2nd Guest Lecture

Bodo Linz

09/25/18

bodo.linz@uga.edu

Today's lecture:

1. ACT – Artemis Comparison Tool

2. PCA – Principal Component Analysis

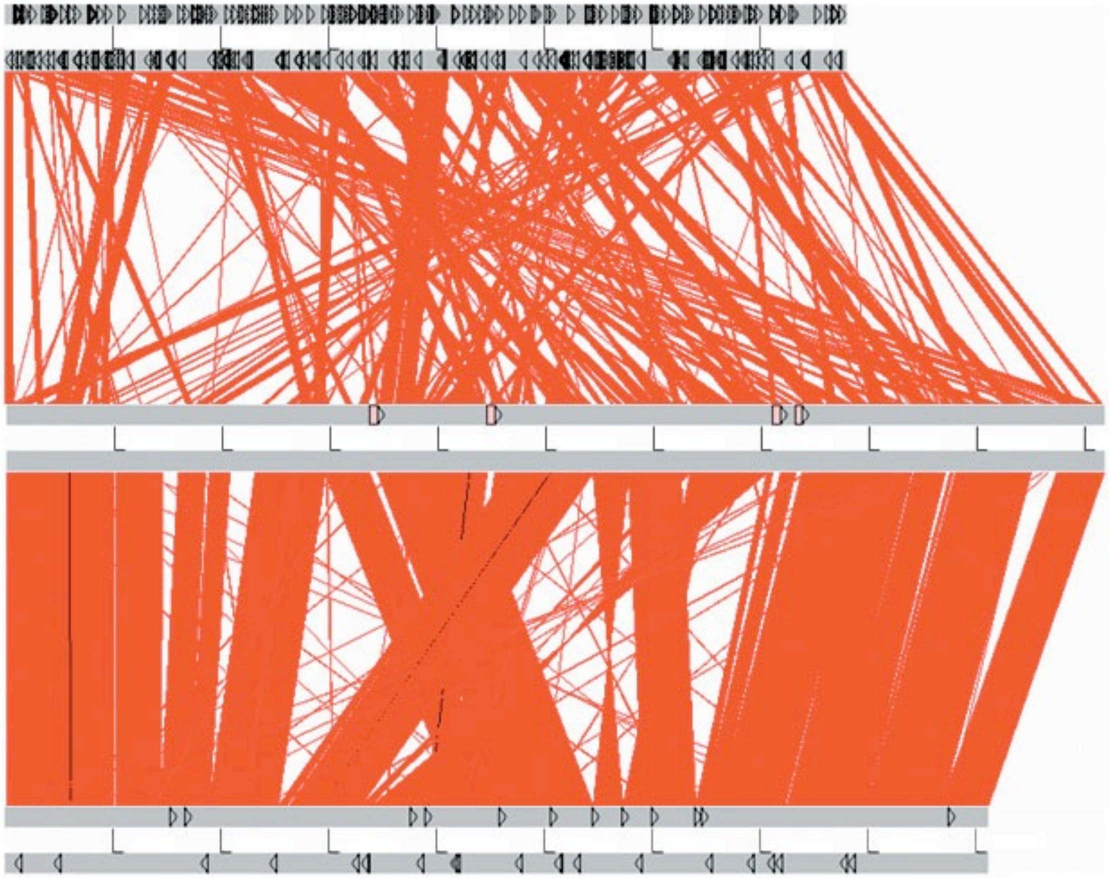
3. Download a Short Read Archive (SRA) from NCBI

4. lastz and YASRA – Yet Another Short Read Assembler

Let’s continue with ACT

We learned how to perform a pairwise genome comparison:

1) at the internet (`double_ACT`)
2) run locally using `blastall` and `MSPcrunch`

