Guest Lecture

Bodo Linz

02/18/20

Bacterial Genomics: From sequencing reads to multiple genome alignment

Guest Lecture
Bodo Linz
02/18/20

Today’s lecture

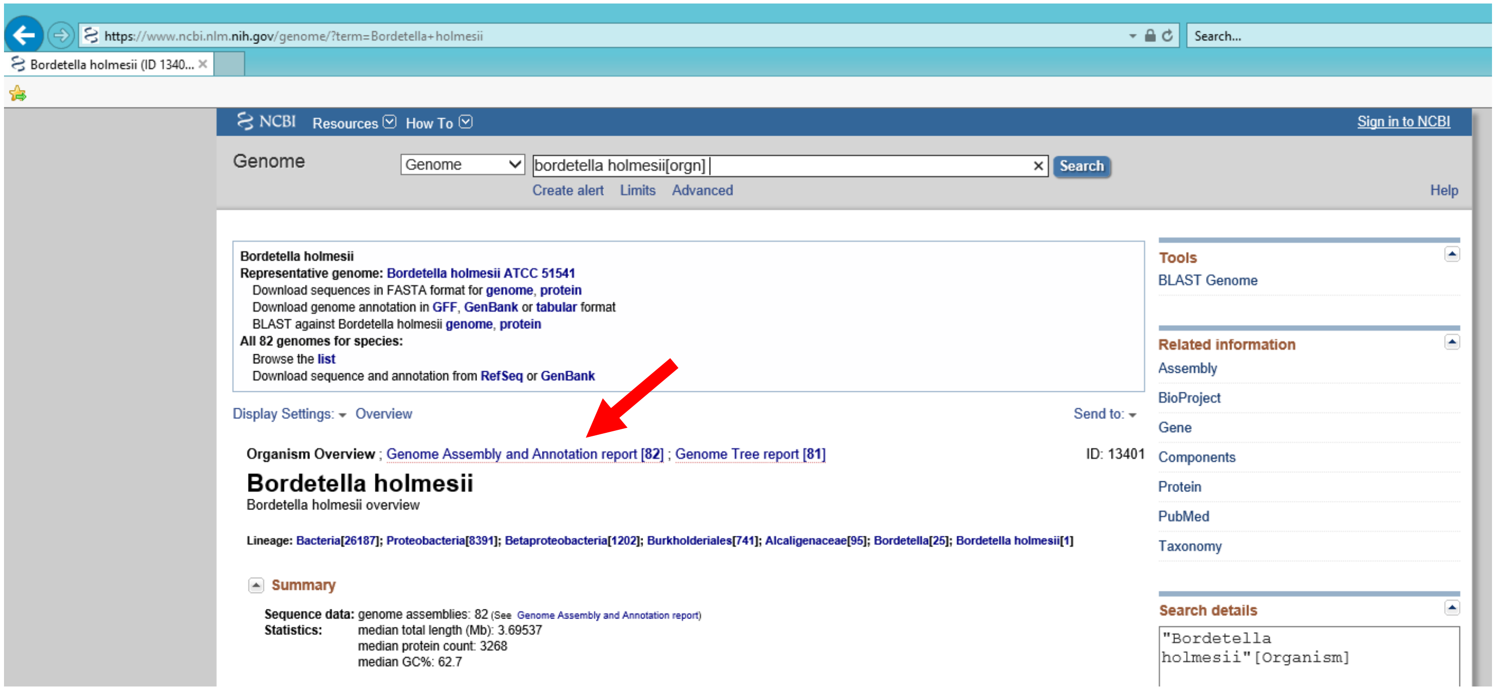
- Download complete genomes from NCBI
- Split complete genome into overlapping “reads”
- Download a Short Read Archive (SRA) from NCBI
- Join paired reads from the archive
- Align joined reads/split “reads” against a reference genome
- Call SNPs, generate consensus sequence
- Generate multiple genome alignment
- Make pairwise genome comparisons using `blastn` and `MSPcrunch`, visualize in Artemis Comparison Tool
- Extract and assemble a gene sequence from an SRA

Download a complete genome from NCBI

go to https://www.ncbi.nlm.nih.gov/genome/

type the species: *Bordetella holmesii*

Select: Genome Assembly and Annotation report



bordetella holmesii[orgn]

Bordetella holmesii

Representative genome: Bordetella holmesii ATCC 51541

Download sequences in FASTA format for genome, protein

Download genome annotation in GFF, GenBank or tabular format

BLAST against Bordetella holmesii genome, protein

All 82 genomes for species:

Browse the list

Download sequence and annotation from RefSeq or GenBank

Organism Overview ; Genome Assembly and Annotation report [82] ; Genome Tree report [81]

ID: 13401

Bordetella holmesii

Bordetella holmesii overview

Lineage: Bacteria[26187]; Proteobacteria[8391]; Betaproteobacteria[1202]; Burkholderiales[741]; Alcaligenaceae[95]; Bordetella[25]; Bordetella holmesii[1]

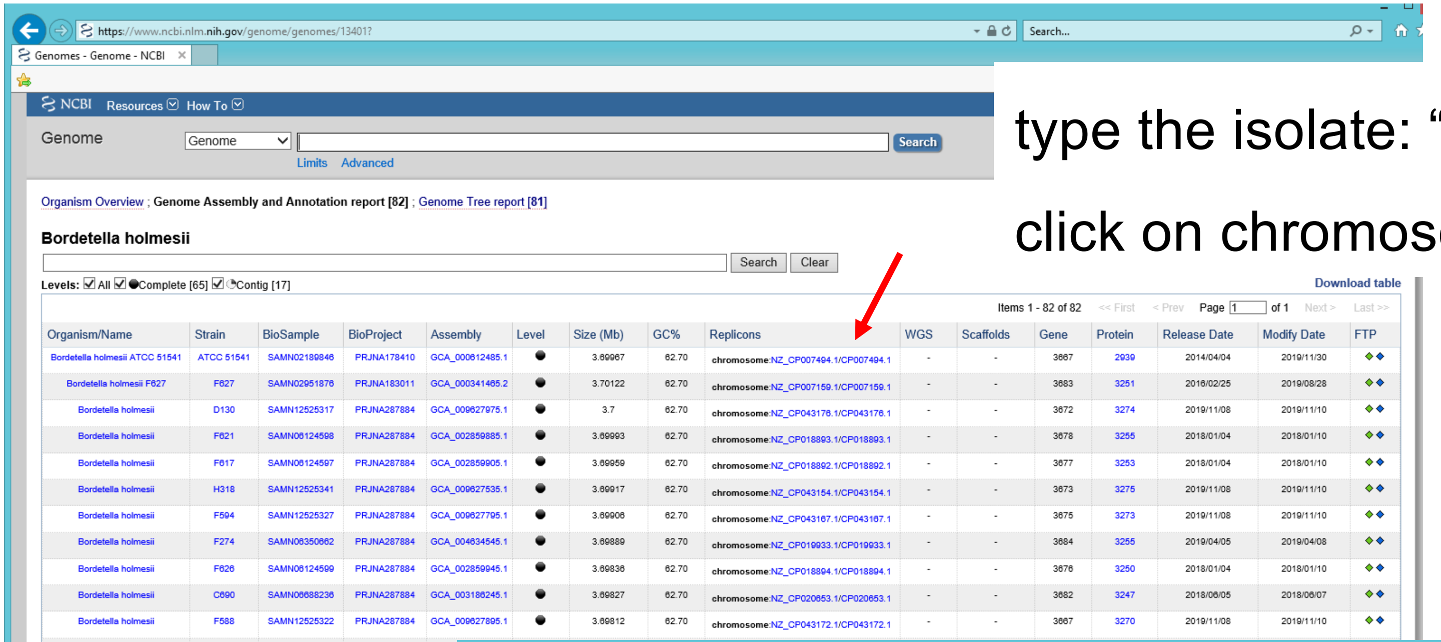
Summary

Sequence data: genome assemblies: 82 (See Genome Assembly and Annotation report)

Statistics: median total length (Mb): 3.69537
median protein count: 3268
median GC%: 62.7

"Bordetella holmesii"[Organism]

Download a complete genome from NCBI



https://www.ncbi.nlm.nih.gov/genome/genomes/13401?

Genomes - Genome - NCBI

Organism Overview ; Genome Assembly and Annotation report [82] ; Genome Tree report [81]

Bordetella holmesii

Levels: ☑ All ☑ ●Complete [65] ☑ ◔Contig [17]

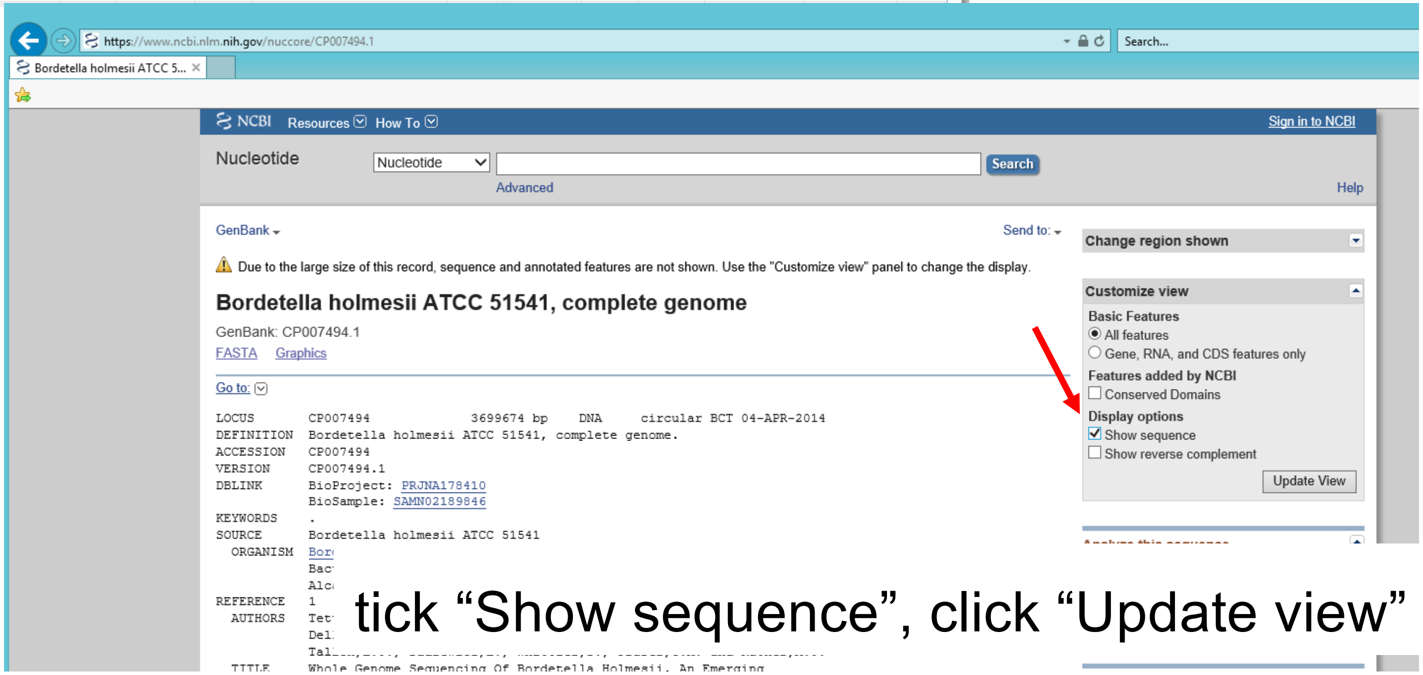
Download table

Items 1 - 82 of 82 << First < Prev Page 1 of 1 Next > Last >>

Organism/Name	Strain	BioSample	BioProject	Assembly	Level	Size (Mb)	GC%	Replicons	WGS	Scaffolds	Gene	Protein	Release Date	Modify Date	FTP
Bordetella holmesii ATCC 51541	ATCC 51541	SAMN02189846	PRJNA178410	GCA_000612485.1	●	3.69967	62.70	chromosome:NZ_CP007494.1/CP007494.1	-	-	3667	2939	2014/04/04	2019/11/30	◆◆
Bordetella holmesii F627	F627	SAMN02951876	PRJNA183011	GCA_000341465.2	●	3.70122	62.70	chromosome:NZ_CP007159.1/CP007159.1	-	-	3683	3251	2016/02/25	2019/08/28	◆◆
Bordetella holmesii	D130	SAMN12525317	PRJNA287884	GCA_009627975.1	●	3.7	62.70	chromosome:NZ_CP043176.1/CP043176.1	-	-	3672	3274	2019/11/08	2019/11/10	◆◆
Bordetella holmesii	F621	SAMN06124598	PRJNA287884	GCA_002859885.1	●	3.69993	62.70	chromosome:NZ_CP018893.1/CP018893.1	-	-	3678	3255	2018/01/04	2018/01/10	◆◆
Bordetella holmesii	F617	SAMN06124597	PRJNA287884	GCA_002859905.1	●	3.69959	62.70	chromosome:NZ_CP018892.1/CP018892.1	-	-	3677	3253	2018/01/04	2018/01/10	◆◆
Bordetella holmesii	H318	SAMN12525341	PRJNA287884	GCA_009627535.1	●	3.69917	62.70	chromosome:NZ_CP043154.1/CP043154.1	-	-	3673	3275	2019/11/08	2019/11/10	◆◆
Bordetella holmesii	F594	SAMN12525327	PRJNA287884	GCA_009627795.1	●	3.69906	62.70	chromosome:NZ_CP043167.1/CP043167.1	-	-	3675	3273	2019/11/08	2019/11/10	◆◆
Bordetella holmesii	F274	SAMN06350662	PRJNA287884	GCA_004634545.1	●	3.69889	62.70	chromosome:NZ_CP019933.1/CP019933.1	-	-	3684	3255	2019/04/05	2019/04/08	◆◆
Bordetella holmesii	F626	SAMN06124599	PRJNA287884	GCA_002859945.1	●	3.69836	62.70	chromosome:NZ_CP018894.1/CP018894.1	-	-	3676	3250	2018/01/04	2018/01/10	◆◆
Bordetella holmesii	C690	SAMN08688236	PRJNA287884	GCA_003186245.1	●	3.69827	62.70	chromosome:NZ_CP020653.1/CP020653.1	-	-	3682	3247	2018/06/05	2018/06/07	◆◆
Bordetella holmesii	F588	SAMN12525322	PRJNA287884	GCA_009627895.1	●	3.69812	62.70	chromosome:NZ_CP043172.1/CP043172.1	-	-	3667	3270	2019/11/08	2019/11/10	◆◆

type the isolate: “ATCC 51541”

click on chromosome in replicons



https://www.ncbi.nlm.nih.gov/nuccore/CP007494.1

Bordetella holmesii ATCC 5...

Due to the large size of this record, sequence and annotated features are not shown. Use the "Customize view" panel to change the display.

Bordetella holmesii ATCC 51541, complete genome

GenBank: CP007494.1

FASTA Graphics

Go to:

```
LOCUS       CP007494             3699674 bp    DNA     circular BCT 04-APR-2014
DEFINITION  Bordetella holmesii ATCC 51541, complete genome.
ACCESSION   CP007494
VERSION     CP007494.1
DBLINK      BioProject: PRJNA178410
            BioSample: SAMN02189846
KEYWORDS    .
SOURCE      Bordetella holmesii ATCC 51541
  ORGANISM  Bor
            Bac
            Alc
REFERENCE   1
  AUTHORS   Tet
            Del
            Tal
  TITLE     Whole Genome Sequencing Of Bordetella Holmesii, An Emerging
```

Customize view

Basic Features
● All features
○ Gene, RNA, and CDS features only

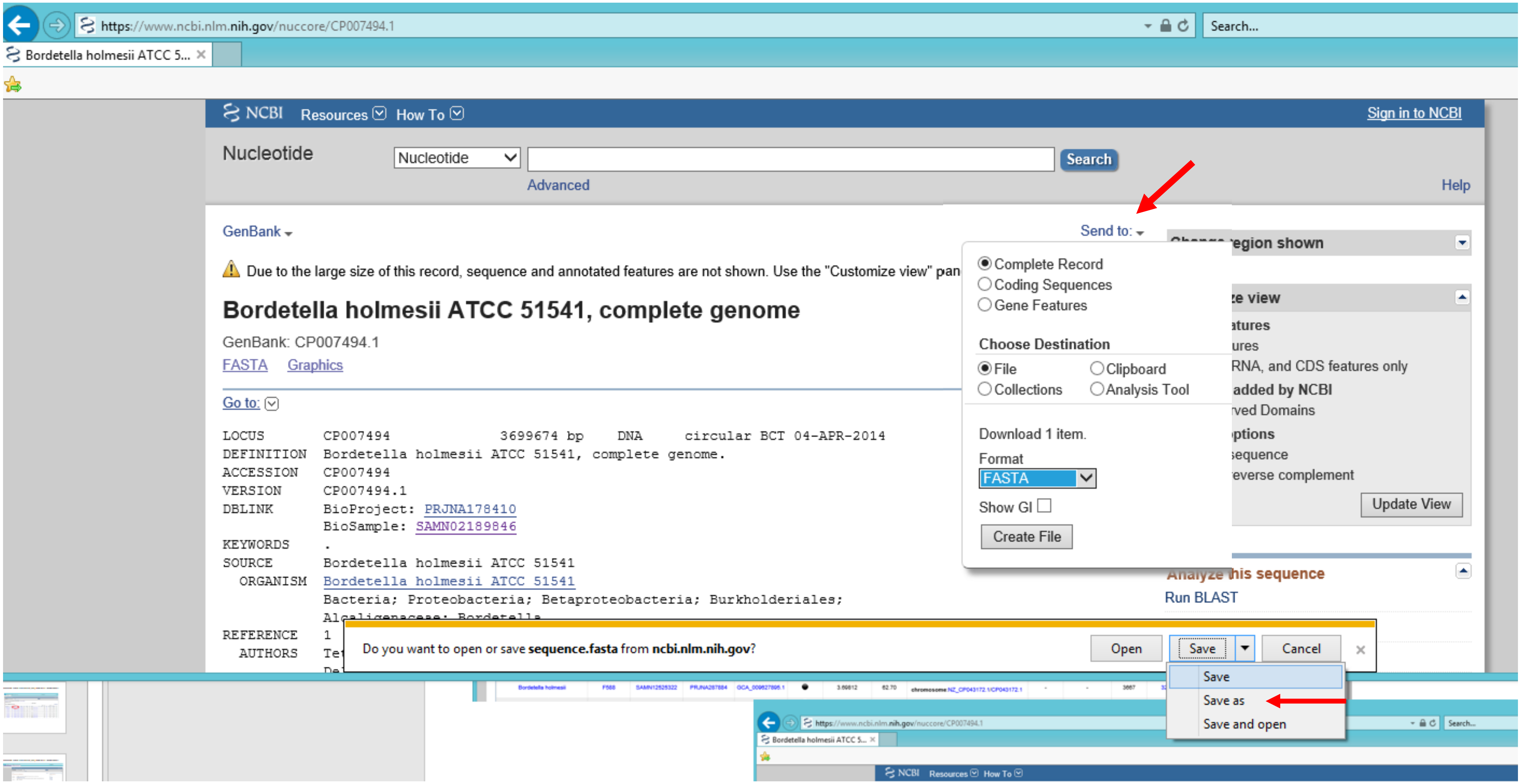
Features added by NCBI
☐ Conserved Domains

Display options
☑ Show sequence
☐ Show reverse complement

Update View

tick “Show sequence”, click “Update view”

Download a complete genome from NCBI



https://www.ncbi.nlm.nih.gov/nuccore/CP007494.1

Bordetella holmesii ATCC 5...

