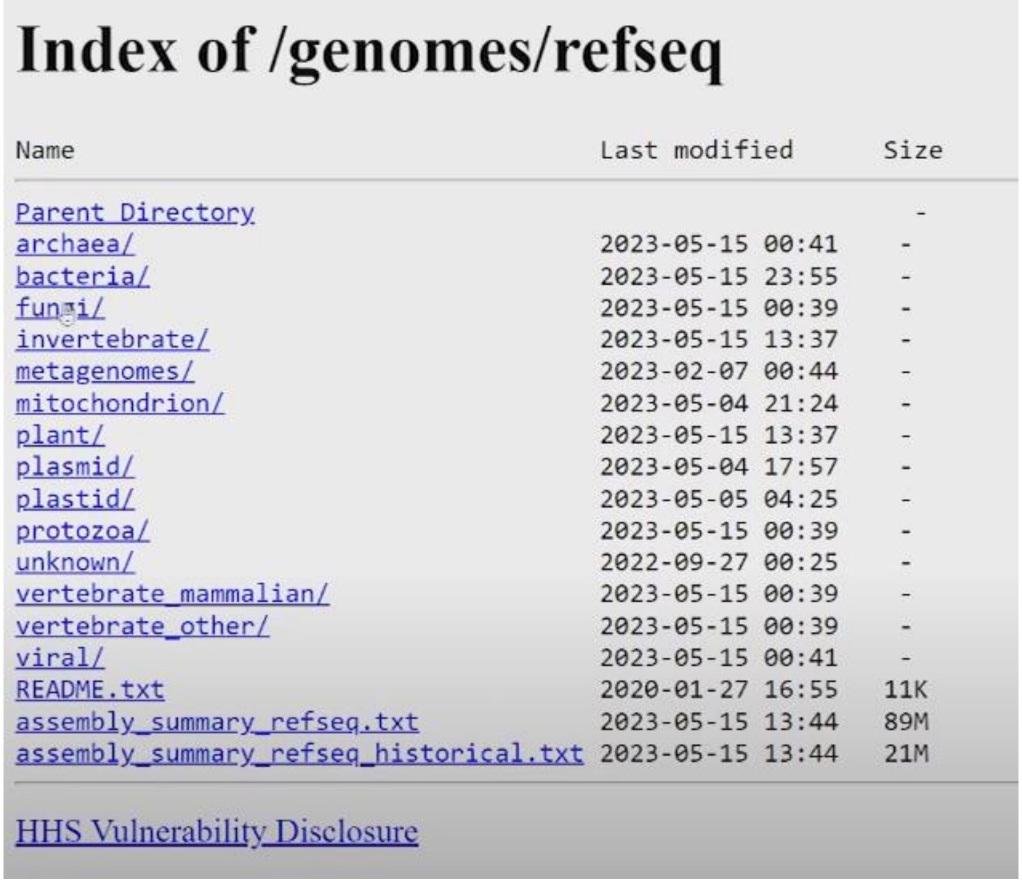
RefSeq genomes FTP

New RefSeq genomic (last 30 days)

New RefSeq transcripts (last 30 days)

New RefSeq proteins (last 30 days)

Searching for RefSeq records (Queries)

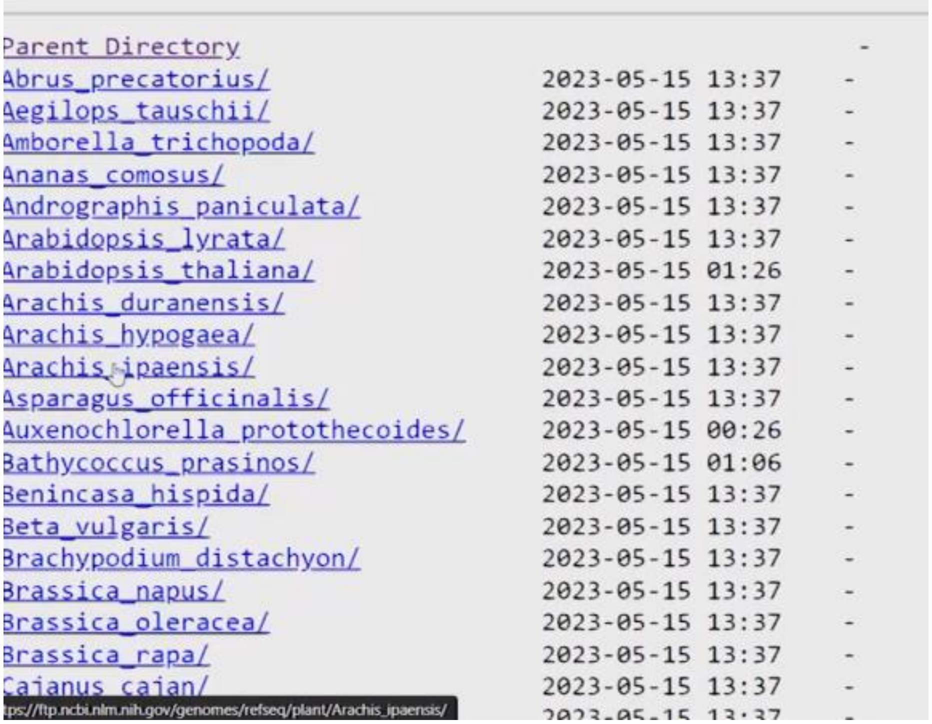


Index of /genomes/refseq

```
Name                                       Last modified      Size

Parent Directory                                               -
archaea/                                   2023-05-15 00:41    -
bacteria/                                  2023-05-15 23:55    -
fungi/                                     2023-05-15 00:39    -
invertebrate/                              2023-05-15 13:37    -
metagenomes/                               2023-02-07 00:44    -
mitochondrion/                             2023-05-04 21:24    -
plant/                                     2023-05-15 13:37    -
plasmid/                                   2023-05-04 17:57    -
plastid/                                   2023-05-05 04:25    -
protozoa/                                  2023-05-15 00:39    -
unknown/                                   2022-09-27 00:25    -
vertebrate_mammalian/                      2023-05-15 00:39    -
vertebrate_other/                          2023-05-15 00:39    -
viral/                                     2023-05-15 00:41    -
README.txt                                 2020-01-27 16:55   11K
assembly_summary_refseq.txt                2023-05-15 13:44   89M
assembly_summary_refseq_historical.txt     2023-05-15 13:44   21M
```