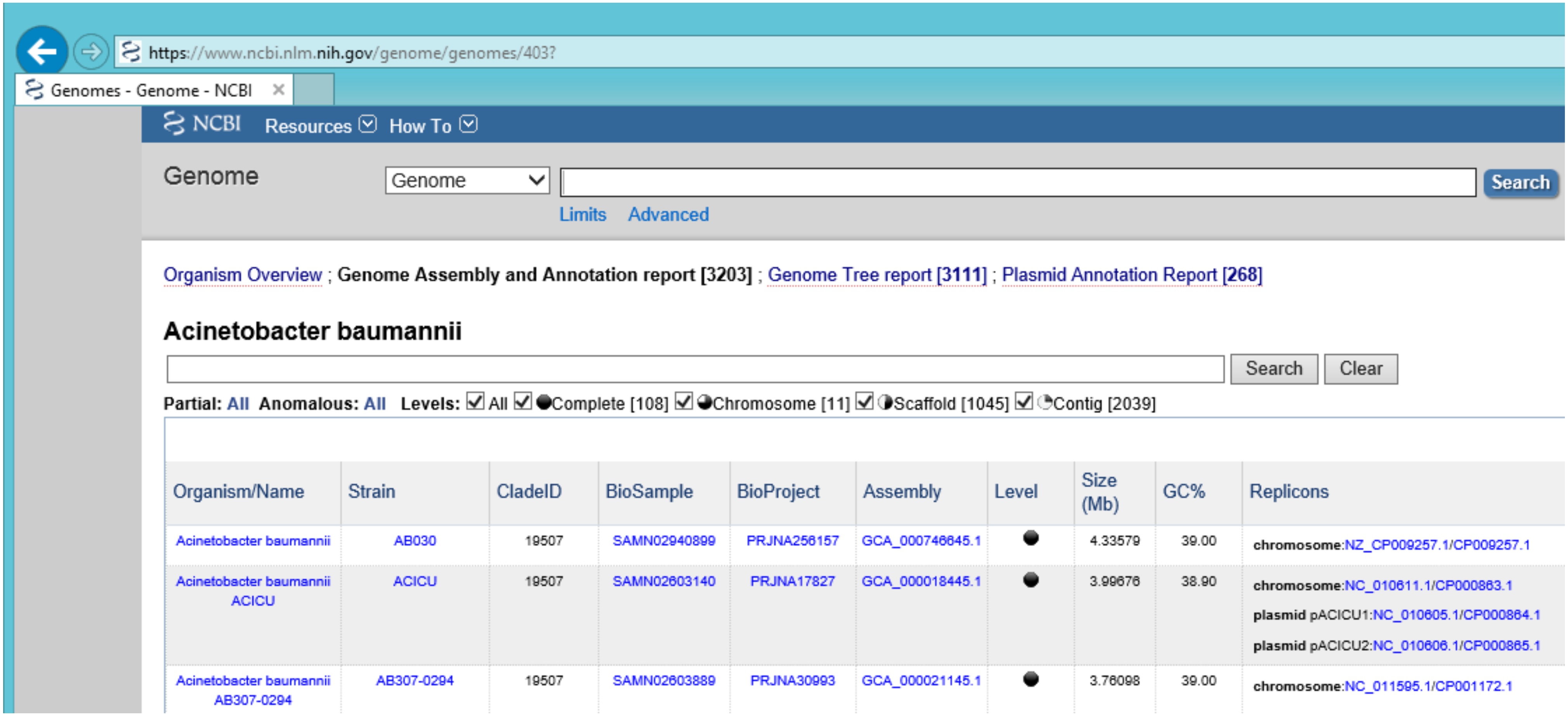


works well for completed genomes

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`

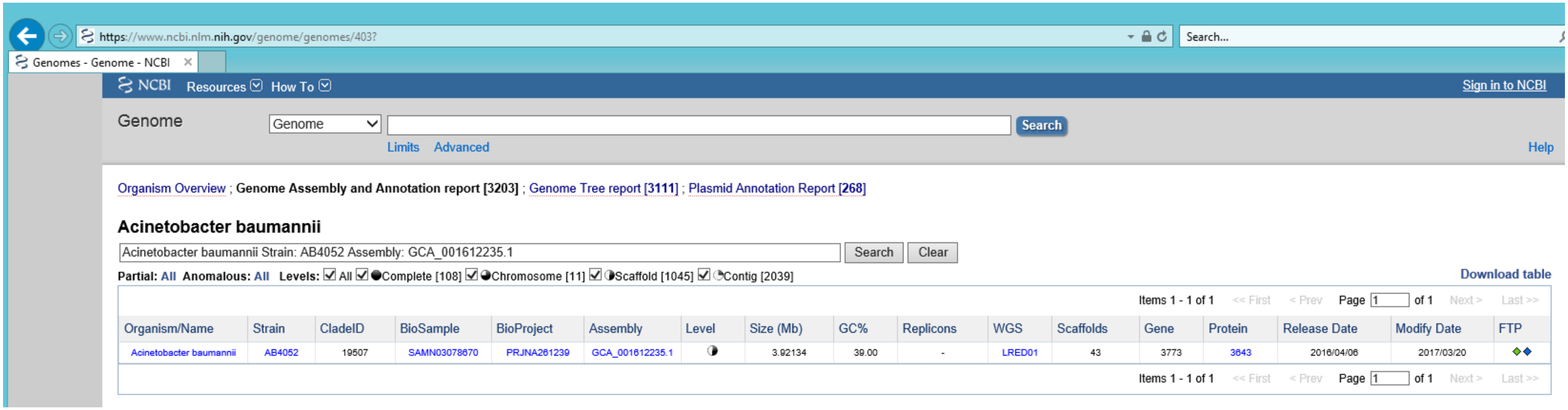
go to https://www.ncbi.nlm.nih.gov/genome/