NCBI Resources How To — Sign in to NCBI

Nucleotide — Nucleotide — Search — Advanced — Help

GenBank

Send to:

⚠ Due to the large size of this record, sequence and annotated features are not shown. Use the "Customize view" pan

Bordetella holmesii ATCC 51541, complete genome

GenBank: CP007494.1

FASTA Graphics

Go to:

```
LOCUS       CP007494              3699674 bp    DNA     circular BCT 04-APR-2014
DEFINITION  Bordetella holmesii ATCC 51541, complete genome.
ACCESSION   CP007494
VERSION     CP007494.1
DBLINK      BioProject: PRJNA178410
            BioSample: SAMN02189846
KEYWORDS    .
SOURCE      Bordetella holmesii ATCC 51541
  ORGANISM  Bordetella holmesii ATCC 51541
            Bacteria; Proteobacteria; Betaproteobacteria; Burkholderiales;
            Alcaligenaceae; Bordetella.
REFERENCE   1
  AUTHORS   Te
            De
```

● Complete Record
○ Coding Sequences
○ Gene Features

Choose Destination

● File ○ Clipboard
○ Collections ○ Analysis Tool

Download 1 item.

Format: FASTA

Show GI □

Create File

Change region shown

Customize view: Features; Features; rRNA, and CDS features only; added by NCBI; Conserved Domains; Options; sequence; reverse complement

Update View

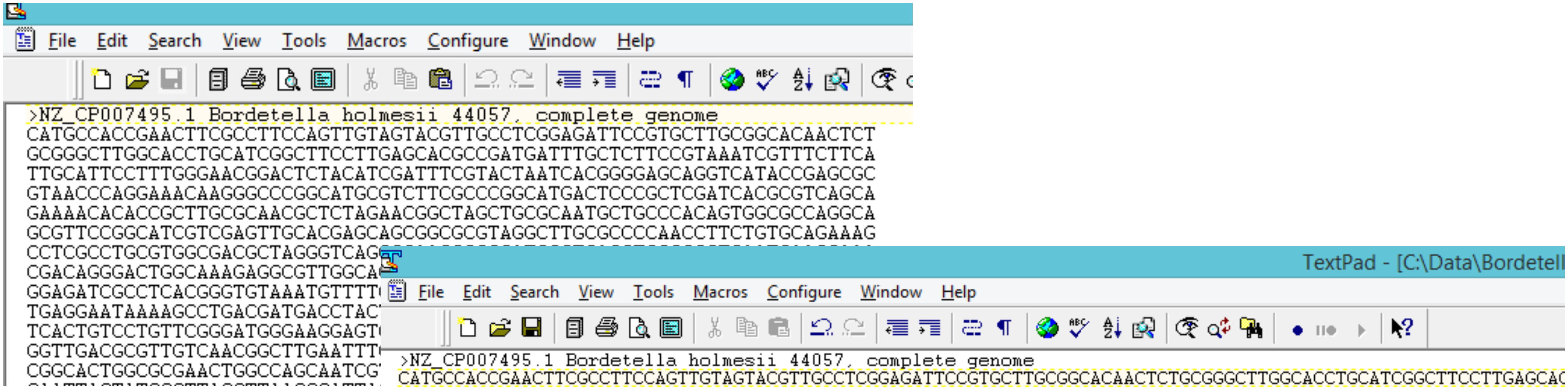
Analyze this sequence

Run BLAST

Do you want to open or save **sequence.fasta** from **ncbi.nlm.nih.gov**?

Open | Save | Cancel

Save
Save as
Save and open



```
>NZ_CP007495.1 Bordetella holmesii 44057, complete genome
CATGCCACCGAACTTCGCCTTCCAGTTGTAGTACGTTGCCTCGGAGATTCCGTGCTTGCGGCACAACTCT
GCGGGCTTGGCACCTGCATCGGCTTCCTTGAGCACGCCGATGATTTGCTCTTCCGTAAATCGTTTCTTCA
TTGCATTCCTTTGGGAACGGACTCTACATCGATTTCGTACTAATCACGGGGAGCAGGTCATACCGAGCGC
GTAACCCAGGAAACAAGGGCCCGGCATGCGTCTTCGCCCGGCATGACTCCCGCTCGATCACGCGTCAGCA
GAAAACACACCGCTTGCGCAACGCTCTAGAACGGCTAGCTGCGCAATGCTGCCCACAGTGGCGCCAGGCA
GCGTTCCGGCATCGTCGAGTTGCACGAGCAGCGGCGCGTAGGCTTGCGCCCCAACCTTCTGTGCAGAAAG
CCTCGCCTGCGTGGCGACGCTAGGGTCAG
CGACAGGGACTGGCAAAGAGGCGTTGGCA
GGAGATCGCCTCACGGGTGTAAATGTTTT
TGAGGAATAAAAGCCTGACGATGACCTAC
TCACTGTCCTGTTCGGGATGGGAAGGAGT
GGTTGACGCGTTGTCAACGGCTTGAATTT
CGGCACTGGCGCGAACTGGCCAGCAATCG
```

```
>NZ_CP007495.1 Bordetella holmesii 44057, complete genome
CATGCCACCGAACTTCGCCTTCCAGTTGTAGTACGTTGCCTCGGAGATTCCGTGCTTGCGGCACAACTCTGCGGGCTTGGCACCTGCATCGGCTTCCTTGAGCAC
```

# if closed genome in multiple line fasta format make single line fasta sequence

```
cat F029.fasta | awk
‘BEGIN{RS=”>”;FS=”\n”}NR>1{seq=seq””;for(i=2;i<=NF;i++)seq=seq””$i;
print ”>”$1”\n”seq}’ > F029g.fasta
```

```
# RS – Record Separator: end of record marker, default new line
        → new entry starts here
# FS – Field Separator: \n separates the fields
# for(i=2;i<=NF;i++)seq=seq””$i
# for all rows starting from 2, until the last row, seq is seq plus
the next seq
# print ”>”$1”\n”seq}’ – print >, fasta header, \n, built sequence
```

# to split genomes into overlapping 400 bp reads run script

```
split_genome_to_reads.sh
```

```bash
#!/bin/bash
# split_genome_to_reads
# author Bodo Linz
# split a genome into 400 bp long overlapping reads, 20 bp steps

file="F061g.fasta"
NAMEGENOME=${file%%".fasta"}

echo ""
echo "load input file $NAMEGENOME.fasta"
echo ""
echo "split genome into 400 bp fragments"
echo "-------------------------------------------"
cat ${NAMEGENOME}.fasta | awk 'NR > 1' | fold -w400 > fake4a

# select all rows except the first      awk 'NR > 1’
# split into chunks of 400 nucleotides         fold -w400

echo ""
echo "R: add read number a."
echo "-------------------------------------------"
# Run R in '--slave' mode to incorporate in bash script
R --slave -f /home/bodo/bin/add_sequence_read_number.R
```

```
#!usr/bin/R
# add_sequence_read_number.R
# delete al objects
rm(list = ls())

# load packages
 library(base)

# set the working dicrectory
setwd("~/bodo.2/bordetella/Bholmesii/align")

# load data in table format
data <- read.table("fake4a", header = FALSE, sep= "\t")

# row count in file data into file pos
pos <- seq.int(nrow(data))

data2 <- cbind(pos,data)  # combine files pos and data

write.table(data2, file = "fake4b", sep = "\t",
row.names = FALSE, col.names = FALSE)
```

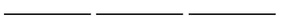
```
#  fake4a
#  ATGTCTGATTGACCGTAGCATTGTAG
#  TGAGTGCGTACCCGTACGTGACCATT
# fake4b
  $1   $2
  1    “ATGTCTGATTGACCGTAGCATTGTAG”
  2    “TGAGTGCGTACCVGTACGTGACCATT”
```

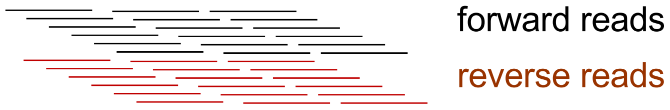
```
cat fake4b | awk -v FS="\t" -v OFS="" '{print ">read"$1"a",
"\n", $2}' | tr -d '"' > ${NAMEGENOME}_reads.fa
```

```
# tr -d '"'
# change " to nothing
```

```
# Let’s look at F061g_reads.fa
>read1a
ATGTCTGATTGACCGTAGCATTGTAG
>read2a
TGAGTGCGTACCCGTACGTGACCATT
```

```
# what we got   # what we want
```





```
# F061g_reads.fa     so far
>read1a
ATGTCTGATTGACCGTAGCATTGTAG
>read2a
TGAGTGCGTACCCGTACGTGACCATT
```

```
# add 20 A’s at beginning of sequence, then split into “reads”
cat ${NAMEGENOME}.fasta | awk 'NR > 1' | awk -v OFS="" '{print
"AAAAAAAAAAAAAAAAAAAA",$1}' | fold -w400 > fake4a

R --slave -f /home/bodo/bin/add_sequence_read_number.R

cat fake4b | awk -v FS="\t" -v OFS="" '{print ">read"$1“b",
"\n", $2}' | tr -d '"' >> ${NAMEGENOME}_reads.fa
```

```
# ">>" - append to this file
```

```
We got
>read1a
ATGTCTGATTGACCGTAGCATTGTAG
>read1b
AAAAAAAAAAATGTCTGATTGACCGT
```

now iterate with more A’s

```
# add 40 A’s at beginning of sequence, then split into “reads”
       read1c read2c read3c ……
# add 60 A’s at beginning of sequence, then split into “reads”
       read1d read2d read3d ……
# add 80 A’s at beginning of sequence, then split into “reads”
       read1e read2e read3e ……
# add 100 A’s at beginning of sequence, then split into “reads”
       read1f read2f read3f ……
……
# add 380 A’s at beginning of sequence, then split into “reads”
       read1t read2t read3t ……
```

We got overlapping forward reads, let’s get the reverse reads

```
echo ""
echo "reverse genome"
echo "----------------------------------------"
cat ${NAMEGENOME}.fasta | awk "NR > 1" | awk '{print $1}' >
temp.fas

cat temp.fas | tr "[ATGCatgcNn]" "[TACGtacgNn]" | rev | awk
'{print ">F061g-rev.fasta","\n",$1}' > ${NAMEGENOME}-rev.fasta
```

```
# Let’s walk through:
echo ""
echo "reverse genome"
echo "-----------------------------------------"
cat ${NAMEGENOME}.fasta | awk "NR > 1" | awk '{print $1}' >
temp.fas
# awk "NR > 1" - select all rows except row 1
# awk '{print $1}' - print what we got

cat temp.fas | tr "[ATGCatgcNn]" "[TACGtacgNn]" | rev | awk
'{print ">F029g-rev.fasta","\n",$1}' > ${NAMEGENOME}-rev.fasta
# tr "[ATGCatgcNn]" "[TACGtacgNn]" - change A to T, T to A,
  C to G, G to C, a to t, t to a, etc.
# rev - reverse resulting sequence
# awk '{print ">F029g-rev.fasta","\n",$1}’
# write header ">F029g-rev.fasta", then new line ("\n")
  , then the new reverse sequence
```

```
# Repeat with reverse genome and add reads to previous file
echo "split reverse genome into 400 bp fragments"
echo "-------------------------------------------"
cat ${NAMEGENOME}-rev.fasta | awk 'NR > 1' | fold -w400 > fake4a

cat fake4b | awk -v FS="\t" -v OFS="" '{print ">read"$1"reva",
"\n", $2}' | tr -d '"' >> ${NAMEGENOME}_reads.fa
```

different suffix

```
# add 20 A’s at beginning of rev sequence, split into “reads”
cat ${NAMEGENOME}-rev.fasta | awk 'NR > 1' | awk -v OFS=""
'{print "AAAAAAAAAAAAAAAAAAAA",$1}' | fold -w400 > fake4a

R --slave -f /home/bodo/bin/add_sequence_read_number.R

cat fake4b | awk -v FS="\t" -v OFS="" '{print ">read"$1“revb",
"\n", $2}' | tr -d '"' >> ${NAMEGENOME}_reads.fa
```

iterate with more A’s

→ We got the reads file from a complete genome.

Download a short read archive (SRA) from NCBI

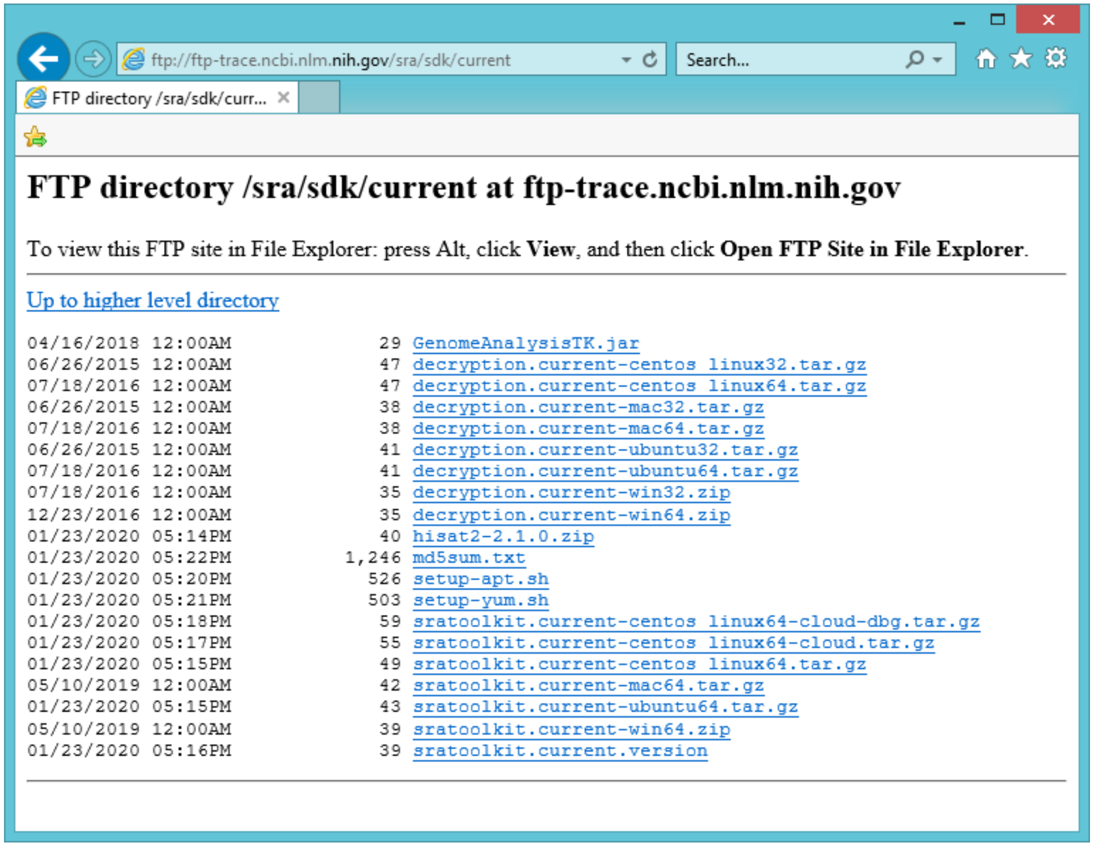
The only option: use the `sratoolkit` from NCBI

- to download sratoolkit, type:

```
wget ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sdk/current/sratoolkit.current-centos_linux64.tar.gz
```

# or wherever the program is currently located at the ncbi website

# still there!


ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sdk/current
Search...
FTP directory /sra/sdk/curr...

FTP directory /sra/sdk/current at ftp-trace.ncbi.nlm.nih.gov

To view this FTP site in File Explorer: press Alt, click **View**, and then click **Open FTP Site in File Explorer**.