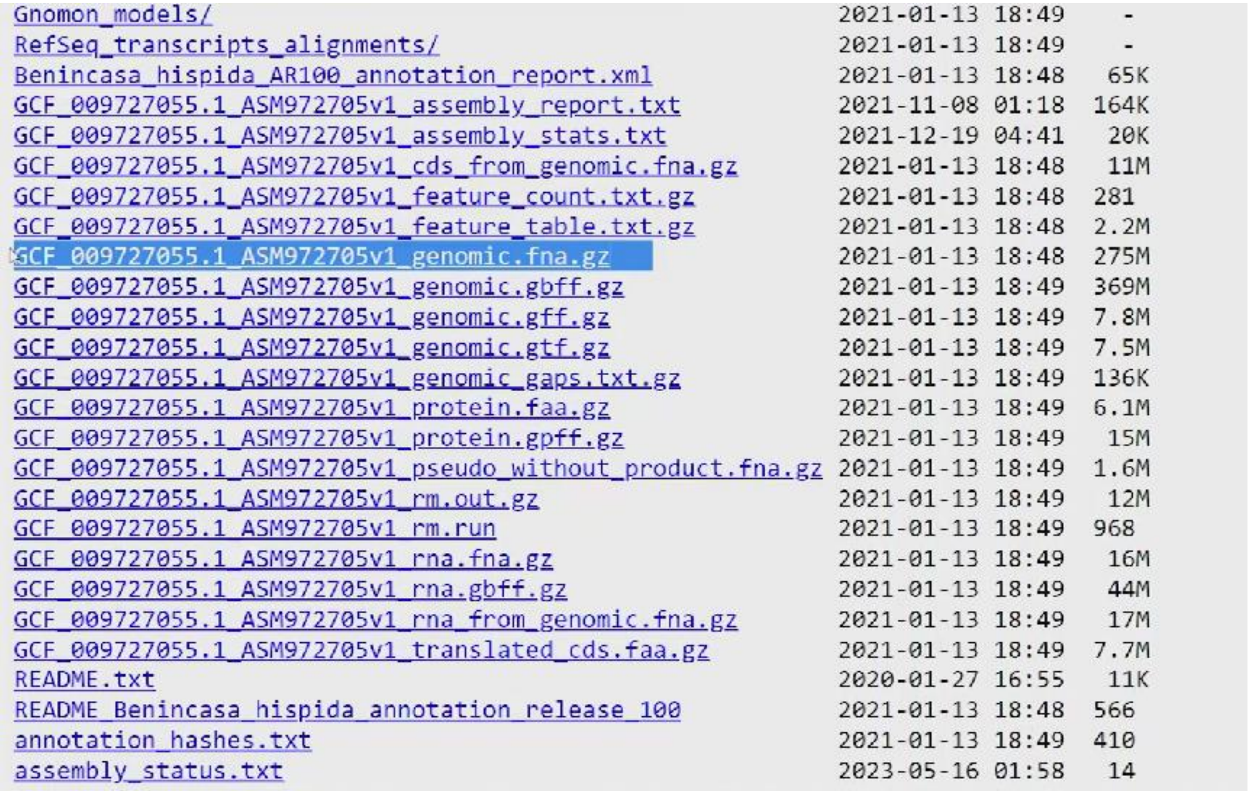
HHS Vulnerability Disclosure



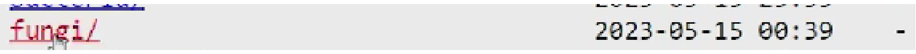
```
Parent Directory                                        -
Abrus_precatorius/               2023-05-15 13:37      -
Aegilops_tauschii/               2023-05-15 13:37      -
Amborella_trichopoda/            2023-05-15 13:37      -
Ananas_comosus/                  2023-05-15 13:37      -
Andrographis_paniculata/         2023-05-15 13:37      -
Arabidopsis_lyrata/              2023-05-15 13:37      -
Arabidopsis_thaliana/            2023-05-15 01:26      -
Arachis_duranensis/              2023-05-15 13:37      -
Arachis_hypogaea/                2023-05-15 13:37      -
Arachis_ipaensis/                2023-05-15 13:37      -
Asparagus_officinalis/           2023-05-15 13:37      -
Auxenochlorella_protothecoides/  2023-05-15 00:26      -
Bathycoccus_prasinos/            2023-05-15 01:06      -
Benincasa_hispida/               2023-05-15 13:37      -
Beta_vulgaris/                   2023-05-15 13:37      -
Brachypodium_distachyon/         2023-05-15 13:37      -
Brassica_napus/                  2023-05-15 13:37      -
Brassica_oleracea/               2023-05-15 13:37      -
Brassica_rapa/                   2023-05-15 13:37      -
Cajanus_cajan/                   2023-05-15 13:37      -
                                 2023-05-15 13:37      -
```

tps://ftp.ncbi.nlm.nih.gov/genomes/refseq/plant/Arachis_ipaensis/

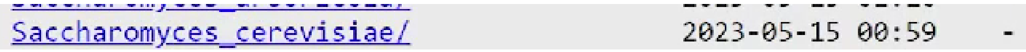
So, maybe I will zoom in with the right. So, if you go back right here is the link to right RefSeq access. Here you have RefSeq genomes ftp where you will find these reference genome sequences, and they are organized by different classes right archaea bacteria, fungi, invertebrate, etcetera, or plants. So, you can go and click any of these links, and you can find a list of species for which you will have the reference genome data. Now, for reference genomes, right? So when you click on these links, you go inside, and what you will get is something like the latest assembly version or representative. So, representative ones will ah be giving you the reference genome, and what you need is the FASTA format file ok. Remember for running this ah bowtie2 indexing, it is preferable to have the FASTA file right, and you have to search for the genomic FASTA ok. So, here you will find some of these right. So, for example, this dot fna is the FASTA file, or this dot fa is the FASTA file, right? So here is the FASTA file, right?



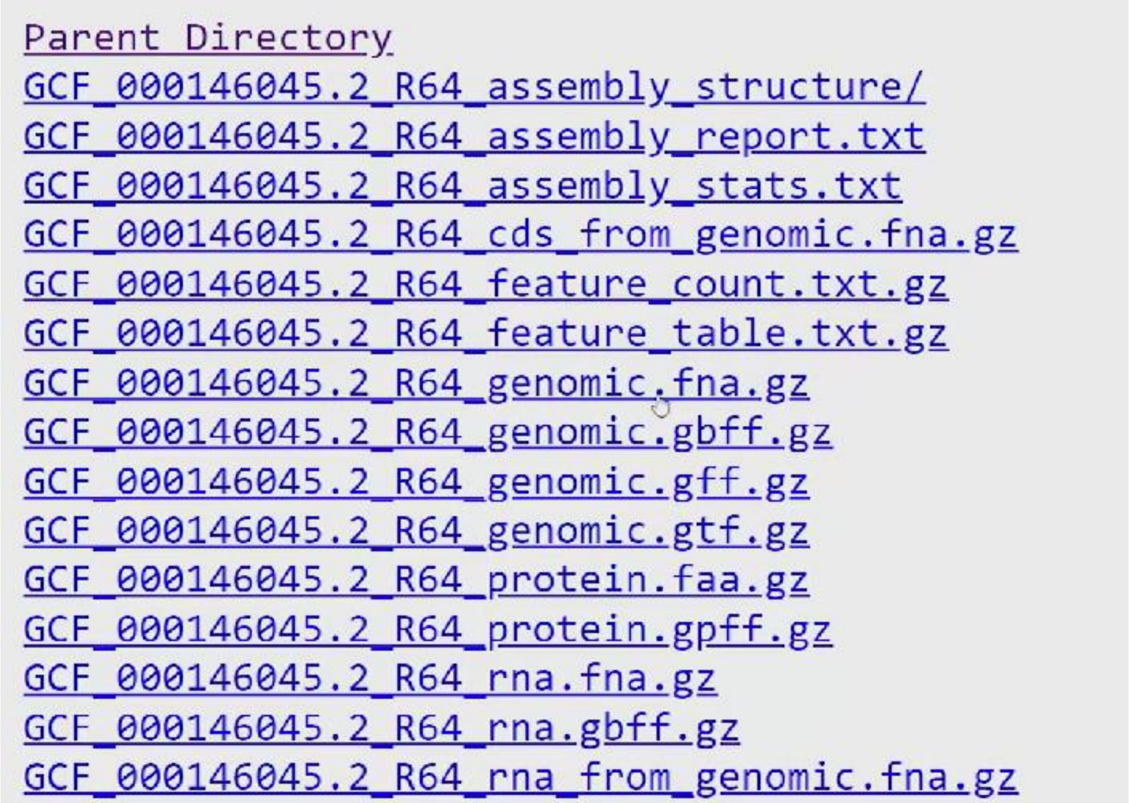
```
Gnomon_models/                                           2021-01-13 18:49    -
RefSeq_transcripts_alignments/                           2021-01-13 18:49    -
Benincasa_hispida_AR100_annotation_report.xml            2021-01-13 18:48   65K
GCF_009727055.1_ASM972705v1_assembly_report.txt          2021-11-08 01:18  164K
GCF_009727055.1_ASM972705v1_assembly_stats.txt           2021-12-19 04:41   20K
GCF_009727055.1_ASM972705v1_cds_from_genomic.fna.gz      2021-01-13 18:48   11M
GCF_009727055.1_ASM972705v1_feature_count.txt.gz         2021-01-13 18:48  281
GCF_009727055.1_ASM972705v1_feature_table.txt.gz         2021-01-13 18:48  2.2M
GCF_009727055.1_ASM972705v1_genomic.fna.gz               2021-01-13 18:48  275M
GCF_009727055.1_ASM972705v1_genomic.gbff.gz              2021-01-13 18:49  369M
GCF_009727055.1_ASM972705v1_genomic.gff.gz               2021-01-13 18:49  7.8M
GCF_009727055.1_ASM972705v1_genomic.gtf.gz               2021-01-13 18:49  7.5M
GCF_009727055.1_ASM972705v1_genomic_gaps.txt.gz          2021-01-13 18:49  136K
GCF_009727055.1_ASM972705v1_protein.faa.gz               2021-01-13 18:49  6.1M
GCF_009727055.1_ASM972705v1_protein.gpff.gz              2021-01-13 18:49   15M
GCF_009727055.1_ASM972705v1_pseudo_without_product.fna.gz 2021-01-13 18:49  1.6M
GCF_009727055.1_ASM972705v1_rm.out.gz                    2021-01-13 18:49   12M
GCF_009727055.1_ASM972705v1_rm.run                       2021-01-13 18:49  968
GCF_009727055.1_ASM972705v1_rna.fna.gz                   2021-01-13 18:49   16M
GCF_009727055.1_ASM972705v1_rna.gbff.gz                  2021-01-13 18:49   44M
GCF_009727055.1_ASM972705v1_rna_from_genomic.fna.gz      2021-01-13 18:49   17M
GCF_009727055.1_ASM972705v1_translated_cds.faa.gz        2021-01-13 18:49  7.7M
README.txt                                               2020-01-27 16:55   11K
README_Benincasa_hispida_annotation_release_100          2021-01-13 18:48  566
annotation_hashes.txt                                    2021-01-13 18:49  410
assembly_status.txt                                      2023-05-16 01:58   14
```



```
fungi/                          2023-05-15 00:39    -
```



```
Saccharomyces_cerevisiae/                 2023-05-15 00:59    -
```