Up to higher level directory

```
04/16/2018 12:00AM          29 GenomeAnalysisTK.jar
06/26/2015 12:00AM          47 decryption.current-centos_linux32.tar.gz
07/18/2016 12:00AM          47 decryption.current-centos_linux64.tar.gz
06/26/2015 12:00AM          38 decryption.current-mac32.tar.gz
07/18/2016 12:00AM          38 decryption.current-mac64.tar.gz
06/26/2015 12:00AM          41 decryption.current-ubuntu32.tar.gz
07/18/2016 12:00AM          41 decryption.current-ubuntu64.tar.gz
07/18/2016 12:00AM          35 decryption.current-win32.zip
12/23/2016 12:00AM          35 decryption.current-win64.zip
01/23/2020 05:14PM          40 hisat2-2.1.0.zip
01/23/2020 05:22PM       1,246 md5sum.txt
01/23/2020 05:20PM         526 setup-apt.sh
01/23/2020 05:21PM         503 setup-yum.sh
01/23/2020 05:18PM          59 sratoolkit.current-centos_linux64-cloud-dbg.tar.gz
01/23/2020 05:17PM          55 sratoolkit.current-centos_linux64-cloud.tar.gz
01/23/2020 05:15PM          49 sratoolkit.current-centos_linux64.tar.gz
05/10/2019 12:00AM          42 sratoolkit.current-mac64.tar.gz
01/23/2020 05:15PM          43 sratoolkit.current-ubuntu64.tar.gz
05/10/2019 12:00AM          39 sratoolkit.current-win64.zip
01/23/2020 05:16PM          39 sratoolkit.current.version
```

Download a short read archive (SRA) from NCBI

The only option: use the `sratoolkit` from NCBI

- to download sratoolkit, type:

```
wget ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sdk/current/sratoolkit.current-centos_linux64.tar.gz
```

# or wherever the program is currently located at the ncbi website

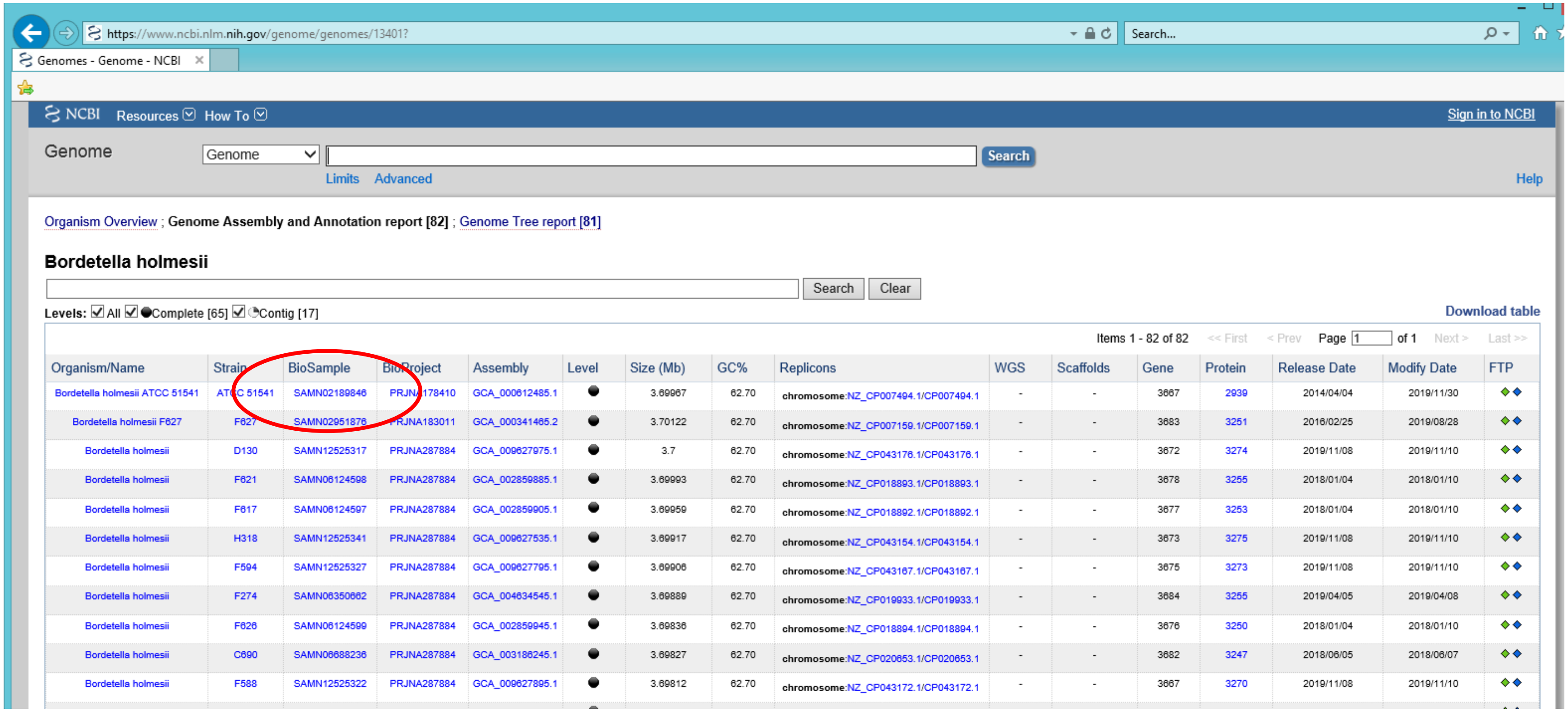
- to unpack the toolkit, type:

```
tar -xzf sratoolkit.current-centos_linux64.tar.gz
```

- location of fastq-dump and other commands:

```
~/[user_name]/sra-toolkit/bin/fastq-dump
```

Download a short read archive (SRA) from NCBI
Where do you find the archive?



Organism Overview ; Genome Assembly and Annotation report [82] ; Genome Tree report [81]

Bordetella holmesii

Levels: ☑ All ☑ ●Complete [65] ☑ ◔Contig [17]

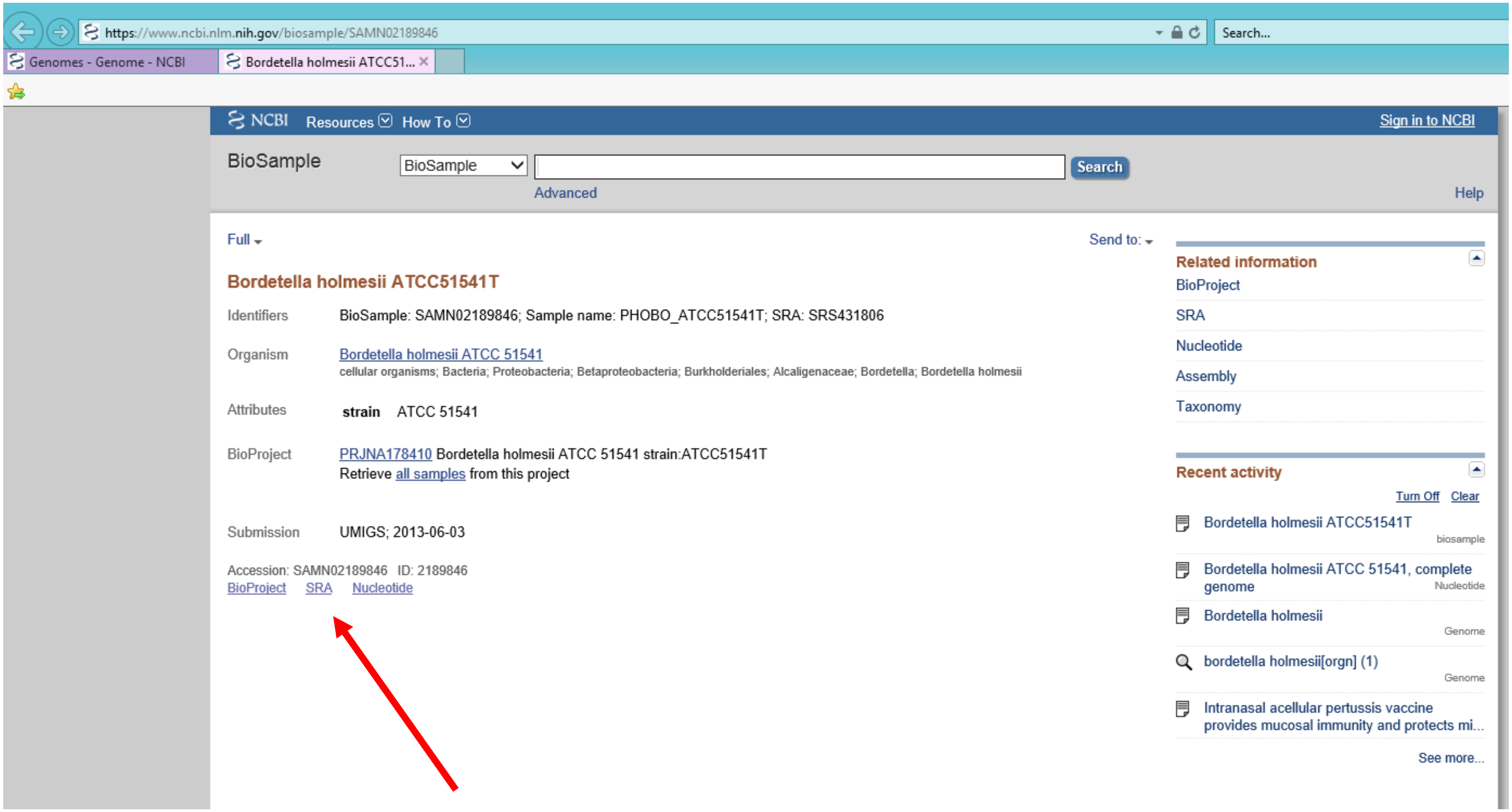
Download table

Items 1 - 82 of 82 << First < Prev Page 1 of 1 Next > Last >>

Organism/Name	Strain	BioSample	BioProject	Assembly	Level	Size (Mb)	GC%	Replicons	WGS	Scaffolds	Gene	Protein	Release Date	Modify Date	FTP
Bordetella holmesii ATCC 51541	ATCC 51541	SAMN02189846	PRJNA178410	GCA_000612485.1	●	3.69967	62.70	chromosome:NZ_CP007494.1/CP007494.1	-	-	3667	2939	2014/04/04	2019/11/30	◆◆
Bordetella holmesii F627	F627	SAMN02951876	PRJNA183011	GCA_000341465.2	●	3.70122	62.70	chromosome:NZ_CP007159.1/CP007159.1	-	-	3683	3251	2016/02/25	2019/08/28	◆◆
Bordetella holmesii	D130	SAMN12525317	PRJNA287884	GCA_009627975.1	●	3.7	62.70	chromosome:NZ_CP043176.1/CP043176.1	-	-	3672	3274	2019/11/08	2019/11/10	◆◆
Bordetella holmesii	F621	SAMN06124598	PRJNA287884	GCA_002859885.1	●	3.69993	62.70	chromosome:NZ_CP018893.1/CP018893.1	-	-	3678	3255	2018/01/04	2018/01/10	◆◆
Bordetella holmesii	F617	SAMN06124597	PRJNA287884	GCA_002859905.1	●	3.69959	62.70	chromosome:NZ_CP018892.1/CP018892.1	-	-	3677	3253	2018/01/04	2018/01/10	◆◆
Bordetella holmesii	H318	SAMN12525341	PRJNA287884	GCA_009627535.1	●	3.69917	62.70	chromosome:NZ_CP043154.1/CP043154.1	-	-	3673	3275	2019/11/08	2019/11/10	◆◆
Bordetella holmesii	F594	SAMN12525327	PRJNA287884	GCA_009627795.1	●	3.69906	62.70	chromosome:NZ_CP043167.1/CP043167.1	-	-	3675	3273	2019/11/08	2019/11/10	◆◆
Bordetella holmesii	F274	SAMN06350662	PRJNA287884	GCA_004634545.1	●	3.69889	62.70	chromosome:NZ_CP019933.1/CP019933.1	-	-	3684	3255	2019/04/05	2019/04/08	◆◆
Bordetella holmesii	F626	SAMN06124599	PRJNA287884	GCA_002859945.1	●	3.69836	62.70	chromosome:NZ_CP018894.1/CP018894.1	-	-	3676	3250	2018/01/04	2018/01/10	◆◆
Bordetella holmesii	C690	SAMN06688236	PRJNA287884	GCA_003186245.1	●	3.69827	62.70	chromosome:NZ_CP020653.1/CP020653.1	-	-	3682	3247	2018/06/05	2018/06/07	◆◆
Bordetella holmesii	F588	SAMN12525322	PRJNA287884	GCA_009627895.1	●	3.69812	62.70	chromosome:NZ_CP043172.1/CP043172.1	-	-	3667	3270	2019/11/08	2019/11/10	◆◆

click on the BioSample, e.g. SAMN02189846

Download a short read archive (SRA) from NCBI – from where?



Bordetella holmesii ATCC51541T

Identifiers	BioSample: SAMN02189846; Sample name: PHOBO_ATCC51541T; SRA: SRS431806
Organism	Bordetella holmesii ATCC 51541 cellular organisms; Bacteria; Proteobacteria; Betaproteobacteria; Burkholderiales; Alcaligenaceae; Bordetella; Bordetella holmesii
Attributes	**strain** ATCC 51541
BioProject	PRJNA178410 Bordetella holmesii ATCC 51541 strain:ATCC51541T Retrieve all samples from this project
Submission	UMIGS; 2013-06-03

Accession: SAMN02189846 ID: 2189846

BioProject SRA Nucleotide

Related information

- BioProject
- SRA
- Nucleotide
- Assembly
- Taxonomy

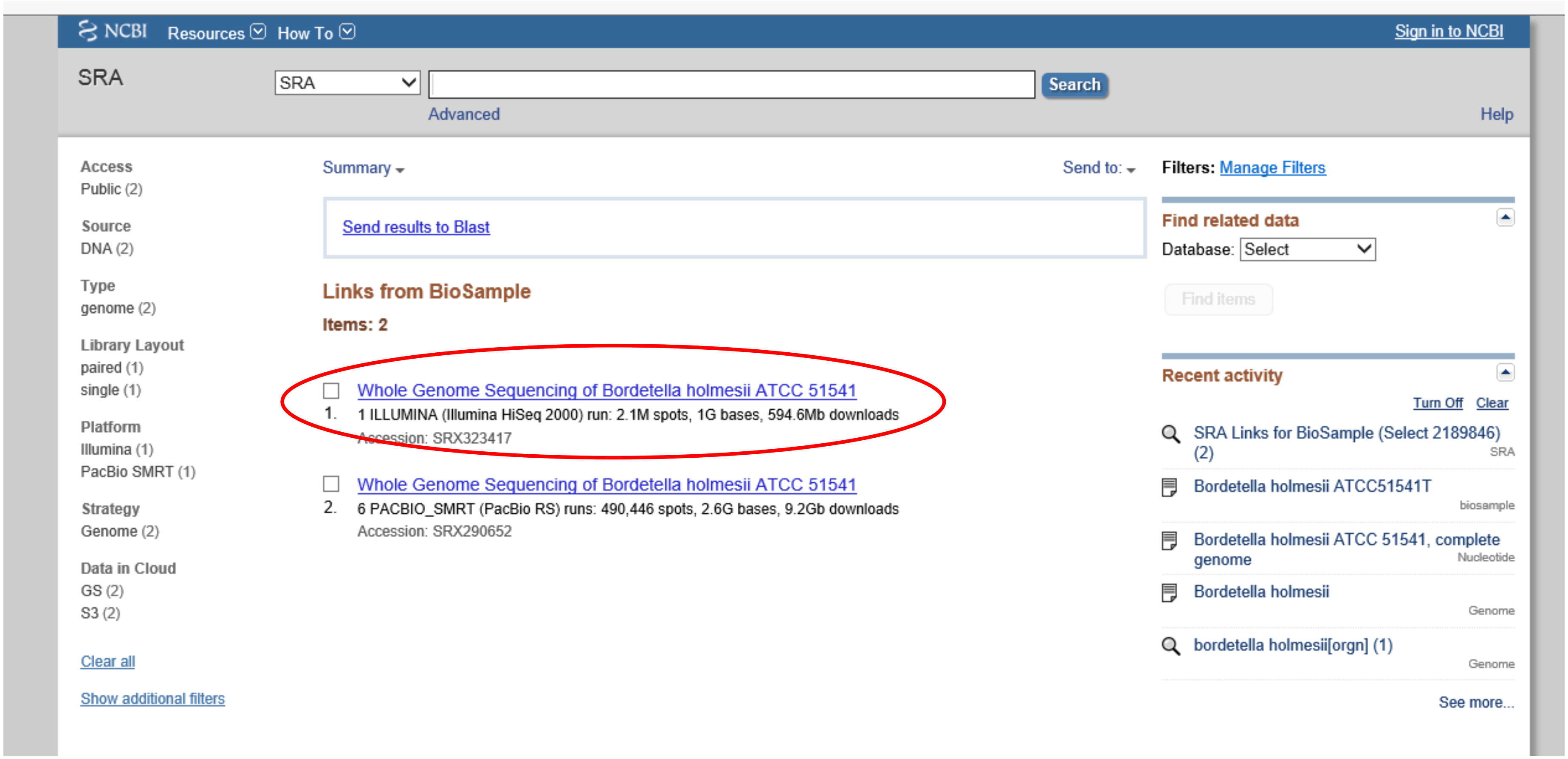
Recent activity

Turn Off Clear

- Bordetella holmesii ATCC51541T — biosample
- Bordetella holmesii ATCC 51541, complete genome — Nucleotide
- Bordetella holmesii — Genome
- bordetella holmesii[orgn] (1) — Genome
- Intranasal acellular pertussis vaccine provides mucosal immunity and protects mi...

See more...

Download a short read archive (SRA) from NCBI – from where?



Access
Public (2)

Source
DNA (2)

Type
genome (2)

Library Layout
paired (1)
single (1)

Platform
Illumina (1)
PacBio SMRT (1)

Strategy
Genome (2)

Data in Cloud
GS (2)
S3 (2)

Clear all

Show additional filters

Send results to Blast

Links from BioSample

Items: 2

☐ 1. Whole Genome Sequencing of Bordetella holmesii ATCC 51541
1 ILLUMINA (Illumina HiSeq 2000) run: 2.1M spots, 1G bases, 594.6Mb downloads
Accession: SRX323417

☐ 2. Whole Genome Sequencing of Bordetella holmesii ATCC 51541
6 PACBIO_SMRT (PacBio RS) runs: 490,446 spots, 2.6G bases, 9.2Gb downloads
Accession: SRX290652

SRA Links for BioSample (Select 2189846) (2) — SRA

Bordetella holmesii ATCC51541T — biosample

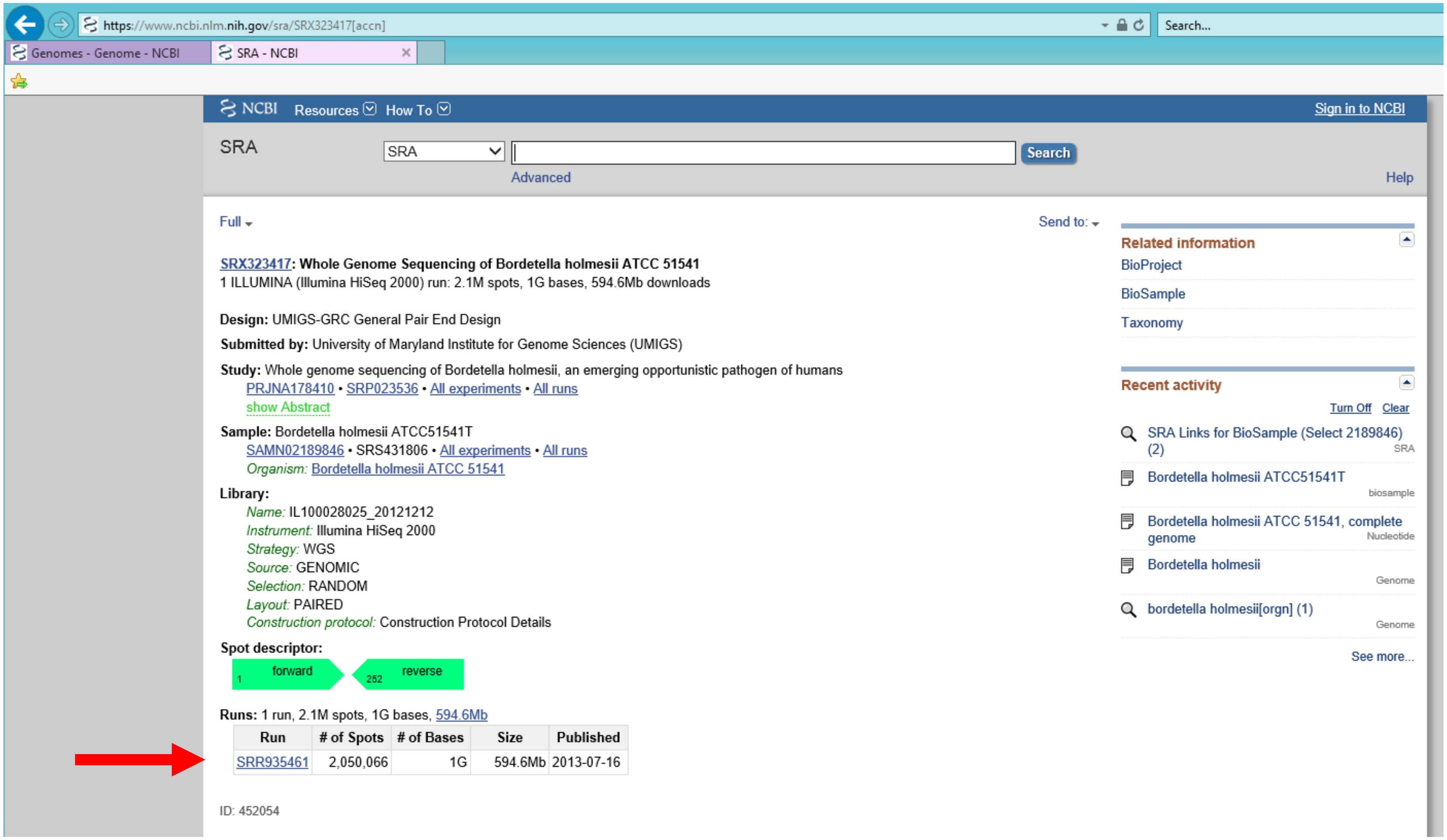
Bordetella holmesii ATCC 51541, complete genome — Nucleotide

Bordetella holmesii — Genome

bordetella holmesii[orgn] (1) — Genome

See more...

Download a short read archive (SRA) from NCBI – from where?



SRX323417: Whole Genome Sequencing of Bordetella holmesii ATCC 51541
1 ILLUMINA (Illumina HiSeq 2000) run: 2.1M spots, 1G bases, 594.6Mb downloads

Design: UMIGS-GRC General Pair End Design

Submitted by: University of Maryland Institute for Genome Sciences (UMIGS)

Study: Whole genome sequencing of Bordetella holmesii, an emerging opportunistic pathogen of humans
PRJNA178410 • SRP023536 • All experiments • All runs
show Abstract

Sample: Bordetella holmesii ATCC51541T
SAMN02189846 • SRS431806 • All experiments • All runs
Organism: Bordetella holmesii ATCC 51541

Library:
Name: IL100028025_20121212
Instrument: Illumina HiSeq 2000
Strategy: WGS
Source: GENOMIC
Selection: RANDOM
Layout: PAIRED
Construction protocol: Construction Protocol Details

Spot descriptor:

Runs: 1 run, 2.1M spots, 1G bases, 594.6Mb

Run	# of Spots	# of Bases	Size	Published
SRR935461	2,050,066	1G	594.6Mb	2013-07-16

ID: 452054

Related information
BioProject
BioSample
Taxonomy

Recent activity
Turn Off Clear
SRA Links for BioSample (Select 2189846) (2) SRA
Bordetella holmesii ATCC51541T biosample
Bordetella holmesii ATCC 51541, complete genome Nucleotide
Bordetella holmesii Genome
bordetella holmesii[orgn] (1) Genome
See more...

This is the SRA for this genome: SRR935461

Download a short read archive (SRA) from NCBI

```
~/[user_name]/sra-toolkit/bin/fastq-dump
```

- go to the /bin directory

- Since the documentation is pretty minimal, here is the command line to type:

```
./fastq-dump --outdir ~/bodo.2/Bholmesii/fastq
--skip-technical  --readids --dumpbase --split-files --clip
SRR_ID
```

# ./fastq-dump – start the command fastq-dump in the current directory "./"
# --outdir – specify the output directory, here `~/bodo.2/Bholmesii/fastq`
# --skip-technical – dump only biological reads, skip info such as:

```
Application Read Forward -> Technical Read Forward <- Application Read
Reverse - Technical Read Reverse.
```

# --readids – append the real read-ID after spot ID 'accession.spot.readid'
# --dumpbase – formats sequence using base space (default other than SOLiD)
# --split-files – Save forward and reverse reads into separate files. Files will receive suffix corresponding to read number.
# --clip SRR_ID – change the SRR_ID to whatever the ID is, e.g. SRR942665

e.g.

```
./fastq-dump --outdir ~/bodo.2/Bholmesii/fastq
--skip-technical  --readids --dumpbase --split-files --clip
SRR942665
```

Downloaded paired reads: SRR942665_1.fastq and SRR942665_2.fastq

Let's have a look at the `FASTQ` format, it's in 4 lines:

```
@SEQ_ID
NUCLEOTIDE_SEQUENCE
+ (sometimes with seqID again)
QUALITY_SCORES_FOR_ALL_NUCLEOTIDES
```

e.g.

```
@SRR942665.3.1 SOLEXA4:47:D1RLFACXX:6:1101:2945:2102 length=101
TTCTGTGGAAAGGTGAGGTCATCGACGTCGGCGTGCGCCTCGGCGCGCAGGCCCACTTTGTCCAGGC
AGTCCCAGGCCAGGGCGCGCATCGGCCAGGCC
+
CCCFFDFFHHFHHIGGIIAEEHHJHGIJJJJIG@AGGIHGIGEADDDDDBDDBDBBBDDDDCDCCCBBBC
DDDDC@BDDBBDDBBBBBBB@B<@DBDABBD
```

quality value characters in left-to-right increasing order of quality (ASCII):

```
#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
```

Join the paired reads:
SRR942665_1.fastq and SRR942665_2.fastq using **FLASH**

Magoc and Salzberg (2011). FLASH: fast length adjustment of short reads to improve genome assemblies. *Bioinformatics* **27**: 2957-2963.

- very accurate and fast tool to merge overlapping paired-end reads
- Merged read pairs result in unpaired longer reads
- Longer reads are preferred in genome assembly and analysis processes

```
flash <mates1.fastq> <mates2.fastq> [-m minOverlap] [-M
maxOverlap] [-x mismatchRatio]
```

```
flash SRR942665_1.fastq SRR942665_2.fastq –m 10 –M 200 –x 0.1
```

You get 5 files:

```
out.extendedFrags.fastq
out.notCombined_1.fastq
out.notCombined_2.fastq
out.hist
out.histogram
```

Joined paired reads are in: `out.extendedFrags.fastq`

rename:

```
mv out.extendedFrags.fastq Bhz5132_SRR942665_joined.fastq
```

`strain archiveID`

if wanted

rename: `mv out.notCombined_1.fastq Bhz5132_SRR942665_nc1.fastq`

rename: `mv out.notCombined_2.fastq Bhz5132_SRR942665_nc2.fastq`

Congratulations,
you got the joined reads from a Short Read Archive!

We got the joined reads from a Short Read Archive in fastq format.

Problem: We got the reads file from a complete genome in fasta format `F061_reads.fa`.

```
# change genome reads.fa to genome reads.fastq
# run perl script fasta_to_fastq.pl
# we have multiple genomes and run the script in a loop
files=$(ls *_reads.fa)
for file in $files; do name={file%%".fa"}; perl
~/bin/fasta_to_fastq.pl ${name}.fa > ${name}.fastq | echo "done
with $name"; done

# files=$(ls *_reads.fa) - create a list that contains all files
ending at _reads.fa
# for file in $files; do name={file%%".fa"}; - for all files in
this list, use the file name without ".fa"
# perl ~/bin/fasta_to_fastq.pl ${name}.fa > ${name}.fastq
for all files run perl script fasta_to_fastq.pl with input file
${name}.fa, save as output file ${name}.fastq
```

```
#Copyright (c) 2010 LUQMAN HAKIM BIN ABDUL HADI
#!/usr/bin/perl
use strict;
my $file = $ARGV[0];
open FILE, $file;
my ($header, $sequence, $sequence_length, $sequence_quality);
while(<FILE>) {
        chomp $_;
        if ($_ =~ /^>(.+)/) {
                if($header ne "") {
                        print "\@".$header."\n";
                        print $sequence."\n";
                        print "+"."\n";
                        print $sequence_quality."\n";
                }
                $header = $1;
                    $sequence = "";
                    $sequence_length = "";
                    $sequence_quality = "";
        }
          else {
                    $sequence .= $_;
                    $sequence_length = length($_);
                    for(my $i=0; $i<$sequence_length; $i++) {$sequence_quality .= "I"}
          }
}
close FILE;
print "\@".$header."\n";
print $sequence."\n";
print "+"."\n";
print $sequence_quality."\n";
```

We will

```
# align multiple genomes and run all steps in a loop
# align *.fastq files from SRA or from split genomes
  to the reference genome using bowtie2
# change .sam file to .bam file using samtools view
# sort bam file using samtools sort
# call variants using bcftools mpileup / bcftools call
# remove low quality variants by setting a threshold
# remove indels, check for potential problems
# zip the variant file using bgzip
# index the variant files using bcftools index
# generate consensus sequence using bcftools consensus
# change header and make fasta sequence one line
# make multiple alignment using cat
```

BOWTIE2

# download from
http://sourceforge.net/projects/bowtie-bio/files/bowtie2/2.2.3/bowtie2-2.2.3-linux-x86_64.zip

# unzip

```
unzip bowtie2-2.2.3-linux-x86_64.zip
```

# copy unzipped executables into $PATH (e.g. `~/bin`)

```
cd bowtie2-2.2.3
cp bowtie* ~/bin/
```

# generate bowtie2 index files of the reference sequence(s)
# bowtie2-build -f <reference> <reference-index>   (-f is fasta format, default fastq (-q)
Build reference genome database: `bowtie2-build -f ref.fas ref`

Install `samtools`, `bcftools` and `htslib`
In root:
Download current releases from www.htslib.org/download:
samtools-1.9   bcftools-1.9   htslib-1.9 into /home/Downloads
extract each

```
# download and install ncurses for samtools
yum install ncurses-devel

# Building and installing samtools
cd samtools-1.9
./configure --prefix=/home/bodo/bin          # optional to define directory
make
make install

# Building and installing bcftools
cd bcftools-1.9
./configure --prefix=/home/bodo/bin          # optional to define directory
make
make install

# Building and installing htslib
cd htslib-1.9
./configure --prefix=/home/bodo/bin          # optional to define directory
make
make install
```

# align *.fastq files from SRA or from split genomes to the reference genome using bowtie2

Is the reference genome database built? `bowtie2-build -f ref.fas ref`

# syntax

bowtie2 -x <db> -1 <mate1> -2 <mate2> -U <unpaired> -S <sam output>

```
files=$(ls *_reads.fastq)
```

```
for file in $files; do name=${file%%”_reads.fastq”}; bowtie2 -p 6 -k 2 -x ref -U ${name}_reads.fastq -S ${name}.sam | echo “done with $name”; done
```

```
# bowtie2 -p 6          if your computer has multiple processors use -p option
# -k 2                  with -k 2, bowtie2 searches for at most 2 distinct alignments
```

# change .sam file to .bam file

```
files=$(ls *.sam)
```

```
for file in $files; do name={file%%”.sam”}; samtools view -S -b ${name}.sam > ${name}.bam | echo “done with $name“; done
```

# sort bam file

```
files=$(ls *.bam)
```

```
for file in $files; do name={file%%”.bam”}; samtools sort
${name}.bam -o ${name}.sorted.bam | echo “done with $name”;
done
```

# call variants from a sorted bam file (<u>important:</u> use the same reference file as in bowtie2)

```
files=$(ls *.sorted.bam)
for file in $files; do name=${file%%".sorted.bam"}; bcftools
mpileup -f ref.fa ${name}.sorted.bam | bcftools call -mv -o
${name}.call.vcf | echo “done with $name”; done
```

# remove indels, remove low quality variants by setting a threshold

```
files=$(ls *.call.vcf)
for file in $files; do name=${file%%".call.vcf"}; cat
${name}.call.vcf | grep -v "INDEL“ | bcftools view -i
‘%QUAL>=80’ > ${name}.calls.vcf | echo “running $name”; done
# grep -v "INDEL“ -  unselect INDELs (optional, if you want
SNPs only, otherwise do not unselect)

# bcftools view -i ‘%QUAL>=80’ -  set quality threshold of 80
```

# remove indels – F061.call.vcf

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO
CP007494	12370	.	TCCCCC	TCCCCCC	124	.	INDEL;IDV=21;IMF=0.954545;DP=22;VDB=0.00….
CP007494	12384	.	GCC	GCCC	161	.	INDEL;IDV=23;IMF=1;DP=23;VDB=0.00131074;S…
CP007494	13477	.	GCCC	GCCCC	171	.	INDEL;IDV=39;IMF=0.975;DP=40;VDB=0.734156;….
CP007494	18817	.	C	A	177	.	DP=40;VDB=0.706575;SGB=-0.693145;MQSB=1;….
CP007494	19713	.	TGGGG	TGGGGG	177	.	INDEL;IDV=39;IMF=0.975;DP=40;VDB=0.74084;S
CP007494	19862	.	GCCCCC	GCCCCCC	173	.	INDEL;IDV=39;IMF=0.975;DP=40;VDB=0.730783;….
CP007494	20286	.	AGGGGG	AGGGGGG	177	.	INDEL;IDV=40;IMF=1;DP=40;VDB=0.727385;SGB=...
CP007494	23192	.	T	C	177	.	DP=40;VDB=0.688692;SGB=-0.693145;MQSB=1;M..
CP007494	23198	.	A	G	177	.	DP=40;VDB=0.699478;SGB=-0.693145;MQSB=1;M…
CP007494	23806	.	GCCC	GCCCC	159	.	INDEL;IDV=40;IMF=1;DP=40;VDB=0.753943;SGB…
CP007494	23826	.	CGGGGGG	CGGGGGGG	119	.	INDEL;IDV=39;IMF=0.975;DP=40;VDB=0.753946;S…
CP007494	26776	.	GCCCCC	GCCCCCC	177	.	INDEL;IDV=39;IMF=0.975;DP=40;VDB=0.699478;S…
CP007494	28257	.	CGGGGG	CGGGGGG	173	.	INDEL;IDV=39;IMF=0.975;DP=40;VDB=0.730783;S…
CP007494	28910	.	GCCCCC	GCCCCCC	171	.	INDEL;IDV=39;IMF=0.975;DP=40;VDB=0.74084;S…
CP007494	36469	.	A	G	175	.	DP=40;VDB=0.727385;SGB=-0.693145;MQSB=1;M…

```
files=$(ls *.call.vcf)
for file in $files; do name=${file%%".call.vcf"}; cat
${name}.call.vcf | grep -v "INDEL" > test.vcf
```