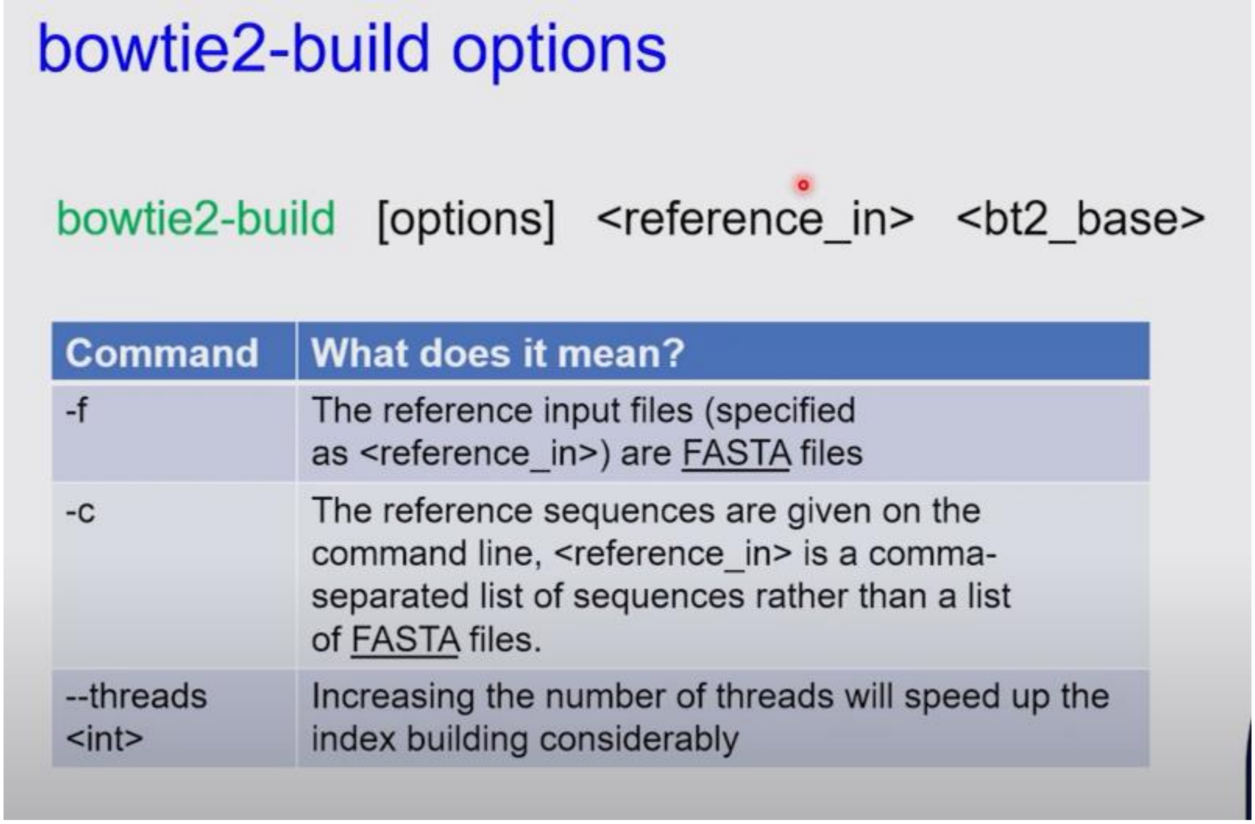
```
Parent Directory
GCF_000146045.2_R64_assembly_structure/
GCF_000146045.2_R64_assembly_report.txt
GCF_000146045.2_R64_assembly_stats.txt
GCF_000146045.2_R64_cds_from_genomic.fna.gz
GCF_000146045.2_R64_feature_count.txt.gz
GCF_000146045.2_R64_feature_table.txt.gz
GCF_000146045.2_R64_genomic.fna.gz
GCF_000146045.2_R64_genomic.gbff.gz
GCF_000146045.2_R64_genomic.gff.gz
GCF_000146045.2_R64_genomic.gtf.gz
GCF_000146045.2_R64_protein.faa.gz
GCF_000146045.2_R64_protein.gpff.gz
GCF_000146045.2_R64_rna.fna.gz
GCF_000146045.2_R64_rna.gbff.gz
GCF_000146045.2_R64_rna_from_genomic.fna.gz
```

```
GCF_000146045.2_R64_genomic.fna.gz
```

So, this fna dot is right; you can see this right. So, this fna file is the FASTA file, and it contains the genomic sequence. It is just compressed right with czip compression, as we have learned about czip compression. So, in our case, the data that we have worked with so far comes from each species. So, let us go in go and find the each genome right. So, here we will go inside right and we will find this right, ok? So, here it is. So, each species is right. So, here is a fungus, and we have Saccharomyces cerevisiae, right? This is the genome that we need reference sequences for, and you can see reference sequences here right. Representative or reference sequences we have here right. We just click on this and you have a lot of files, but what we need is this genomic dot fna ok. So, you can simply click on this and you can download right, and you can then use this file for building the AH index. You would have to unzip right.