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO
CP007494	18817	.	C	A	177	.	DP=40;VDB=0.706575;SGB=-0.693145;MQSB=1;
CP007494	23192	.	T	C	177	.	DP=40;VDB=0.688692;SGB=-0.693145;MQSB=1;
CP007494	23198	.	A	G	177	.	DP=40;VDB=0.699478;SGB=-0.693145;MQSB=1;
CP007494	36469	.	A	G	175	.	DP=40;VDB=0.727385;SGB=-0.693145;MQSB=1;
CP007494	49966	.	G	A	176	.	DP=40;VDB=0.589467;SGB=-0.693144;MQSB=1;
CP007494	56749	.	C	T	176	.	DP=40;VDB=0.611779;SGB=-0.693144;MQSB=1;
CP007494	101035	.	C	G	39.3362	.	DP=18;VDB=0.0014347;SGB=-0.636426;RPB=0.
CP007494	101036	.	G	T	39.3362	.	DP=18;VDB=0.00113077;SGB=-0.636426;RPB=0
CP007494	101042	.	C	G	4.03223	.	DP=18;VDB=0.00113077;SGB=-0.636426;RPB=0
CP007494	101045	.	C	A	4.03223	.	DP=18;VDB=0.0014347;SGB=-0.636426;RPB=0.
CP007494	101046	.	A	T	4.03223	.	DP=18;VDB=0.00113077;SGB=-0.636426;RPB=0
CP007494	101048	.	T	G	4.03223	.	DP=18;VDB=0.00130514;SGB=-0.636426;RPB=0
CP007494	101095	.	G	A	141	.	DP=16;VDB=0.000162435;SGB=-0.688148;MQSB
CP007494	101182	.	A	G	165	.	DP=24;VDB=0.00155009;SGB=-0.692717;MQSB=
CP007494	135659	.	G	A	176	.	DP=40;VDB=0.658446;SGB=-0.693144;MQSB=1;
CP007494	158886	.	A	G	178	.	DP=40;VDB=0.647495;SGB=-0.693144;MQSB=1;
CP007494	181835	.	C	T	177	.	DP=40;VDB=0.720531;SGB=-0.693145;MQSB=1;
CP007494	185923	.	T	C	177	.	DP=40;VDB=0.706575;SGB=-0.693145;MQSB=1;
CP007494	187739	.	G	A	91	.	DP=8;VDB=0.000585975;SGB=-0.651104;MQSB=
CP007494	191456	.	G	A	178	.	DP=40;VDB=0.595224;SGB=-0.693144;MQSB=1;

# set quality treshold

```
#CHROM      POS       ID    REF   ALT   QUAL       FILTER   INFO
CP007494    23198     .     A     G     177        .        DP=40;VDB=0.699478;SGB=-0.693145;MQSB=1;
CP007494    36469     .     A     G     175        .        DP=40;VDB=0.727385;SGB=-0.693145;MQSB=1;
CP007494    56749     .     C     T     176        .        DP=40;VDB=0.611779;SGB=-0.693144;MQSB=1;
CP007494    101035    .     C     G     39.3362    .        DP=18;VDB=0.0014347;SGB=-0.636426;RPB=0.
CP007494    101036    .     G     T     39.3362    .        DP=18;VDB=0.00113077;SGB=-0.636426;RPB=0
CP007494    101042    .     C     G     4.03223    .        DP=18;VDB=0.00113077;SGB=-0.636426;RPB=0
CP007494    101045    .     C     A     4.03223    .        DP=18;VDB=0.0014347;SGB=-0.636426;RPB=0.
CP007494    101046    .     A     T     4.03223    .        DP=18;VDB=0.00113077;SGB=-0.636426;RPB=0
CP007494    101048    .     T     G     4.03223    .        DP=18;VDB=0.00130514;SGB=-0.636426;RPB=0
CP007494    101095    .     G     A     141        .        DP=16;VDB=0.000162435;SGB=-0.688148;MQSB
CP007494    101182    .     A     G     165        .        DP=24;VDB=0.00155009;SGB=-0.692717;MQSB=
CP007494    135659    .     G     A     176        .        DP=40;VDB=0.658446;SGB=-0.693144;MQSB=1;
CP007494    158886    .     A     G     178        .        DP=40;VDB=0.647495;SGB=-0.693144;MQSB=1;
CP007494    181835    .     C     T     177        .        DP=40;VDB=0.720531;SGB=-0.693145;MQSB=1;
CP007494    187739    .     G     A     91         .        DP=8;VDB=0.000585975;SGB=-0.651104;MQSB=
CP007494    191456    .     G     A     178        .        DP=40;VDB=0.595224;SGB=-0.693144;MQSB=1;
```

```
cat test.vcf | bcftools view -i '%QUAL>=80' > F061.calls.vcf
```

```
#CHROM      POS       ID    REF   ALT   QUAL       FILTER   INFO
CP007494    23198     .     A     G     177        .        DP=40;VDB=0.699478;SGB=-0.693145;MQSB=1;
CP007494    36469     .     A     G     175        .        DP=40;VDB=0.727385;SGB=-0.693145;MQSB=1;
CP007494    56749     .     C     T     176        .        DP=40;VDB=0.611779;SGB=-0.693144;MQSB=1;
CP007494    101095    .     G     A     141        .        DP=16;VDB=0.000162435;SGB=-0.688148;MQSB
CP007494    101182    .     A     G     165        .        DP=24;VDB=0.00155009;SGB=-0.692717;MQSB=
CP007494    135659    .     G     A     176        .        DP=40;VDB=0.658446;SGB=-0.693144;MQSB=1;
CP007494    158886    .     A     G     178        .        DP=40;VDB=0.647495;SGB=-0.693144;MQSB=1;
CP007494    181835    .     C     T     177        .        DP=40;VDB=0.720531;SGB=-0.693145;MQSB=1;
CP007494    187739    .     G     A     91         .        DP=8;VDB=0.000585975;SGB=-0.651104;MQSB=
CP007494    191456    .     G     A     178        .        DP=40;VDB=0.595224;SGB=-0.693144;MQSB=1;
```

# zip the manipulated file using bgzip

```
files=$(ls *.calls.vcf)
for file in $files; do name=${file%%".calls.vcf"}; bgzip
${name}.calls.vcf > ${name}.calls.vcf.gz; done
```

# index the variant files

```
files=$(ls *.calls.vcf.gz)
for file in $files; do name=${file%%".calls.vcf.gz"}; bcftools
index ${name}.calls.vcf.gz; done
```

# generate consensus sequence from variants (use the same ref file as in bowtie2)

```
files=$(ls *.calls.vcf.gz)
for file in $files; do name=${file%%".calls.vcf.gz"}; cat
ref.fa | bcftools consensus -o ${name}.cns.fa
${name}.calls.vcf.gz > ${name}.out; done
```

```
# change header to file name plus title string and make fasta sequence one line
files=$(ls *.cns.fa)
for file in $files; do name=${file%%".cns.fa"}; printf
">"$name" alignment against ref genome \n" > ${name}-cns.fasta

# changes the header to the actual strain name alignment …
# >CP007494 Bordetella holmesii ATCC 54514, IS masked
# to >$name alignment against ref genome
# writes only this line

cat ${name}.cns.fa \
| awk 'BEGIN{RS=">";FS="\n"}NR>1{seq="";for (i=2;i<=NF;i++)
seq=seq""$i; print seq}' >> ${name}-cns.fasta | echo “done with
$name”; done
# RS - Record Separator: end of record marker, default new line
# FS - Field Separator: \n separates the fields
# for all fields except the first, until the last field, seq is
seq plus the next seq
```

# join all consensus sequences into a multiple genome alignment

format: $1(Title) $2(Seq)

```
cat ${name}.cns.fa | perl ~/bin/mergelines2.pl | awk -v FS=" "
-v OFS="\t"} ‘{print $1,$6}’ | tr -d “>” > Bholmesii.phy
```

```
# cat ${name}.cns.fa – opens all consensus sequences
# perl ~/bin/mergelines2.pl – merges every 2 lines into 1

#  from  >$name alignment against ref genome
         Sequence

# to  >$name alignment against ref genome     Sequence

# awk -v FS=" " -v OFS="\t"} ‘{print $1,$6}’
# fields are separated by spaces,
# print $1 (>$name) and $6 (Sequence)

# tr -d “>” – delete “>”

 $1       $2
 Strain   Sequence
```

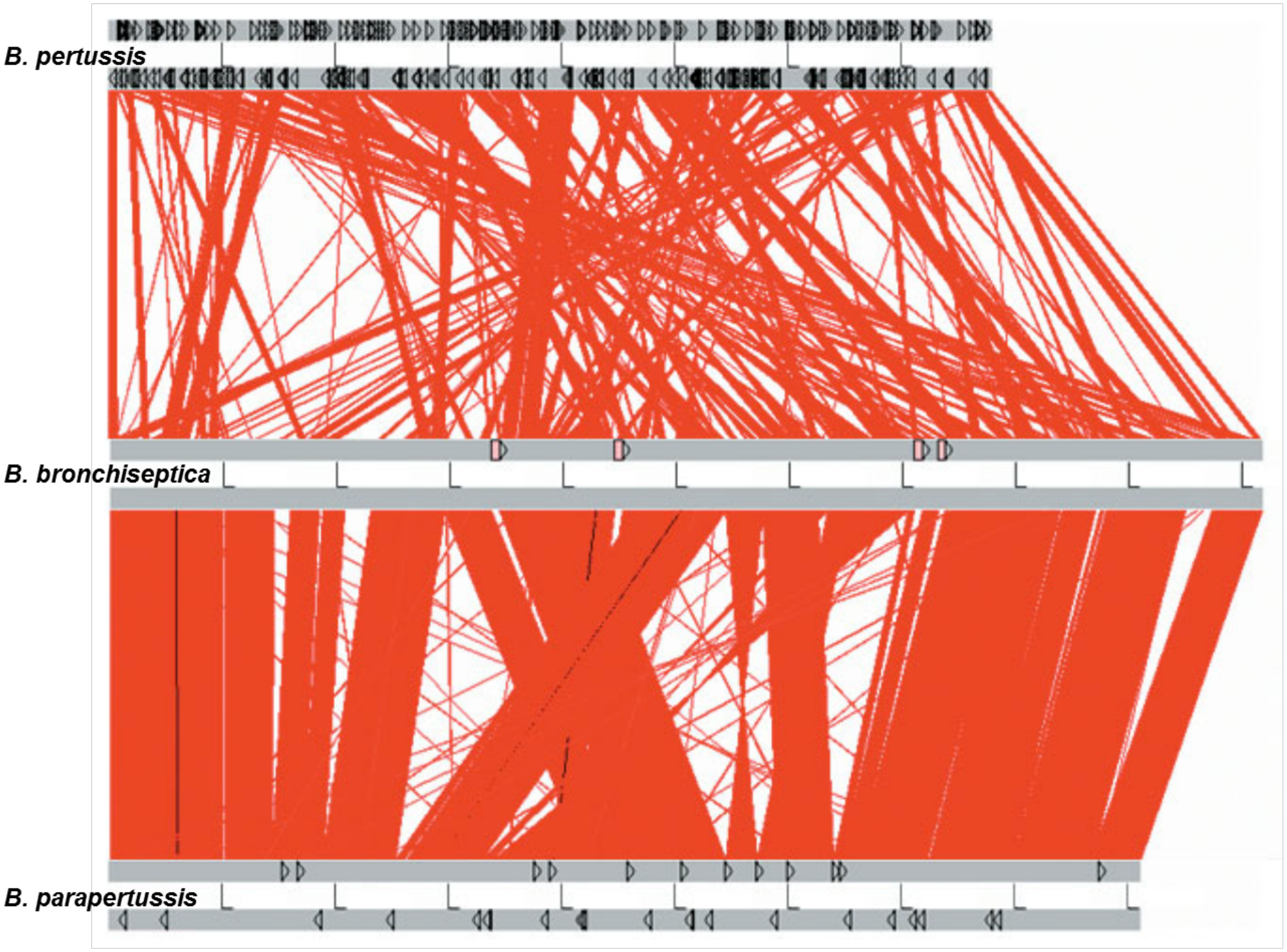
→ Generated a multiple genome alignment

format: $1(Title) $2(Seq)

How to perform a pairwise genome comparison and display in ACT?

1. Whole Genome Blast – genome comparison
2. MSPcrunch – change blast format to Artemis input

ACT – Artemis Comparison Tool



Parkhill et al 2003 Nat Genet 35: 32-40

How to perform a pairwise genome comparison and display in ACT?

1. Whole Genome Blast – genome comparison
2. MSPcrunch – change blast format to Artemis input

`Blastall`

go to: `ftp://ftp.ncbi.nlm.nih.gov/toolbox/ncbi_tools/old`

select toolbox folder, e.g. `20120620`

click on `ncbi.tar.gz` to download

go to "Downloads" on your computer

to unpack type: `tar -xvzf ncbi.tar.gz`

to `make` type: `./ncbi/make/makedis.csh`

change directory: `cd ncbi/bin`

copy everything to: `/home/[user]/bin` (change to your bin directory)

`MSPcrunch`

Get MSPcrunch from:

`http://sonnhammer.sbc.su.se/download/software/MSPcrunch+Blixem/`

install (or get the compiled program from me)

How to perform a pairwise genome comparison and display in ACT?

1. Whole Genome Blast – genome comparison
2. MSPcrunch – change blast format to Artemis input

```
# need fasta files of both genomes
# generate data base, use “formatbd”
formatdb –i genome1.fasta –p F –o T

# -i: input Fasta file
# -p: T input type protein, F nucleotide sequence
# -o: T output database NCBI styled, F none

# output:
# genome1.nhr
# genome1.nin
# genome1.nsd
# genome1.nsi
# genome1.nsq
```

How to perform a genome comparison and display in ACT?

1. Whole Genome Blast – genome comparison
2. MSPcrunch – change blast format to Artemis input

```
# need fasta files of both genomes
# run blastn
# syntax: blastall -p [program] -d [database] -i
[subject genome] -b [max hits] -v [max hits] -o
[output file]
```

```
blastall -p blastn -d genome1.fasta -i genome2.fasta
-o genome1-genome2.out -v 1000000 -b 1000000
```

```
Query: 4599606 tggtgaggtcgggcgaatcgtcca
               |||||||||||| |||||||||||
Sbjct: 4074107 tggtgaggtcggacgaatcgtcca

Query: 4599666 caggagcttgttgcattgcgatgc
               ||||  ||||||||||||||||||
Sbjct: 4074047 cagggacttgttgcattgcgatgc
```

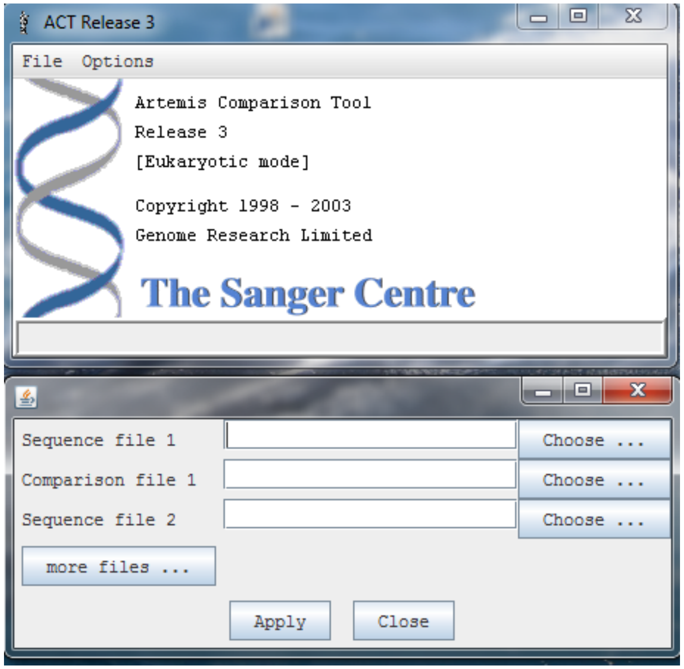
How to perform a genome comparison and display in ACT?