So, because this is gz compressed, all right. So, what I have done is that I have actually downloaded this file already and I have this in my system, ok? So, let us now look into Ah. Let us now go back into Ah slides right and look into the options right. So, we have learned about this

Bowtie 2 build and have understood the reference AH file where this reference file will come from. We have talked about bt2 base; this is something we have to specify where the index file will be created, the name of the index file, and then you have something called options. So, these options could be useful in some cases. So, that is, it is good to know what these options are. So, for example, the options that you can give are: ah, this minus f means the reference input file; ah, these are in FASTA format; ah, this is the default option; ah, but you can also specify that we have FASTA files. So, there is another option, which is minus c, which actually you can specify when you are giving the reference sequences in the command line itself instead of the file names ok. So, if you are writing this command, bowtie2 build right instead of reference in file names, right instead of file names in reference in here, right, what you do is instead of reference in files, ah, files here, you put sequences here, ok, but this is quite rare right because the sequences are so huge we would not put them ah in the command line right.

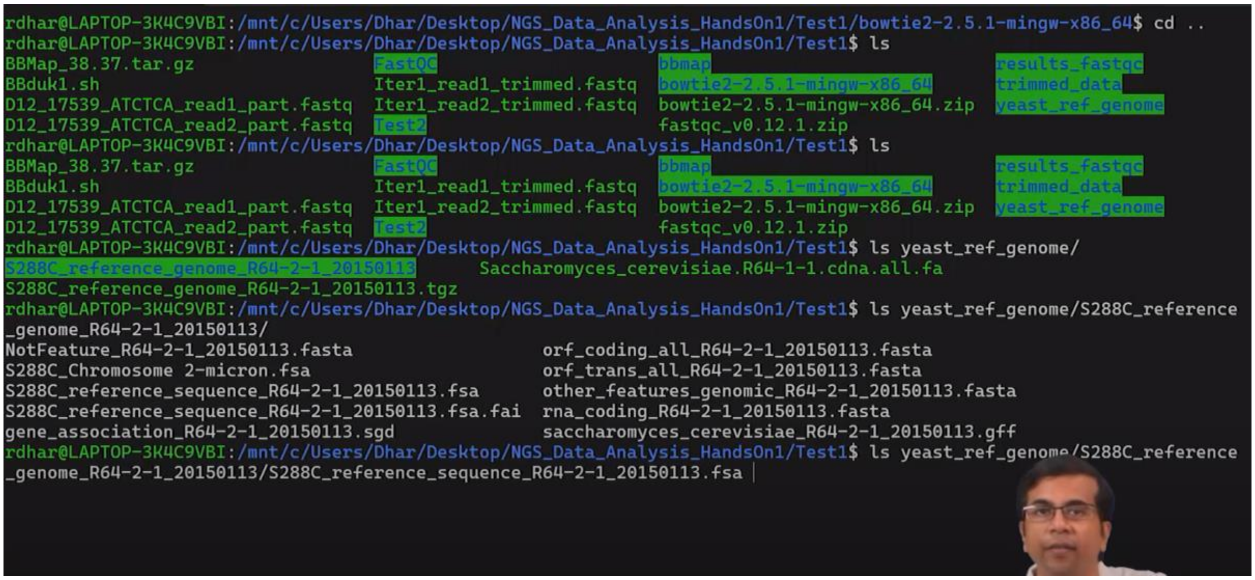


bowtie2-build options

```
bowtie2-build [options] <reference_in> <bt2_base>
```

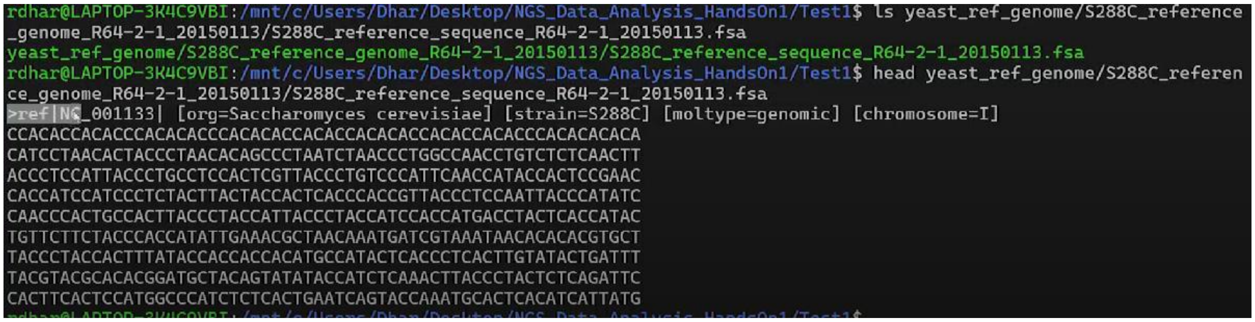
Command	What does it mean?
-f	The reference input files (specified as <reference_in>) are FASTA files
-c	The reference sequences are given on the command line, <reference_in> is a comma-separated list of sequences rather than a list of FASTA files.
--threads <int>	Increasing the number of threads will speed up the index building considerably

So, we will usually put them in files and then direct to the files from this command line ok? There are other options. So, for example, you can use this minus minus thread with a number followed by a number. So, this is an integer, which means there should be an integer, right? So, you will say, for example, minus minus threads 4, right? So, it is kind of parallelizing the process. So, it will actually speed up the index-building process, right? So, this has something to do with the computer system, ok? So, once you have built the AH indexes, So, then you can go ahead and map bridge with bowtie2, ok. So, in this class, in the remaining part of this class, what we will do is actually start ah looking at this build index building process right now that we have downloaded the reference genome. So, what we are going to do now is look at the commands and options and build the index. So, let us do that. Ah, let us go to the command line right, and we will actually see this process in action, ok? So, one of the things you will probably find out. So, ah, one of the things you will probably find out here, right? So, we have the reference genome in this folder. So, here is where I have stored the IS reference genome.