1. Whole Genome Blast – genome comparison
2. MSPcrunch – change blast format to Artemis input

```
# take blast output and change format to table
MSPcrunch -d genome1-genome1.out > genome1-
genome2.cmp

what you get:
score % sim from   to     gen1  from  to   genome2
10689 99.58 181497 183650 AXSJ     1  2154 Bb_RB50
 8233 99.82 183699 185350 AXSJ  2143  3794 Bb_RB50
```

```
# so, we got:
#     genome1.fasta (or genome1.gbk)
#     genome1-genome2.cmp
#     genome2.fasta (or genome2.gbk)
```

Load your files in ACT



Artemis Comparison Tool
Release 3
[Eukaryotic mode]

Copyright 1998 - 2003
Genome Research Limited

The Sanger Centre

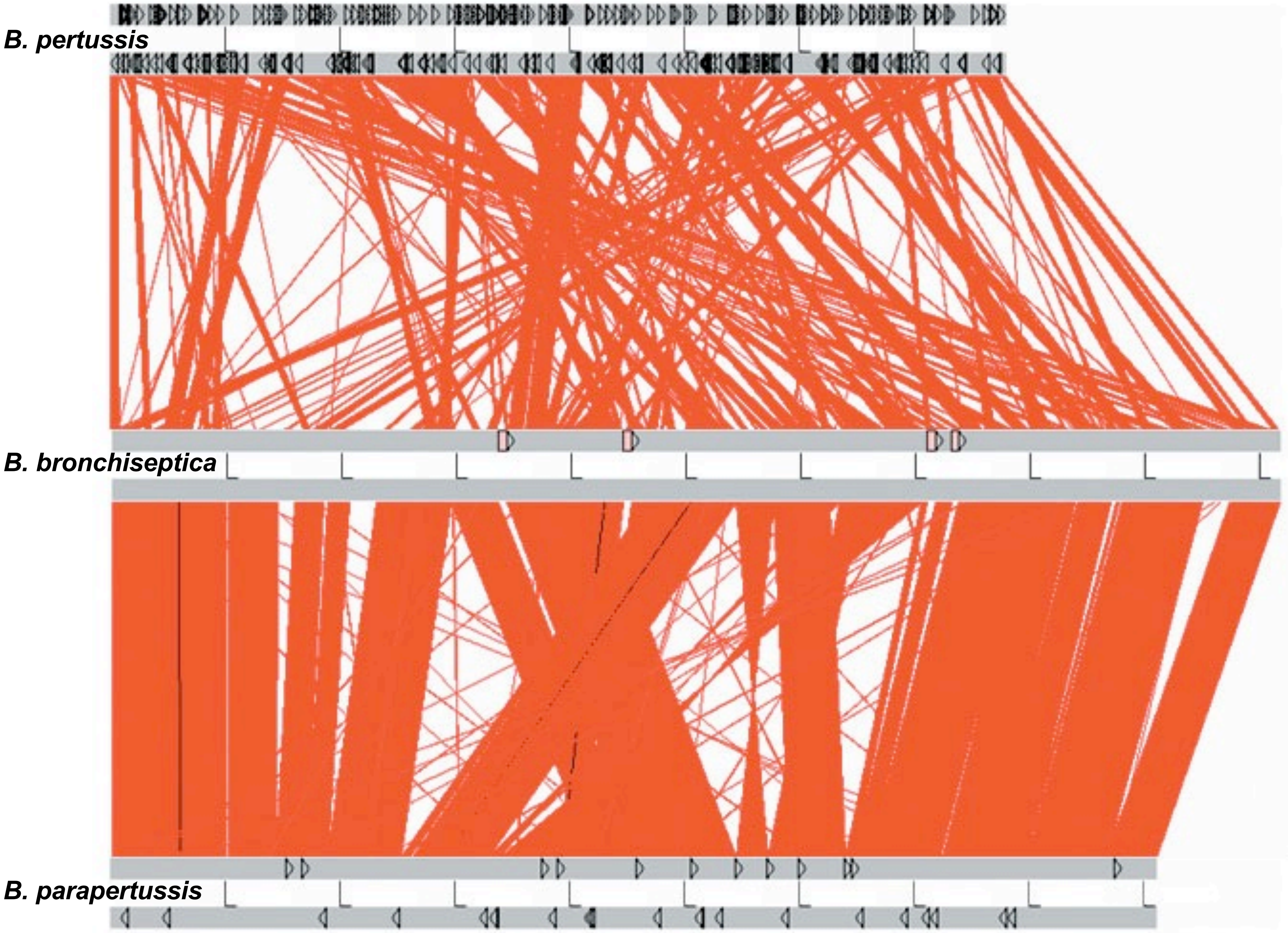
genome1.fasta (or genome1.gbk)

genome1-genome2.cmp

genome2.fasta (or genome2.gbk

https://www.sanger.ac.uk/science/tools/artemis-comparison-tool-act

ACT – Artemis Comparison Tool



Parkhill et al 2003 Nat Genet 35: 32-40

Let’s shift gears:
run genome comparison against multiple genomes in a loop

genome1: BhinziiL60.fasta vs genome2: BhinziiF582.fa

BhinziiH568.fa
BhinziiNCTC.fa
Bhinzii5132.fa
Bhinzii1277.fa
BhinziiCA90.fa

```
#!/bin/bash
# multiple_genomes_to_ACT.sh
# Author Bodo Linz
# run BLASTn and MSPcrunch for several genomes

DATABASE=BhinziiL60.fasta
BLASTALL=~/bin/blastall        # define location of program blastall
MSPCRUNCH=~/bin/MSPcrunch      # define location of program MSPcrunch
GENOME1=${DATABASE%%".fasta"} # database name without “.fasta”

# has the database already been formatted?

if [ -f ${DATABASE}.nhr -a ${DATABASE}.nin -a ${DATABASE}.nsd -a
${DATABASE}.nsi -a ${DATABASE}.nsq ]; then \
        echo "The database is already formatted"
else
        formatdb -i ${DATABASE} -p F -o T
        echo "Done formatting the database $GENOME1.fasta"
fi
```

Let’s shift gears:
run genome comparison against multiple genomes in a loop

genome1: BhinziiL60.fasta vs genome2:

BhinziiF582.fa
BhinziiH568.fa
BhinziiNCTC.fa
Bhinzii5132.fa
Bhinzii1277.fa
BhinziiCA90.fa

```
# list the genomes to compare
files=$(ls Bhinzii*.fa)# generate list of files

# BLAST the target sequence against the reference genome
echo "Running blastn of $GENOME1 against
$files"
echo "-------------------------------------"
echo ""

for file in $files; do GENOME2=${file%%".fa"}; $BLASTALL -p
blastn -d $DATABASE -i $GENOME2.fa -o $GENOME1-$GENOME2.out;
done

# loop: for all file(s) in list $files; do something; done


echo "Done with BLAST of $GENOME1 against
$files"
echo "-------------------------------------"
```

Let’s shift gears:
run genome comparison against multiple genomes in a loop

genome1: BhinziiL60.fasta
genome2: BhinziiF582.fa, BhinziiH568.fa, BhinziiNCTC.fa, Bhinzii5132.fa, Bhinzii1277.fa, BhinziiCA90.fa

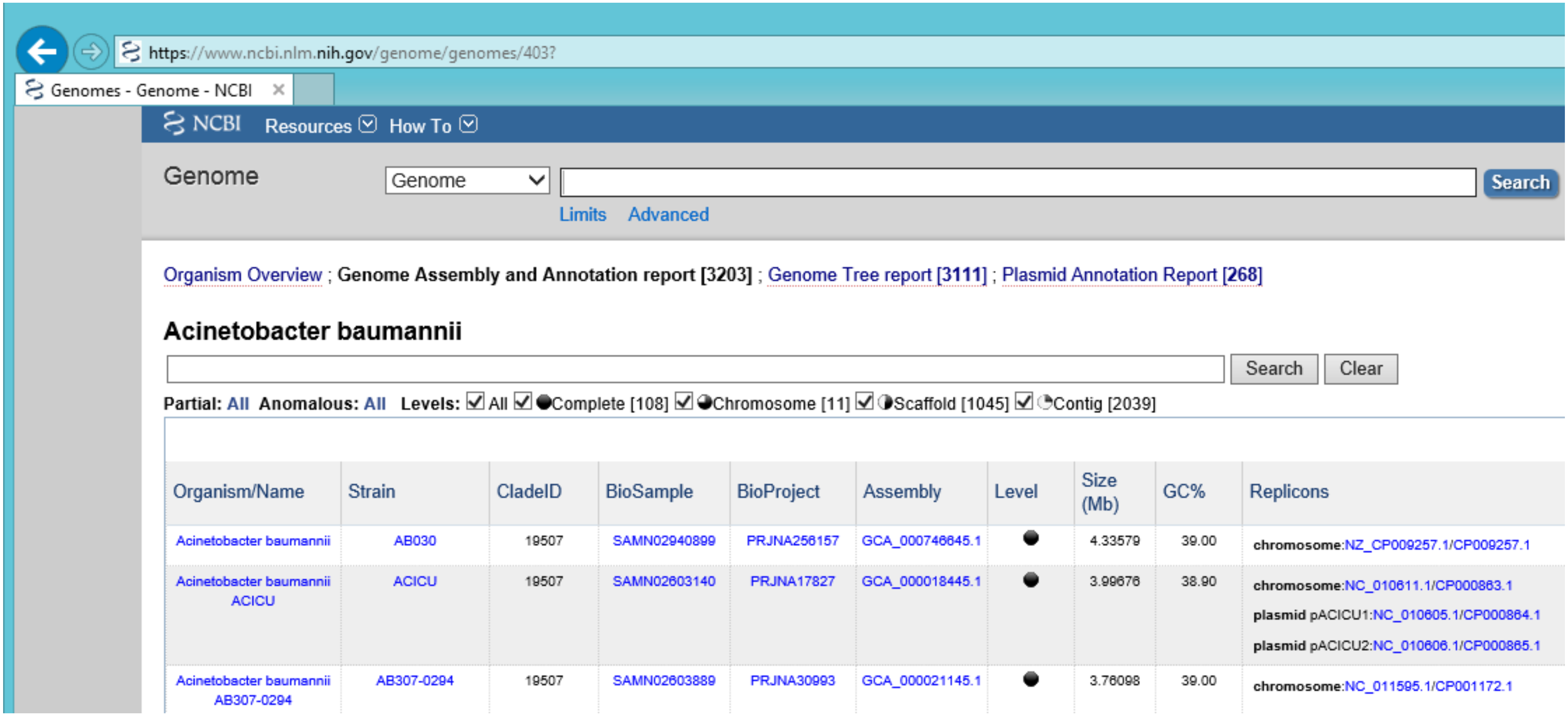
```
# Now: do the same for MSPcrunch
# list the BLAST output files
files=$(ls Bhinzii*.out)      # BhinziiL60-BhinziiF582.out etc.

# transform the blast output to ACT *.cmp table in MSPcrunch
echo "Running MSPcrunch of files
$files"
echo ""
echo "------------------------------------------"
echo ""
for file in $files; do name=${file%%".out"}; $MSPCRUNCH -d
$name.out > $name.cmp; done

echo "Done with MSPcrunch."
echo "------------------------------------------"
echo ""
echo "Done. Run ACT to visualize the genome comparison."
echo ""
```

works well for completed genomes

Problem: not suitable for genomes present as contigs

SADLY: most genomes are incomplete

EXAMPLE: *Acinetobacter baumannii* at ncbi genomes



Organism Overview ; Genome Assembly and Annotation report [3203] ; Genome Tree report [3111] ; Plasmid Annotation Report [268]

Acinetobacter baumannii

Partial: All **Anomalous:** All **Levels:** ☑ All ☑ ●Complete [108] ☑ ◕Chromosome [11] ☑ ◑Scaffold [1045] ☑ ◔Contig [2039]

Organism/Name	Strain	CladeID	BioSample	BioProject	Assembly	Level	Size (Mb)	GC%	Replicons
Acinetobacter baumannii	AB030	19507	SAMN02940899	PRJNA256157	GCA_000746645.1	●	4.33579	39.00	**chromosome**:NZ_CP009257.1/CP009257.1
Acinetobacter baumannii ACICU	ACICU	19507	SAMN02603140	PRJNA17827	GCA_000018445.1	●	3.99676	38.90	**chromosome**:NC_010611.1/CP000863.1 **plasmid** pACICU1:NC_010605.1/CP000864.1 **plasmid** pACICU2:NC_010606.1/CP000865.1
Acinetobacter baumannii AB307-0294	AB307-0294	19507	SAMN02603889	PRJNA30993	GCA_000021145.1	●	3.76098	39.00	**chromosome**:NC_011595.1/CP001172.1

Let’s download genomes

as contigs to run `blastall` and `MSPcrunch`

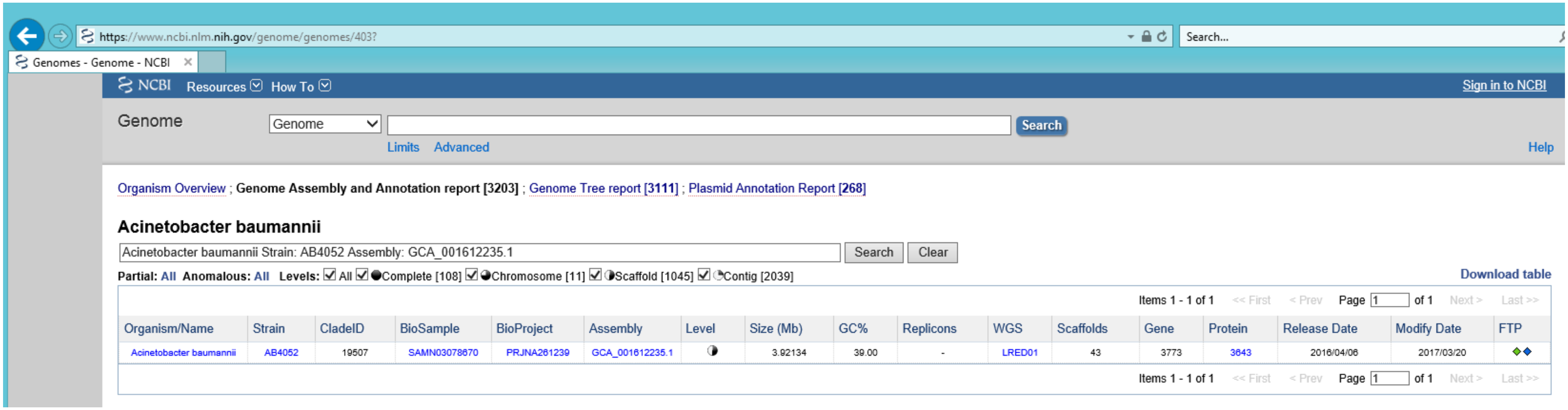
go to https://www.ncbi.nlm.nih.gov/genome/

type the species: *Acinetobacter baumannii*

Select: Genome Assembly and Annotation report

type the isolate: AB4052

click on LRED01 in WGS



Organism Overview ; **Genome Assembly and Annotation report [3203]** ; Genome Tree report [3111] ; Plasmid Annotation Report [268]

Acinetobacter baumannii

Acinetobacter baumannii Strain: AB4052 Assembly: GCA_001612235.1 [Search] [Clear]

Partial: All **Anomalous:** All **Levels:** ☑ All ☑ ●Complete [108] ☑ ◕Chromosome [11] ☑ ◑Scaffold [1045] ☑ ◔Contig [2039]

Download table

Items 1 - 1 of 1 << First < Prev Page 1 of 1 Next > Last >>

Organism/Name	Strain	CladeID	BioSample	BioProject	Assembly	Level	Size (Mb)	GC%	Replicons	WGS	Scaffolds	Gene	Protein	Release Date	Modify Date	FTP
Acinetobacter baumannii	AB4052	19507	SAMN03078670	PRJNA261239	GCA_001612235.1	◑	3.92134	39.00	-	LRED01	43	3773	3643	2016/04/06	2017/03/20	◆◆

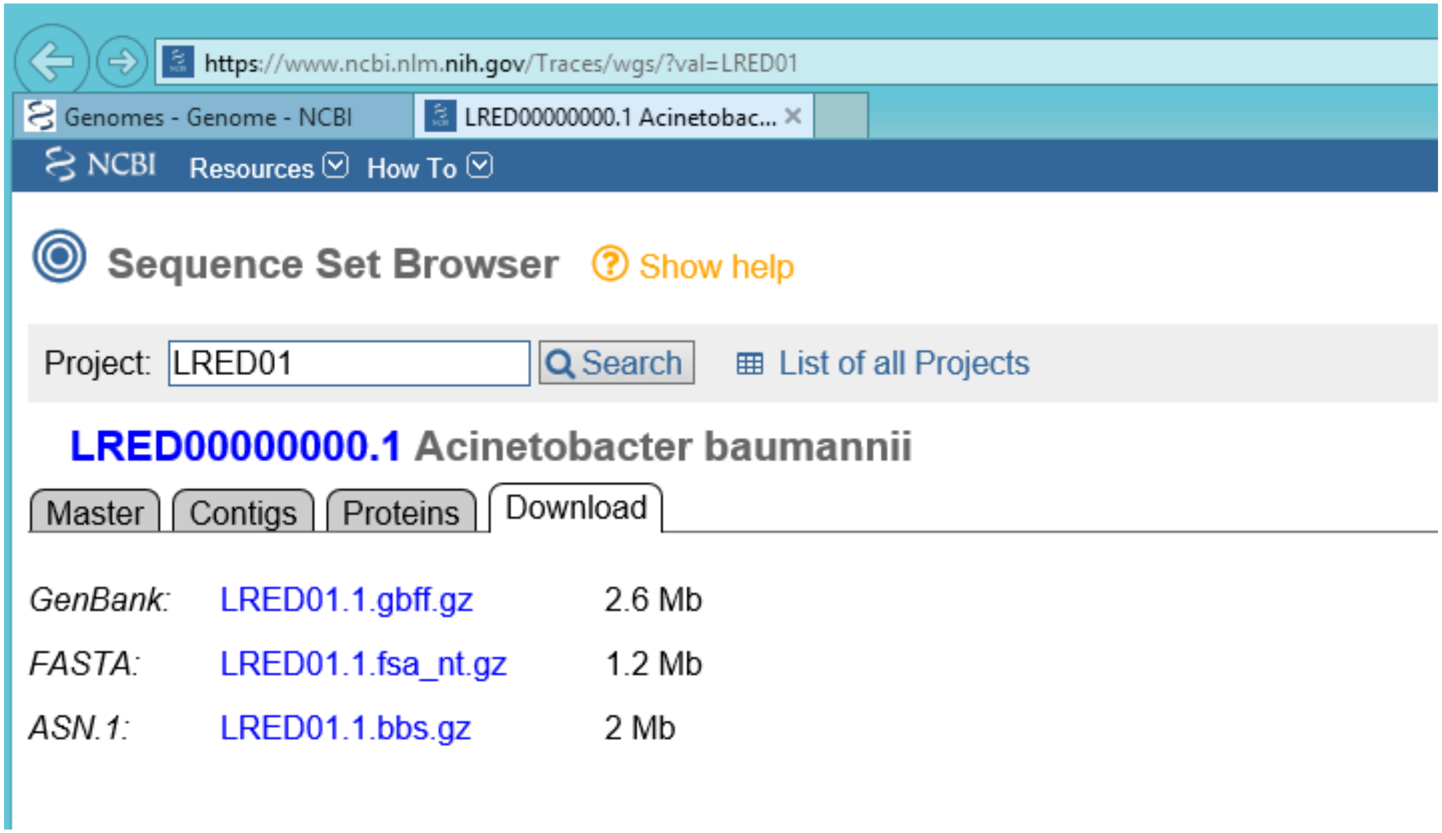
Items 1 - 1 of 1 << First < Prev Page 1 of 1 Next > Last >>

Let’s download genomes

click on LRED01.1.fsa_nt.gz, download

unpack: `gzip LRED01.1.fsa_nt.gz`

rename: `mv LRED01.1.fsa_nt LRED01.1.fsa`



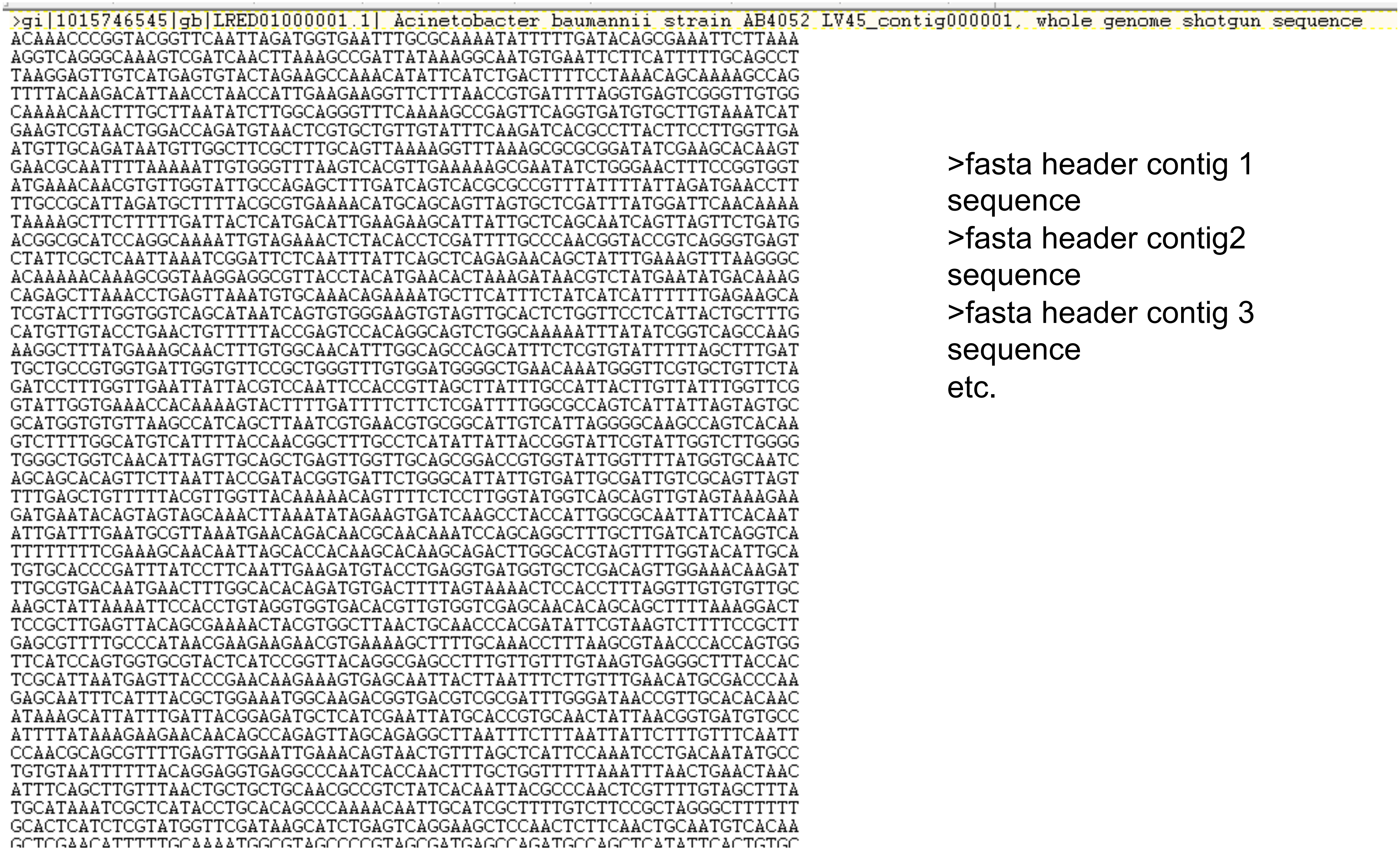
https://www.ncbi.nlm.nih.gov/Traces/wgs/?val=LRED01

Project: LRED01

LRED00000000.1 Acinetobacter baumannii

GenBank:	LRED01.1.gbff.gz	2.6 Mb
FASTA:	LRED01.1.fsa_nt.gz	1.2 Mb
ASN.1:	LRED01.1.bbs.gz	2 Mb

We get

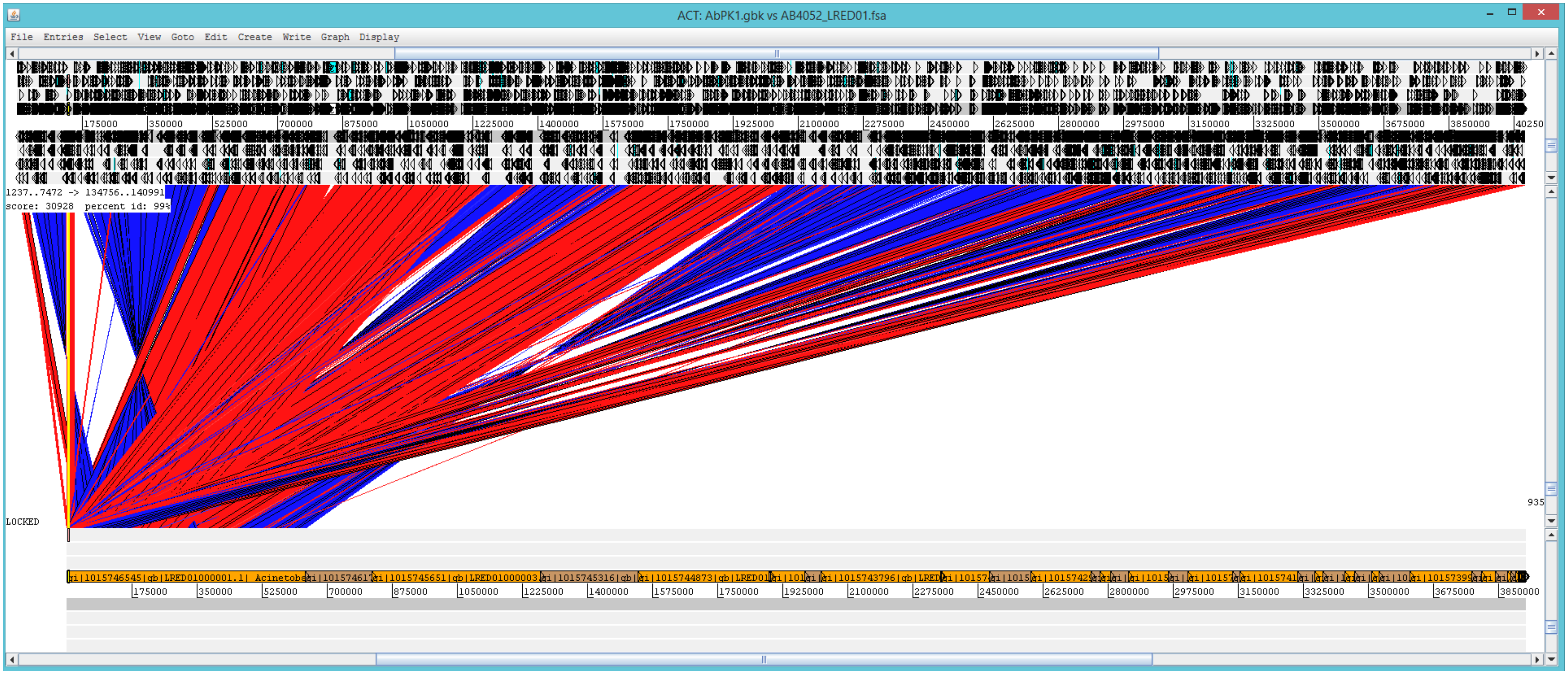


```
>gi|1015746545|gb|LRED01000001.1| Acinetobacter baumannii strain AB4052 LV45_contig000001, whole genome shotgun sequence
ACAAACCCGGTACGGTTCAATTAGATGGTGAATTTGCGCAAAATATTTTTGATACAGCGAAATTCTTAAA
AGGTCAGGGCAAAGTCGATCAACTTAAAGCCGATTATAAAGGCAATGTGAATTCTTCATTTTTGCAGCCT
TAAGGAGTTGTCATGAGTGTACTAGAAGCCAAACATATTCATCTGACTTTTCCTAAACAGCAAAAGCCAG
TTTTACAAGACATTAACCTAACCATTGAAGAAGGTTCTTTAACCGTGATTTTAGGTGAGTCGGGTTGTGG
CAAAACAACTTTGCTTAATATCTTGGCAGGGTTTCAAAAGCCGAGTTCAGGTGATGTGCTTGTAAATCAT
GAAGTCGTAACTGGACCAGATGTAACTCGTGCTGTTGTATTTCAAGATCACGCCTTACTTCCTTGGTTGA
ATGTTGCAGATAATGTTGGCTTCGCTTTGCAGTTAAAAGGTTTAAAGCGCGCGGATATCGAAGCACAAGT
GAACGCAATTTTAAAAATTGTGGGTTTAAGTCACGTTGAAAAAGCGAATATCTGGGAACTTTCCGGTGGT
ATGAAACAACGTGTTGGTATTGCCAGAGCTTTGATCAGTCACGCGCCGTTTATTTTATTAGATGAACCTT
TTGCCGCATTAGATGCTTTTACGCGTGAAAACATGCAGCAGTTAGTGCTCGATTTATGGATTCAACAAAA
TAAAAGCTTCTTTTTGATTACTCATGACATTGAAGAAGCATTATTGCTCAGCAATCAGTTAGTTCTGATG
ACGGCGCATCCAGGCAAAATTGTAGAAACTCTACACCTCGATTTTGCCCAACGGTACCGTCAGGGTGAGT
CTATTCGCTCAATTAAATCGGATTCTCAATTTATTCAGCTCAGAGAACAGCTATTTGAAAGTTTAAGGGC
ACAAAAACAAAGCGGTAAGGAGGCGTTACCTACATGAACACTAAAGATAACGTCTATGAATATGACAAAG
CAGAGCTTAAACCTGAGTTAAATGTGCAAACAGAAAATGCTTCATTTCTATCATCATTTTTTGAGAAGCA
TCGTACTTTGGTGGTCAGCATAATCAGTGTGGGAAGTGTAGTTGCACTCTGGTTCCTCATTACTGCTTTG
CATGTTGTACCTGAACTGTTTTTACCGAGTCCACAGGCAGTCTGGCAAAAATTTATATCGGTCAGCCAAG
AAGGCTTTATGAAAGCAACTTTGTGGCAACATTTGGCAGCCAGCATTTCTCGTGTATTTTTAGCTTTGAT
TGCTGCCGTGGTGATTGGTGTTCCGCTGGGTTTGTGGATGGGGCTGAACAAATGGGTTCGTGCTGTTCTA
GATCCTTTGGTTGAATTATTACGTCCAATTCCACCGTTAGCTTATTTGCCATTACTTGTTATTTGGTTCG
GTATTGGTGAAACCACAAAAGTACTTTTGATTTTCTTCTCGATTTTGGCGCCAGTCATTATTAGTAGTGC
GCATGGTGTGTTAAGCCATCAGCTTAATCGTGAACGTGCGGCATTGTCATTAGGGGCAAGCCAGTCACAA
GTCTTTTGGCATGTCATTTTACCAACGGCTTTGCCTCATATTATTACCGGTATTCGTATTGGTCTTGGGG
TGGGCTGGTCAACATTAGTTGCAGCTGAGTTGGTTGCAGCGGACCGTGGTATTGGTTTTATGGTGCAATC
AGCAGCACAGTTCTTAATTACCGATACGGTGATTCTGGGCATTATTGTGATTGCGATTGTCGCAGTTAGT
TTTGAGCTGTTTTTACGTTGGTTACAAAAACAGTTTTCTCCTTGGTATGGTCAGCAGTTGTAGTAAAGAA
GATGAATACAGTAGTAGCAAACTTAAATATAGAAGTGATCAAGCCTACCATTGGCGCAATTATTCACAAT
ATTGATTTGAATGCGTTAAATGAACAGACAACGCAACAAATCCAGCAGGCTTTGCTTGATCATCAGGTCA
TTTTTTTTCGAAAGCAACAATTAGCACCACAAGCACAAGCAGACTTGGCACGTAGTTTTGGTACATTGCA
TGTGCACCCGATTTATCCTTCAATTGAAGATGTACCTGAGGTGATGGTGCTCGACAGTTGGAAACAAGAT
TTGCGTGACAATGAACTTTGGCACACAGATGTGACTTTTAGTAAAACTCCACCTTTAGGTTGTGTGTTGC
AAGCTATTAAAATTCCACCTGTAGGTGGTGACACGTTGTGGTCGAGCAACACAGCAGCTTTTAAAGGACT
TCCGCTTGAGTTACAGCGAAAACTACGTGGCTTAACTGCAACCCACGATATTCGTAAGTCTTTTCCGCTT
GAGCGTTTTGCCCATAACGAAGAAGAACGTGAAAAGCTTTTGCAAACCTTTAAGCGTAACCCACCAGTGG
TTCATCCAGTGGTGCGTACTCATCCGGTTACAGGCGAGCCTTTGTTGTTTGTAAGTGAGGGCTTTACCAC
TCGCATTAATGAGTTACCCGAACAAGAAAGTGAGCAATTACTTAATTTCTTGTTTGAACATGCGACCCAA
GAGCAATTTCATTTACGCTGGAAATGGCAAGACGGTGACGTCGCGATTTGGGATAACCGTTGCACACAAC
ATAAAGCATTATTTGATTACGGAGATGCTCATCGAATTATGCACCGTGCAACTATTAACGGTGATGTGCC
ATTTTATAAAGAAGAACAACAGCCAGAGTTAGCAGAGGCTTAATTTCTTTAATTATTCTTTGTTTCAATT
CCAACGCAGCGTTTTGAGTTGGAATTGAAACAGTAACTGTTTAGCTCATTCCAAATCCTGACAATATGCC
TGTGTAATTTTTTACAGGAGGTGAGGCCCAATCACCAACTTTGCTGGTTTTTAAATTTAACTGAACTAAC
ATTTCAGCTTGTTTAACTGCTGCTGCAACGCCGTCTATCACAATTACGCCCAACTCGTTTTGTAGCTTTA
TGCATAAATCGCTCATACCTGCACAGCCCAAAACAATTGCATCGCTTTTGTCTTCCGCTAGGGCTTTTTT
GCACTCATCTCGTATGGTTCGATAAGCATCTGAGTCAGGAAGCTCCAACTCTTCAACTGCAATGTCACAA
GCTCGAACATTTTTGCAAAATGGCGTAGCCCCGTAGCGATGAGCCAGATGCCAGCTCATATTCACTGTGC
```

>fasta header contig 1
sequence
>fasta header contig2
sequence
>fasta header contig 3
sequence
etc.