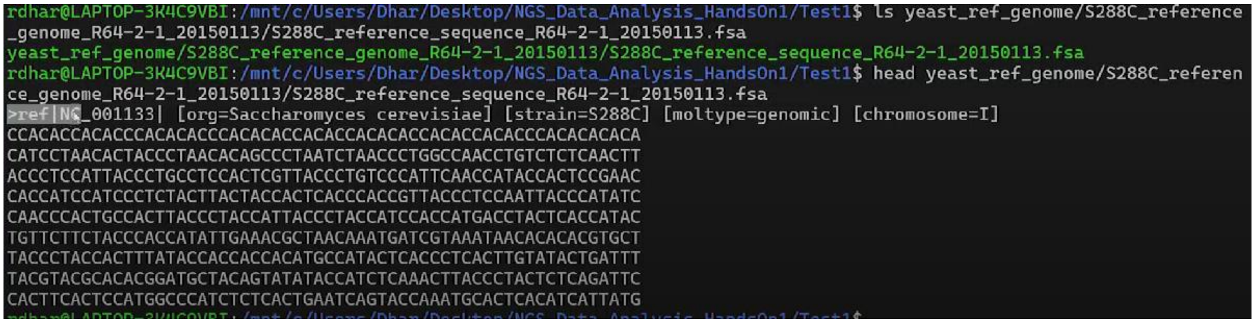
```
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1/bowtie2-2.5.1-mingw-x86_64$ cd ..
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$ ls
BBMap_38.37.tar.gz                 FastQC                      bbmap                               results_fastqc
BBduk1.sh                          Iter1_read1_trimmed.fastq   bowtie2-2.5.1-mingw-x86_64          trimmed_data
D12_17539_ATCTCA_read1_part.fastq  Iter1_read2_trimmed.fastq   bowtie2-2.5.1-mingw-x86_64.zip      yeast_ref_genome
D12_17539_ATCTCA_read2_part.fastq  Test2                       fastqc_v0.12.1.zip
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$ ls
BBMap_38.37.tar.gz                 FastQC                      bbmap                               results_fastqc
BBduk1.sh                          Iter1_read1_trimmed.fastq   bowtie2-2.5.1-mingw-x86_64          trimmed_data
D12_17539_ATCTCA_read1_part.fastq  Iter1_read2_trimmed.fastq   bowtie2-2.5.1-mingw-x86_64.zip      yeast_ref_genome
D12_17539_ATCTCA_read2_part.fastq  Test2                       fastqc_v0.12.1.zip
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$ ls yeast_ref_genome/
S288C_reference_genome_R64-2-1_20150113      Saccharomyces_cerevisiae.R64-1-1.cdna.all.fa
S288C_reference_genome_R64-2-1_20150113.tgz
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$ ls yeast_ref_genome/S288C_reference
_genome_R64-2-1_20150113/
NotFeature_R64-2-1_20150113.fasta                 orf_coding_all_R64-2-1_20150113.fasta
S288C_Chromosome 2-micron.fsa                     orf_trans_all_R64-2-1_20150113.fasta
S288C_reference_sequence_R64-2-1_20150113.fsa     other_features_genomic_R64-2-1_20150113.fasta
S288C_reference_sequence_R64-2-1_20150113.fsa.fai rna_coding_R64-2-1_20150113.fasta
gene_association_R64-2-1_20150113.sgd             saccharomyces_cerevisiae_R64-2-1_20150113.gff
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$ ls yeast_ref_genome/S288C_reference
_genome_R64-2-1_20150113/S288C_reference_sequence_R64-2-1_20150113.fsa
```

You can see right and inside that this is where the file is OK. So, I have a lot of these files, but the one that we need is this one dot fsa ok s 2 8 h c reference sequence r 64 2 ah 1 dot fsa. This is the reference file right again in first format. So, in the first format file, the extensions could be different, right? You can have dot fa dot fsa dot fna, right? You can open up those files, and you can simply check right now whether they are in first-person format, or you can simply write right now and see whether the files are in first-person format. Here you can see this is the header line starting with this greater than sign right and you have the organism name saccharomyces cerevisiae and then from the next line you have the sequences ok.



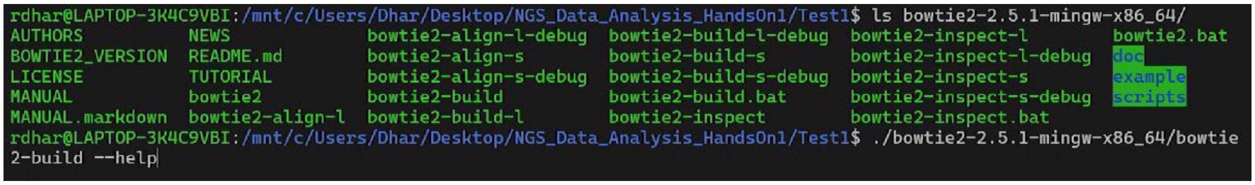
```
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$ ls yeast_ref_genome/S288C_reference
_genome_R64-2-1_20150113/S288C_reference_sequence_R64-2-1_20150113.fsa
yeast_ref_genome/S288C_reference_genome_R64-2-1_20150113/S288C_reference_sequence_R64-2-1_20150113.fsa
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$ head yeast_ref_genome/S288C_referen
ce_genome_R64-2-1_20150113/S288C_reference_sequence_R64-2-1_20150113.fsa
>ref|NC_001133| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=genomic] [chromosome=I]
CCACACCACACCCACACACCCACACACCACACCACACACCACACCACACCCACACACACA
CATCCTAACACTACCCTAACACAGCCCTAATCTAACCCTGGCCAACCTGTCTCTCAACTT
ACCCTCCATTACCCTGCCTCCACTCGTTACCCTGTCCCATTCAACCATACCACTCCGAAC
CACCATCCATCCCTCTACTTACTACCACTCACCCACCGTTACCCTCCAATTACCCATATC
CAACCCACTGCCACTTACCCTACCATTACCCTACCATCCACCATGACCTACTCACCATAC
TGTTCTTCTACCCACCATATTGAAACGCTAACAAATGATCGTAAATAACACACACGTGCT
TACCCTACCACTTTATACCACCACCACATGCCATACTCACCCTCACTTGTATACTGATTT
TACGTACGCACACGGATGCTACAGTATATACCATCTCAAACTTACCCTACTCTCAGATTC
CACTTCACTCCATGGCCCATCTCTCACTGAATCAGTACCAAATGCACTCACATCATTATG
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$
```

So, this is the first file that we need for building the index ok. So, we have the reference file now we want to see how you will run this boti to command ok, boti to build command ok. So, we have installed this bowtie2 in this folder we have extracted and inside this folder we have all these commands and there is this command bowtie2 build ok. So, if you search simply inside this right you will see this command here you have boti to only boti to we have boti to align and then you have boti to build you have other options boti to build l etcetera right.