Let's assume we ran `blastall` and `MSPcrunch`: complete genome against genome in contigs

This is what we get:

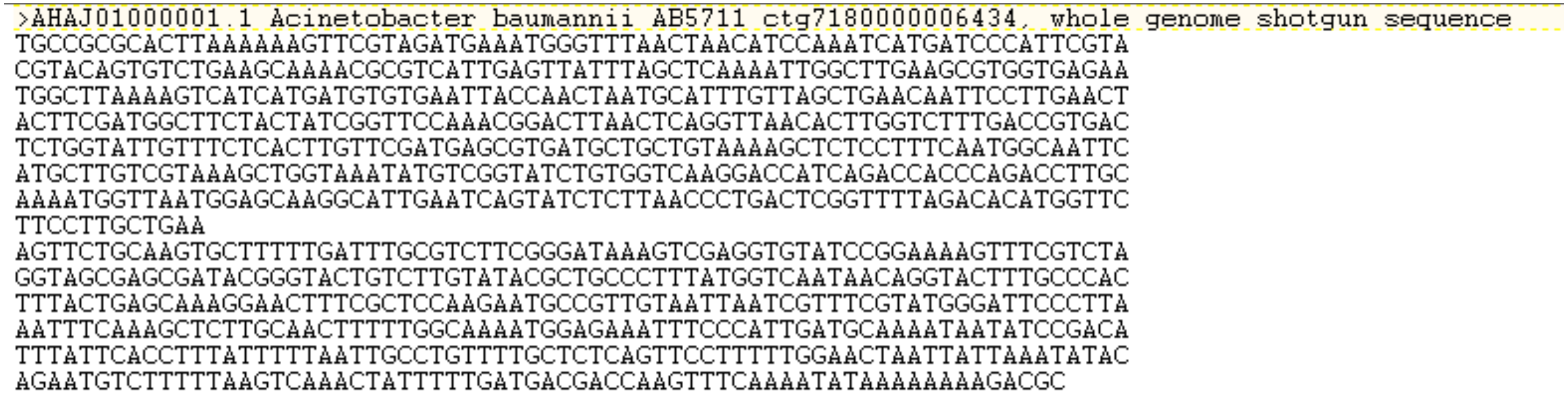


```
1237..7472 -> 134756..140991
score: 30928  percent id: 99%
```

All hits against the first contig

Solution: modify the genome format

Solution 1: keep only the first fasta header
remove all following fasta headers



```
>AHAJ01000001.1 Acinetobacter baumannii AB5711 ctg7180000006434, whole genome shotgun sequence
TGCCGCGCACTTAAAAAAGTTCGTAGATGAAATGGGTTTAACTAACATCCAAATCATGATCCCATTCGTA
CGTACAGTGTCTGAAGCAAAACGCGTCATTGAGTTATTTAGCTCAAAATTGGCTTGAAGCGTGGTGAGAA
TGGCTTAAAAGTCATCATGATGTGTGAATTACCAACTAATGCATTTGTTAGCTGAACAATTCCTTGAACT
ACTTCGATGGCTTCTACTATCGGTTCCAAACGGACTTAACTCAGGTTAACACTTGGTCTTTGACCGTGAC
TCTGGTATTGTTTCTCACTTGTTCGATGAGCGTGATGCTGCTGTAAAAGCTCTCCTTTCAATGGCAATTC
ATGCTTGTCGTAAAGCTGGTAAATATGTCGGTATCTGTGGTCAAGGACCATCAGACCACCCAGACCTTGC
AAAATGGTTAATGGAGCAAGGCATTGAATCAGTATCTCTTAACCCTGACTCGGTTTTAGACACATGGTTC
TTCCTTGCTGAA
AGTTCTGCAAGTGCTTTTTGATTTGCGTCTTCGGGATAAAGTCGAGGTGTATCCGGAAAAGTTTCGTCTA
GGTAGCGAGCGATACGGGTACTGTCTTGTATACGCTGCCCTTTATGGTCAATAACAGGTACTTTGCCCAC
TTTACTGAGCAAAGGAACTTTCGCTCCAAGAATGCCGTTGTAATTAATCGTTTCGTATGGGATTCCCTTA
AATTTCAAAGCTCTTGCAACTTTTTGGCAAAATGGAGAAATTTCCCATTGATGCAAAATAATATCCGACA
TTTATTCACCTTTATTTTTAATTGCCTGTTTTGCTCTCAGTTCCTTTTTGGAACTAATTATTAAATATAC
AGAATGTCTTTTTAAGTCAAACTATTTTTGATGACGACCAAGTTTCAAAATATAAAAAAAAGACGC
```

```
printf ">AHAJ01000001.1\n" > AHAJ01.fa
# print everything between " "
# and save as file AHAJ01.fa
cat AHAJ01.1.fsa | grep -v ">"  >> AHAJ01.fa
# >> add to file AHAJ01.fa and save

# What does the grep command do?
```

>AHAJ01000001.1

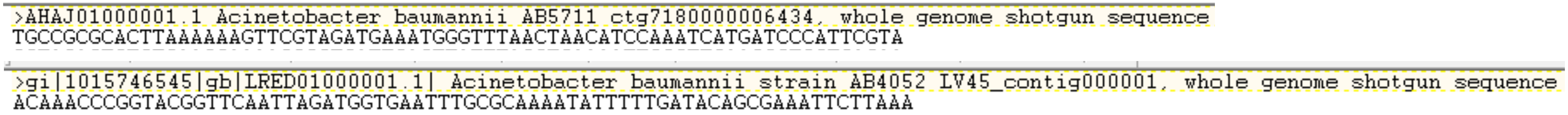
Solution: modify the genome format

OR (a little more sophisticated)

```
printf ">AHAJ01000001.1\n" > AHAJ01.fa
cat AHAJ01.1.fsa \
   | awk '{
            if(substr($1,1,1) == ">"){
                  printf "";
            }else{
                  printf "%s",$1;
                  printf "\n";
           }
     }’ >> AHAJ01.fa
```

```
# substr: substring
# if $1 at position 1 for 1 character = ">", print nothing
# else print
# printf "%s" - take the first of the following arguments ($1) and
print it as a string (s), "%d" - as a number (decimal)
# then print "\n"
# >> add to file AHAJ01.fa
```

Note the different headers



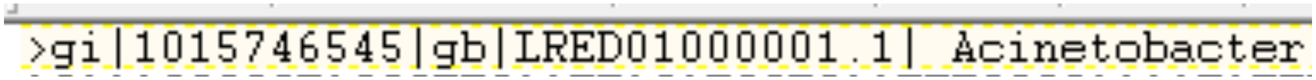
```
>AHAJ01000001.1 Acinetobacter baumannii AB5711 ctg7180000006434, whole genome shotgun sequence
TGCCGCGCACTTAAAAAAGTTCGTAGATGAAATGGGTTTAACTAACATCCAAATCATGATCCCATTCGTA
```

```
>gi|1015746545|gb|LRED01000001.1| Acinetobacter baumannii strain AB4052 LV45_contig000001, whole genome shotgun sequence
ACAAACCCGGTACGGTTCAATTAGATGGTGAATTTGCGCAAAATATTTTTGATACAGCGAAATTCTTAAA
```

```
# modify genome input file to format ">LRED01000001.1"
cat $GENOME2 \
| awk '{
       if(substr($1,1,3) == ">gi"){
               printf ">";
               printf substr($1,19,14);
               printf "\n";
       }else{
               printf "%s",$1;
               printf "\n"
       }
      }' \
> AB4052_genome.fasta
```

Ready for blast and MSPcrunch ….

Let’s walk through



```
>gi|1015746545|gb|LRED01000001.1| Acinetobacter
```

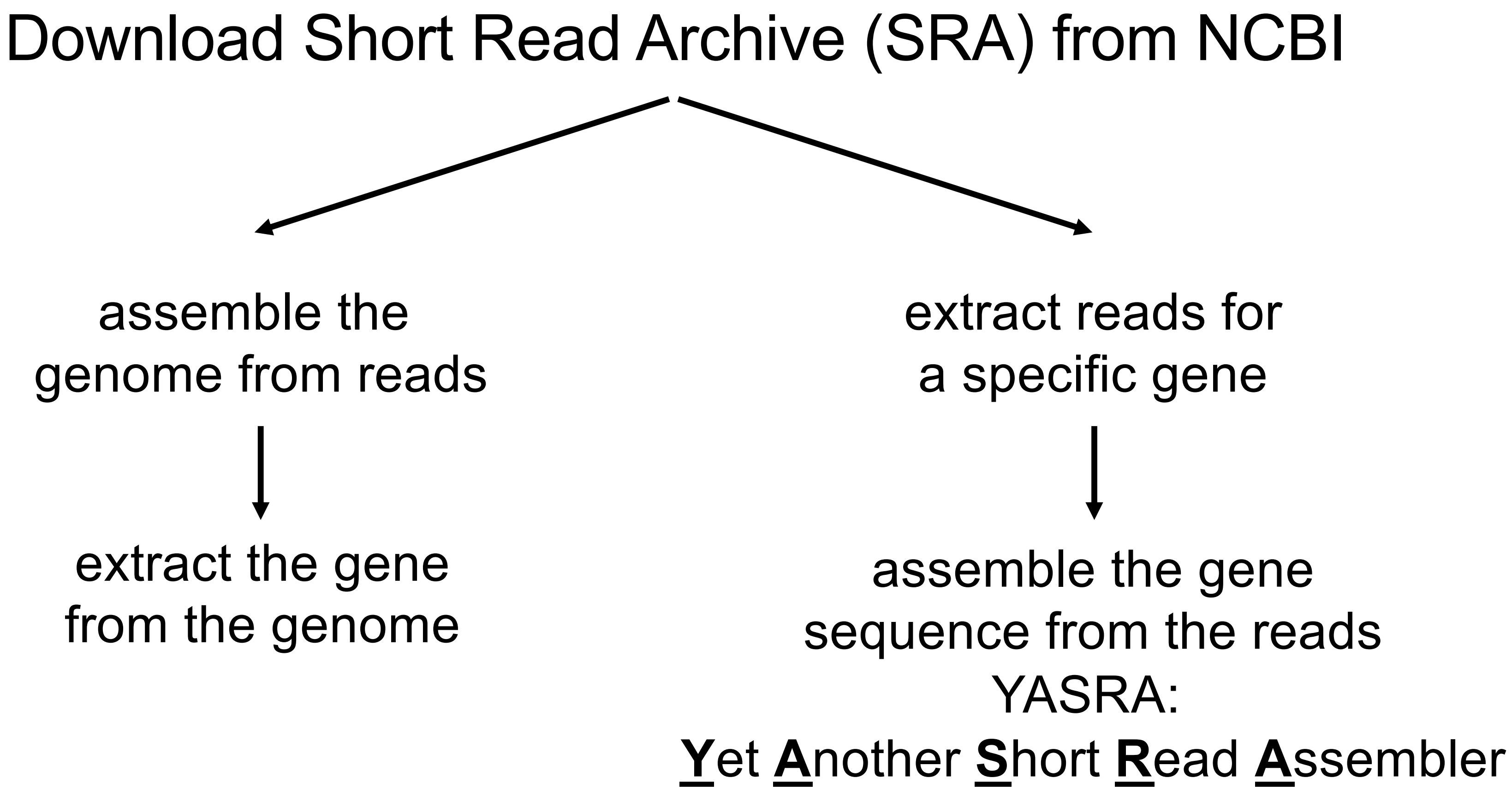
```
cat $GENOME2 \
| awk '{
       if(substr($1,1,3) == ">gi"){
# if at pos $1 the substring starting from character 1 for 3 characters
# equals (exactly) ">gi"
               printf">";
               printf substr($1,19,14);
               printf"\n";
# then print “>”
# then print the substring of 14 characters starting from character 19
# which is “LRED01000001.1”
# then print “\n” (carriage return)
       }else{
               printf"%s",$1;
               printf"\n"
# if criterion is not met, print all lines, then print “\n”
       }
      }' \
> AB4052_genome.fasta
We Get: >LRED01000001.1
        >AHAJ01000001.1 Acinetobacter baumannii AB5711 ctg7180000…
        → We took care of the different headers
```

Thank you.

Question?

Let’s back up some:

How to get a specific gene sequence from a Short Read Archive



Download Short Read Archive (SRA) from NCBI

assemble the
genome from reads

extract the gene
from the genome

extract reads for
a specific gene

assemble the gene
sequence from the reads
YASRA:
Yet **A**nother **S**hort **R**ead **A**ssembler

Extract the reads for a certain membrane transporter gene
(locus_tag BB1335 in *B. bronchiseptica* RB50)

to check for a frameshift mutation in a *B. hinzii* genome

Expected length without frameshift: 1416 bp
Expected length with -1 frameshift: 1415 bp

- We use `lastZ` and `YASRA`

Harris, R.S. (2007) Improved pairwise alignment of genomic DNA. Ph.D. Thesis, The Pennsylvania State University. (http://www.bx.psu.edu/~rsharris/lastz/)
download and install

- Download SRA
- Run FLASH to join the reads

```
flash SRR942665_1.fastq SRR942665_2.fastq -m 10 -M 100 -x 0.1
```

rename: `mv out.extendedFrags.fastq Bhz5132_SRR942665_joined.fastq`
rename: `mv out.notCombined_1.fastq Bhz5132_SRR942665_nc1.fastq`
rename: `mv out.notCombined_2.fastq Bhz5132_SRR942665_nc2.fastq`

Let’s dig in:

```
cat SRR942665_joined.fastq | lastz BB1335.fa[nameparse=darkspace]
/dev/stdin[nameparse=-full] --yasra90 --coverage=75
--ambiguous=iupac --format=general:name1,zstart1,end1,
name2,strand2,zstart2,end2,nucs2,quals2
| grep -v "^#"
| awk -v FS="\t" '{print $0,$4}'
| uniq -u -f 8
| awk -v FS="\t" -v OFS="\t" '{print $1,$2,$3,$4,$5,$6,$7,$8,$9}'
| sort -k 2,2n -k 3,3n
| ~/bodo.1/bin/YASRA-2.33/src/assembler -r -o -c -h /dev/stdin
> Bhinzii5132_BB1335_consensus.fa
```

- WOW!

- DON’T PANIC !!!

- Let’s walk through …

```
cat SRR942665_joined.fastq  # open file
| lastz BB1335.fa[nameparse=darkspace] /dev/stdin[nameparse=-
full]   # call the program lastz, which aligns the reads against
sequence BB1335.fa, our target gene
--yasra90 --coverage=75  # min identity 90%, min length 75%
--ambiguous=iupac  # IUPAC Nucleotides allowed
--format=general:name1,zstart1,end1,
name2,strand2,zstart2,end2,nucs2,quals2 # format
# name1,zstart1,end1 – our target sequence BB1335.fa
# name2,nucs2,quals2 – sequencing reads to align
| grep -v "^#"  # don’t select reads that start with bad quality
| awk -v FS="\t" '{print $0,$4}' # print all $ plus $4 again
| uniq -u -f 8  # take only lines where field 8 ($8 = nucs2) is
  a unique sequence = if duplicated sequence take only once
| awk -v FS="\t" -v OFS="\t" '{print $1,$2,$3,$4,$5,$6,$7,$8,$9}'
# print all fields again
| sort -k 2,2n -k 3,3n
# sort by increasing position in target, first start then end
| ~/bodo.1/bin/YASRA-2.33/src/assembler -r -o -c -h /dev/stdin
# run the assembler
> Bhinzii5132_BB1335_consensus.fa
# save
```

Created consensus sequence: Bhinzii5132_BB1335_consensus.fa

```
>Contig1_BB1335_0_1415
ATGCTATCGACCATATTTTCGTTTTCCTCGCTGTACTTCGCCACGCTGTTGATGTTGATC
GGCACGGGCCTGTTCAACACCTATATGGGCCTGACCCTGACGGCGAAATCCGTCAACGAA
GTCTGGATCGGCTCCATGATCGCAGGGTATTACCTCGGCCTGGTCTGCGGGGCGCGGCTG
GGCCACAAACTCATCATCCGGGTGGGCCATATCCGGGCCTTCGTGGCCTGCGCGGCCGTG
GCCACCAGCATGATCCTGCTGCAGGCCCAGATCGACTACCTGCCCATCTGGCTGCTGCTG
CGCCTGGTCTCGGGCATCATGATGGTGACCGAATTCATGGTCATCGAAAGCTGGCTCAAC
GAACAAACCGAAAACCGCCAGCGCGGCCGCGTATTCTCGGTGTACATGGTGGTCTCCGGC
CTGGGCACGGTGCTGGGACAGCTGGCGCTCACGCTCTACGGCGCGCTGGACGACGGGCCG
CTCATCCTGGTGGCCATGTGCCTGGTCCTGTGCCTGGTGCCCATCGCCGTGACGGCGCGC
TCGCACCCGCCCACGCCGCGTCCGGCGCCGCTGGACTTCTTCTTTTTCGTCAAGCGCGTG
CCGCTGGCCATGACGGTCCTGTTCGTGGCCGGCAACCTGAGTGGCGCCTTCTACGGGCTG
GCCCCGGTCTATGCCGCCAAGCATGGCCTGCAGACTTCCCAGGTGGCCTTGTTCGTCGCC
GTGTCCGTCACCGCCGGCCTGCTGTCGCAATGGCCCATCGGCTGGCTGTCCGACCGCGTC
AATCGCGCCGGCCTGATCCGTTTAACGCCGCCGTGCTGGTGCTGCTGCCCACGCTGATGT
GGGGCTGGCTGGACCTGCCTTTCTGGCTGCTGCTCTGCCTCTCGGCGCTGCTGGGCGTGC
TGCAGTTCACCCTCTATCCGCTGGGCGCGGCCCTGGCCAATGACCATGTGGAGGCCGAGC
GCCGGGTGAGCCTGAGCGCCGTGCTGCTGATGGTCTACGGGGTGGGCGCCTGCCTGGGCC
CGCTGGTCGCCGGCATCCTCATGTCGCTCGGCGGGCACGCCATGTACTACGTCTTCGTGC
CGGCCTGCGCCCTTATCCTGGTCTGGCGCGTGCGGCCCAGCGCCGTCACTGGCGTGCACC
AGGTCGAGGAGGCGCCGGTGCAATTCGTGCCCATGCCCGACACGCTGCAGTCCTCGCCCG
CCATGGTGGCCTTGGATCCCCGTGTGGATCCCGAGGTGGACCCGGCCATGGAGATGGTCA
CGCCCGAGGCCGGCGTGGTGCAGCCGCCGCCGCCGGCCGCCGAACCCGCTGCCGGCACGG
CGGCCTTCGACAACGTCGTGGCCGAGCCGGGCGAGCCGGCCACCGTCCTGTCCGCAGACG
GCGCGCCGAGTCCGCGCACAGGGACGGACGCCTGA
```

How many nucleotides?
Easiest solution:

```
cat Bhinzii5132_BB1335_consensus.fa \
| grep -v “>” | tr -d “\n” | wc -L

# output: 1415
# grep -v “>” - select lines that do not contain “>”
# → only sequence without fasta header
# tr -d “\n” - translate carriage return “\n” to nothing
# → concatenates all sequence lines
# wc - word count
# wc -L returns number of characters in longest line
# Result: 1415
# That means, we are dealing with the frameshift gene variant
```