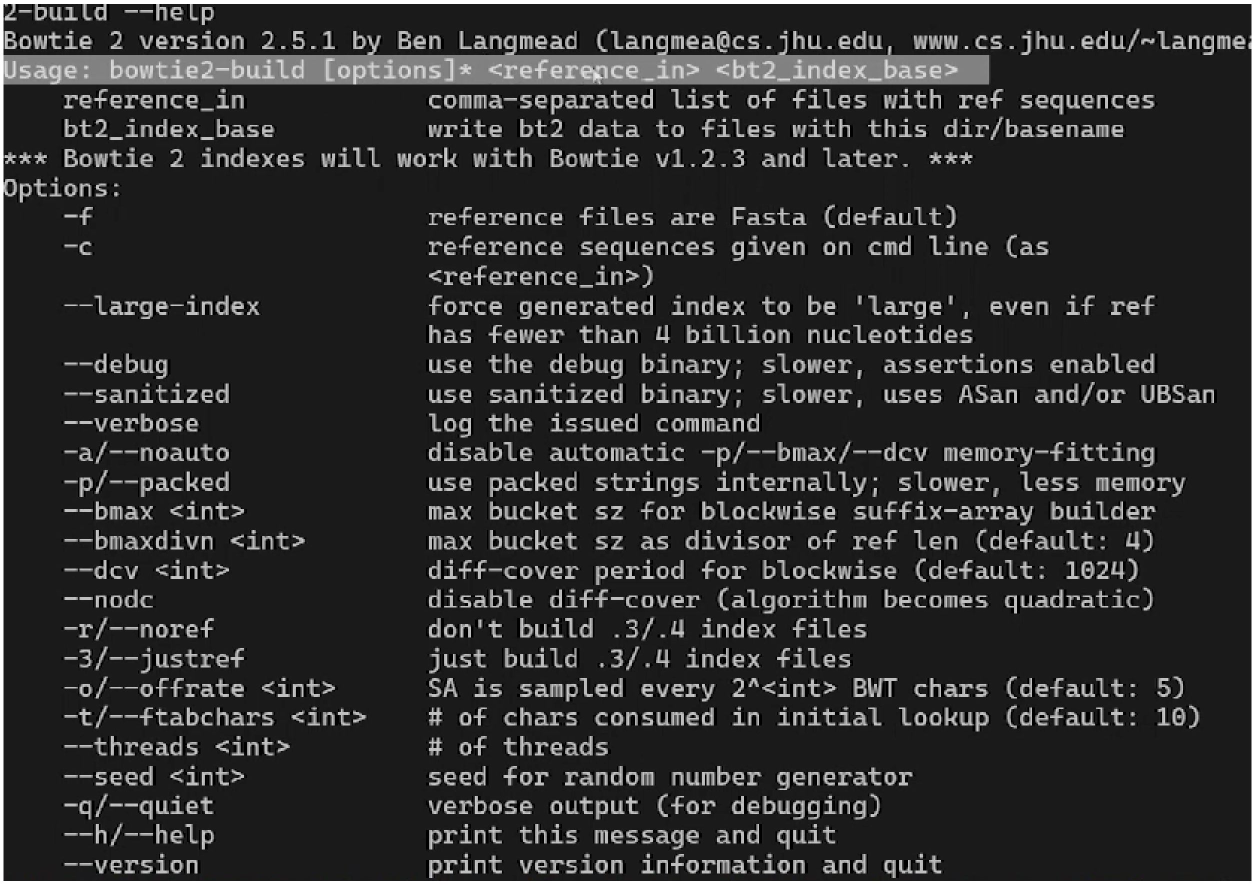


```
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$ ls yeast_ref_genome/S288C_reference
_genome_R64-2-1_20150113/S288C_reference_sequence_R64-2-1_20150113.fsa
yeast_ref_genome/S288C_reference_genome_R64-2-1_20150113/S288C_reference_sequence_R64-2-1_20150113.fsa
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$ head yeast_ref_genome/S288C_referen
ce_genome_R64-2-1_20150113/S288C_reference_sequence_R64-2-1_20150113.fsa
>ref|NC_001133| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=genomic] [chromosome=I]
CCACACCACACCCACACACCCACACACCACACCACACACCACACCACACCCACACACACA
CATCCTAACACTACCCTAACACAGCCCTAATCTAACCCTGGCCAACCTGTCTCTCAACTT
ACCCTCCATTACCCTGCCTCCACTCGTTACCCTGTCCCATTCAACCATACCACTCCGAAC
CACCATCCATCCCTCTACTTACTACCACTCACCCACCGTTACCCTCCAATTACCCATATC
CAACCCACTGCCACTTACCCTACCATTACCCTACCATCCACCATGACCTACTCACCATAC
TGTTCTTCTACCCACCATATTGAAACGCTAACAAATGATCGTAAATAACACACACGTGCT
TACCCTACCACTTTATACCACCACCACATGCCATACTCACCCTCACTTGTATACTGATTT
TACGTACGCACACGGATGCTACAGTATATACCATCTCAAACTTACCCTACTCTCAGATTC
CACTTCACTCCATGGCCCATCTCTCACTGAATCAGTACCAAATGCACTCACATCATTATG
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$
```

So, this bot to build command is what we require, ok? So, what you can do is call this command by typing the path. So, this is the folder in which this command and this function reside, right? So, you have to specify the path and we are using this dot beforehand because just to denote that this is an executable ok. So, dot slash bowtie2 try to build and you can simply run this by saying minus minus help ok. So, what will happen if you run this with minus minus help it will actually show you all the options ok.



```
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$ ls bowtie2-2.5.1-mingw-x86_64/
AUTHORS          NEWS             bowtie2-align-l-debug  bowtie2-build-l-debug  bowtie2-inspect-l        bowtie2.bat
BOWTIE2_VERSION  README.md        bowtie2-align-s        bowtie2-build-s        bowtie2-inspect-l-debug  doc
LICENSE          TUTORIAL         bowtie2-align-s-debug  bowtie2-build-s-debug  bowtie2-inspect-s        example
MANUAL           bowtie2          bowtie2-build          bowtie2-build.bat      bowtie2-inspect-s-debug  scripts
MANUAL.markdown  bowtie2-align-l  bowtie2-build-l        bowtie2-inspect        bowtie2-inspect.bat
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$ ./bowtie2-2.5.1-mingw-x86_64/bowtie
2-build --help
```

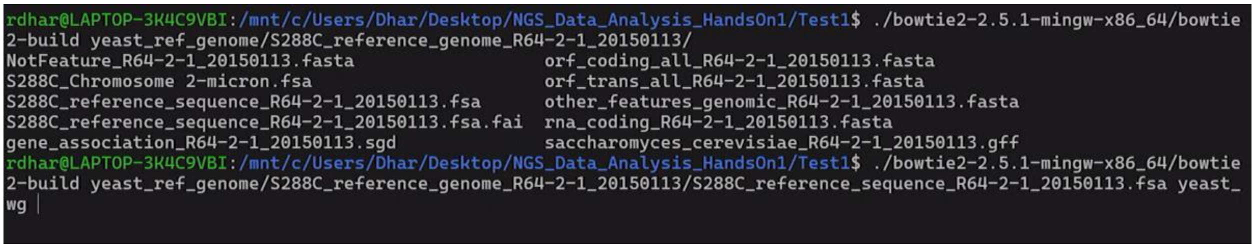


```
2-build --help
Bowtie 2 version 2.5.1 by Ben Langmead (langmea@cs.jhu.edu, www.cs.jhu.edu/~langmea
Usage: bowtie2-build [options]* <reference_in> <bt2_index_base>
    reference_in            comma-separated list of files with ref sequences
    bt2_index_base          write bt2 data to files with this dir/basename
*** Bowtie 2 indexes will work with Bowtie v1.2.3 and later. ***
Options:
    -f                      reference files are Fasta (default)
    -c                      reference sequences given on cmd line (as
                            <reference_in>)
    --large-index           force generated index to be 'large', even if ref
                            has fewer than 4 billion nucleotides
    --debug                 use the debug binary; slower, assertions enabled
    --sanitized             use sanitized binary; slower, uses ASan and/or UBSan
    --verbose               log the issued command
    -a/--noauto             disable automatic -p/--bmax/--dcv memory-fitting
    -p/--packed             use packed strings internally; slower, less memory
    --bmax <int>            max bucket sz for blockwise suffix-array builder
    --bmaxdivn <int>        max bucket sz as divisor of ref len (default: 4)
    --dcv <int>             diff-cover period for blockwise (default: 1024)
    --nodc                  disable diff-cover (algorithm becomes quadratic)
    -r/--noref              don't build .3/.4 index files
    -3/--justref            just build .3/.4 index files
    -o/--offrate <int>      SA is sampled every 2^<int> BWT chars (default: 5)
    -t/--ftabchars <int>    # of chars consumed in initial lookup (default: 10)
    --threads <int>         # of threads
    --seed <int>            seed for random number generator
    -q/--quiet              verbose output (for debugging)
    --h/--help              print this message and quit
    --version               print version information and quit
```

It is taking a bit of time I do not know why ok. So, it is here and it tells you this bowtie2 version the author name etcetera ok and the next line is very very important ok. So, have a have a look carefully ok. So, please check this line because this gives you the usage information ok. So, what it means how you should use this command ok. So, usage is boti to build right then the options and then the reference sequence then the bt2 index space right.

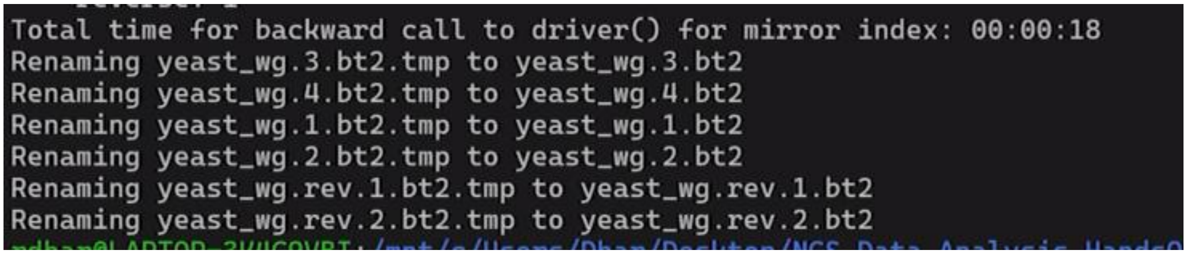
So, the name of this index file that you have to specify is right, and we have discussed this reference in bt2 index space, and then the options are given here, specified here, ok. You can see these multiple options; we have discussed only some of these options, right? For example, we have discussed minus af and minus c, but we have not discussed the other options. So, these are not really very very important, but you can of course, go through and if you see some of these would be useful in your case you can use them, but what I can assure that these are probably not very very much required for daily use ok. So, we can just use this minus f or minus c right or minus minus threads right this is the most important one because we can speed up the process if you have a fast computer where you can run multiple threads or in servers for example, we can use these multiple threads ok.

So, this is this is how this whole process will work right. So, what we can do now is we can follow this right and we can run this commands ok. So, remember now right sorry ok. So, this is the command bowtie2 build right we are calling this we can give options, but let us skip options we do not really need that what we need. So, we have given space, and then what we need is the path to the reference file, ok? So, where is the reference file? Let us keep that path. So, it is inside this folder right in your system; it will be the path where you have stored this downloaded file, ok. So, that is where you should point, ok, and this is the dot fsa right; this is the reference sequence dot fsa in first format; and that is the bt2 index name right. So, that is the base name we need, and I will simply write this as yeast whole genome, okay? So, we are creating this index for the whole genome. So, why am I saying this whole genome? Because sometimes you will see that for some organisms, you can also find a reference transcriptome that is different from the reference genome. So, you can probably understand what a reference transcriptome is: the least of or the sequence of all mRNAs that are expressed, ok, and that will be different from the whole genome reference sequence, because the mRNA expressed will be just part of the genome, there would not be any introns, and the non-coding regions will not be there, right? So, sometimes you will find that, for some organisms, you have a reference transcriptome, and you want to use that reference transcriptome if you are mapping, for example, RNA sequencing data. So, in the case of transcriptome analysis, you generate RNA sequencing data, and in that case, you will want to build an index for the reference transcriptome, and then you will map against that reference transcriptome using those indexes. So, why do you want to build these two separate indexes right for the genome and for the transcriptome if they are available? Because the search against the transcriptome will be much faster because the transcriptome contains a much smaller fraction of the genome. So, the index that will be created will be much smaller in size, and the search process and mapping process will be much faster anyway. So, I am naming this the yeast whole genome right because this is the genomic reference sequence, and we will simply place it in and it will run OK.

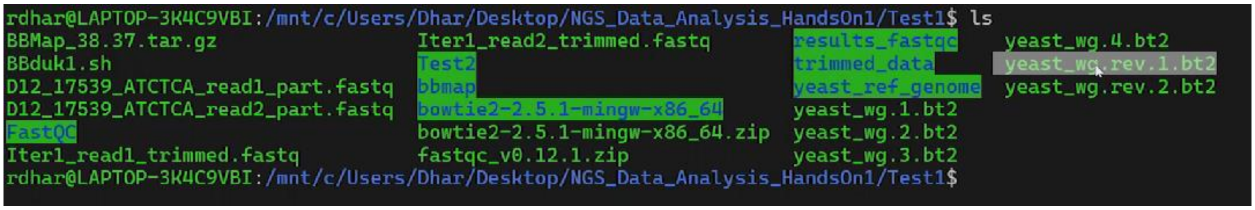


```
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$ ./bowtie2-2.5.1-mingw-x86_64/bowtie
2-build yeast_ref_genome/S288C_reference_genome_R64-2-1_20150113/
NotFeature_R64-2-1_20150113.fasta                 orf_coding_all_R64-2-1_20150113.fasta
S288C_Chromosome 2-micron.fsa                     orf_trans_all_R64-2-1_20150113.fasta
S288C_reference_sequence_R64-2-1_20150113.fsa     other_features_genomic_R64-2-1_20150113.fasta
S288C_reference_sequence_R64-2-1_20150113.fsa.fai rna_coding_R64-2-1_20150113.fasta
gene_association_R64-2-1_20150113.sgd             saccharomyces_cerevisiae_R64-2-1_20150113.gff
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$ ./bowtie2-2.5.1-mingw-x86_64/bowtie
2-build yeast_ref_genome/S288C_reference_genome_R64-2-1_20150113/S288C_reference_sequence_R64-2-1_20150113.fsa yeast_
wg
```

So, it might take a few seconds instead of a few minutes to run, and it is working right. So, you see, it will give you a lot of this process, of course, but not so important really; it is kind of creating these indexes in a stepwise manner, and finally, it will actually generate those indexes. So, you can see right this is going on this Burrows-wheeler transform the FM index calculation the double indexing the checkpoints right all those things were created here ok. So, what you have now you have you have these last commands you can see right you can see this right you have this dot bt2 files ok.

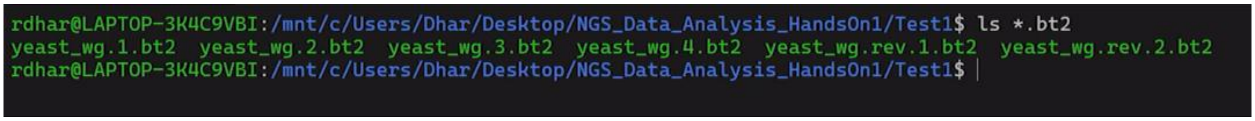


```
Total time for backward call to driver() for mirror index: 00:00:18
Renaming yeast_wg.3.bt2.tmp to yeast_wg.3.bt2
Renaming yeast_wg.4.bt2.tmp to yeast_wg.4.bt2
Renaming yeast_wg.1.bt2.tmp to yeast_wg.1.bt2
Renaming yeast_wg.2.bt2.tmp to yeast_wg.2.bt2
Renaming yeast_wg.rev.1.bt2.tmp to yeast_wg.rev.1.bt2
Renaming yeast_wg.rev.2.bt2.tmp to yeast_wg.rev.2.bt2
```



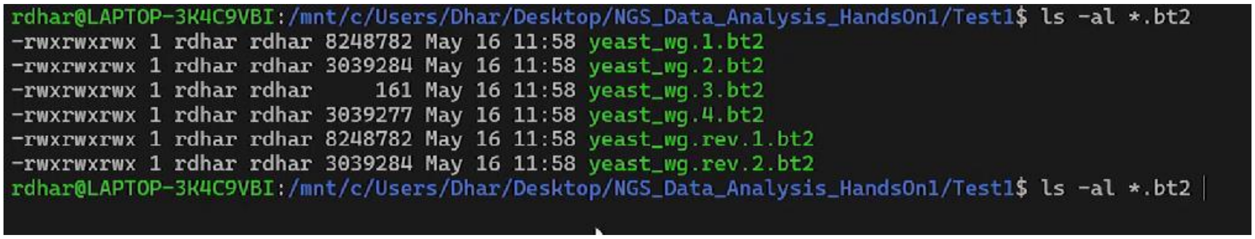
```
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$ ls
BBMap_38.37.tar.gz                 Iter1_read2_trimmed.fastq        results_fastqc     yeast_wg.4.bt2
BBduk1.sh                          Test2                            trimmed_data       yeast_wg.rev.1.bt2
D12_17539_ATCTCA_read1_part.fastq  bbmap                            yeast_ref_genome   yeast_wg.rev.2.bt2
D12_17539_ATCTCA_read2_part.fastq  bowtie2-2.5.1-mingw-x86_64       yeast_wg.1.bt2
FastQC                             bowtie2-2.5.1-mingw-x86_64.zip   yeast_wg.2.bt2
Iter1_read1_trimmed.fastq          fastqc_v0.12.1.zip               yeast_wg.3.bt2
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$
```

We can see these bt2 files also right here. We can simply say ls right and we will see multiple bt2 files that have been created here. So, you have yeast wg dot 1, w 2 wg dot 2, wg dot 3, etcetera, and you will also see some of the reverse in the indexes right. So, you have this wg reverse dot 1 right reverse dot 2, and this is the double indexing that we have talked about, right?



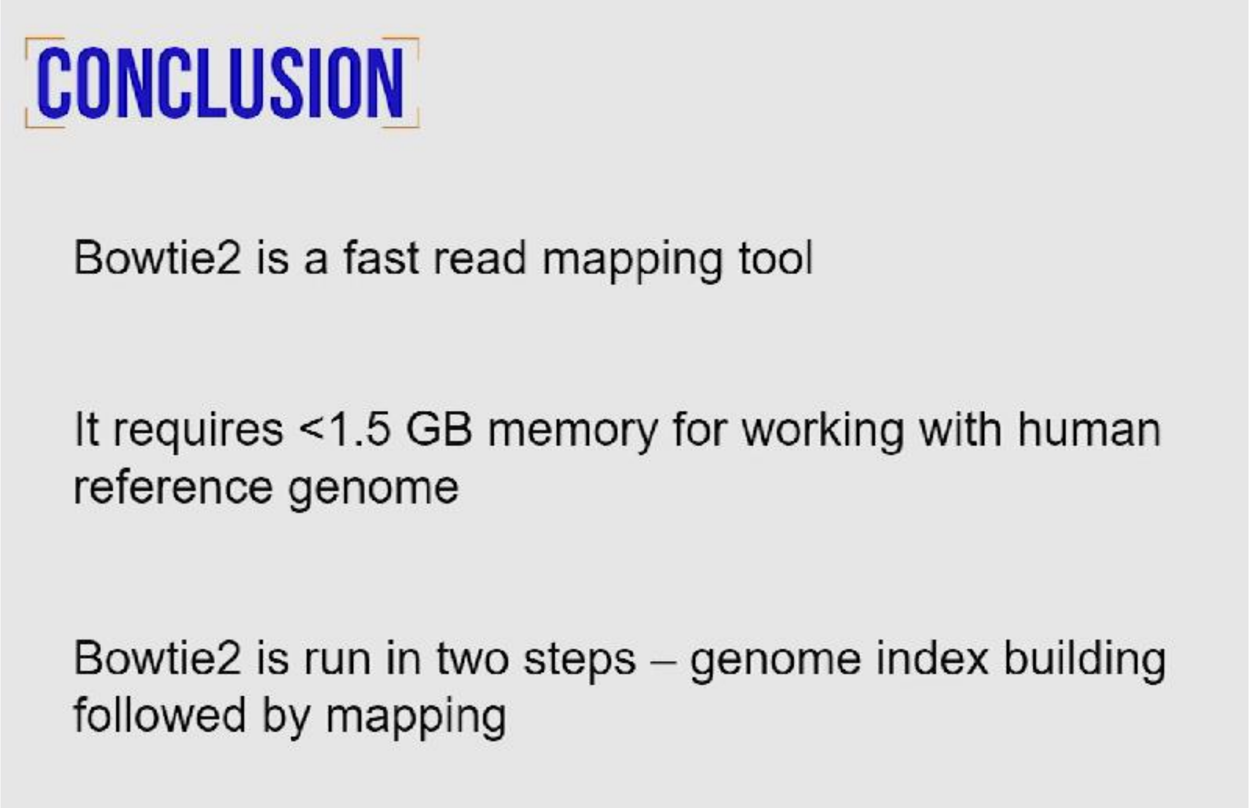
```
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$ ls *.bt2
yeast_wg.1.bt2  yeast_wg.2.bt2  yeast_wg.3.bt2  yeast_wg.4.bt2  yeast_wg.rev.1.bt2  yeast_wg.rev.2.bt2
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$
```

So, by the way, if you want to just see these BT2 files right in your system rather than checking all the files, you can simply say, "Start Dot BT2."So, what it means is that star means wildcard, right? So, it will match anything that it defines. So, it will just show all the files that end with this dot bt2 ok all files in bt2 format ok and simply you will find these 6 files created in your system ok. So, once you have created these files, So, this is the first step that is now complete, ok? We have built this index, and you can keep these bt2 files for as long as you want. As long as you are working with this genome you do not have to recreate these bt2 files unless the reference genome sequence is updated by a newer version, or so you do not need to work with or change these reference indexes anymore, ok? So, once you have created them, you can just keep them forever, and whenever you want to map reads against these sacraments, you can utilize these indexes. So, it is a good idea, right? Once you create it, you can keep it, and they do not take up too much space. So, you can also look at their space usage, and you can see they are not taking up a huge amount of space here in the system.



```
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$ ls -al *.bt2
-rwxrwxrwx 1 rdhar rdhar 8248782 May 16 11:58 yeast_wg.1.bt2
-rwxrwxrwx 1 rdhar rdhar 3039284 May 16 11:58 yeast_wg.2.bt2
-rwxrwxrwx 1 rdhar rdhar     161 May 16 11:58 yeast_wg.3.bt2
-rwxrwxrwx 1 rdhar rdhar 3039277 May 16 11:58 yeast_wg.4.bt2
-rwxrwxrwx 1 rdhar rdhar 8248782 May 16 11:58 yeast_wg.rev.1.bt2
-rwxrwxrwx 1 rdhar rdhar 3039284 May 16 11:58 yeast_wg.rev.2.bt2
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/Test1$ ls -al *.bt2
```

You can compare this right. So, if you can, you can compare this to the read sequence data, and then they are actually quite okay, right? So, going back to the presentation, So, we have now created the indexes right out of this reference sequence using this bt2 build command. The next step is actually the mapping process, the actual read mapping process, and we will discuss that in the next class because this is a very elaborate step. We have a lot of options, and some of these options are very useful for mapping.



CONCLUSION

Bowtie2 is a fast read mapping tool

It requires <1.5 GB memory for working with human reference genome

Bowtie2 is run in two steps – genome index building followed by mapping

So, here are the references for this class, and to summarize what we have seen, BT2 is a very fast-read mapping tool. So, this is something we have discussed, and it requires less than 1.5 g of memory for working with the human reference genome, and BT2 is run in two steps. So, we have the first genome index building step, and then you have the actual mapping part. So, in this class, we have actually done the hands-on, where we have actually looked at the genome index building part. Right, we have not looked into the mapping part, which we will be doing in the next class and what we have shown is that BT2 builds the function that actually builds these index files and you can store these index files once you have created them and you can utilize them for mapping reads of that organism in the future. Thank you.