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	◆◆

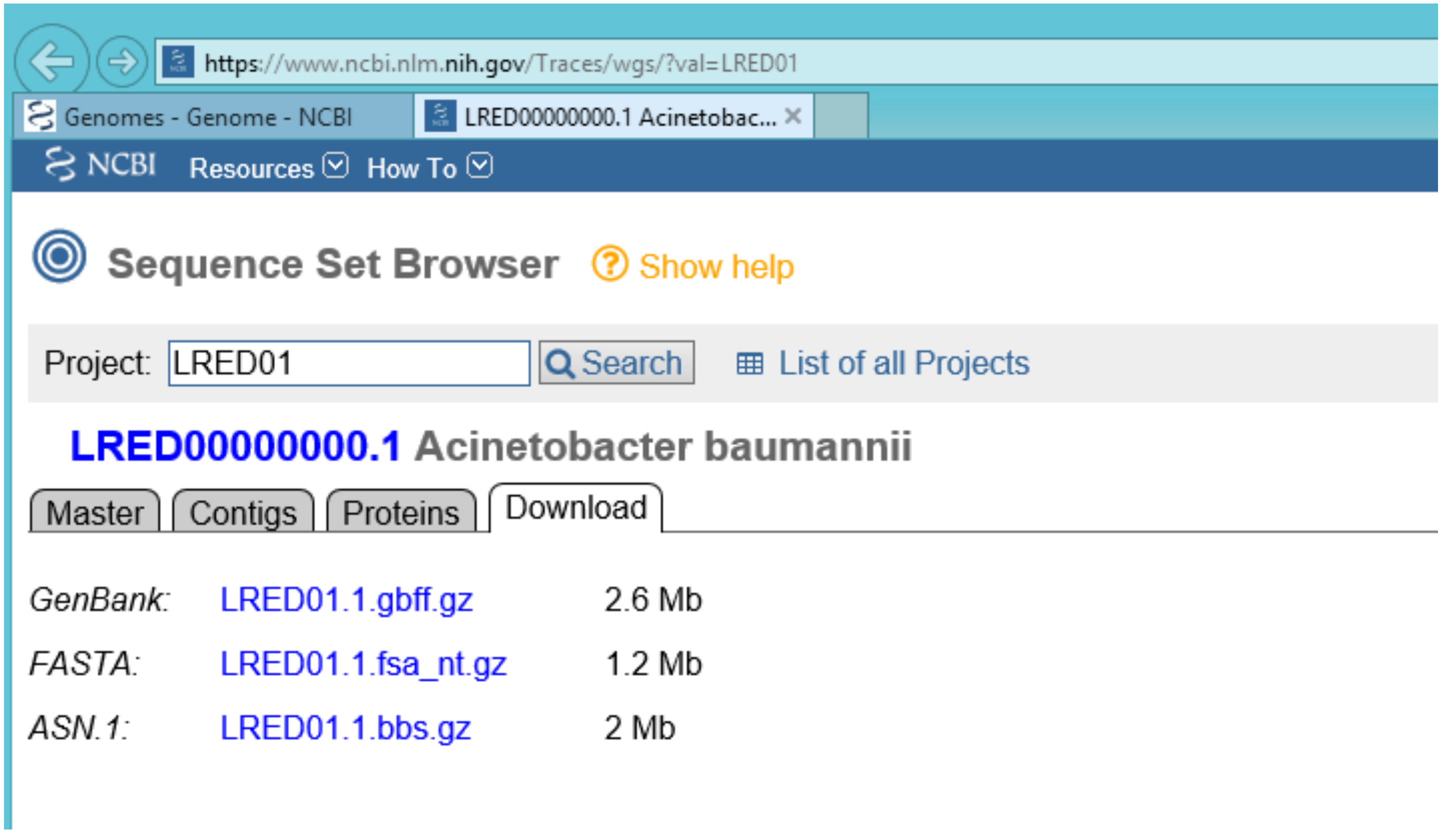
Items 1 - 1 of 1 << First < Prev Page 1 of 1 Next > Last >>

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`

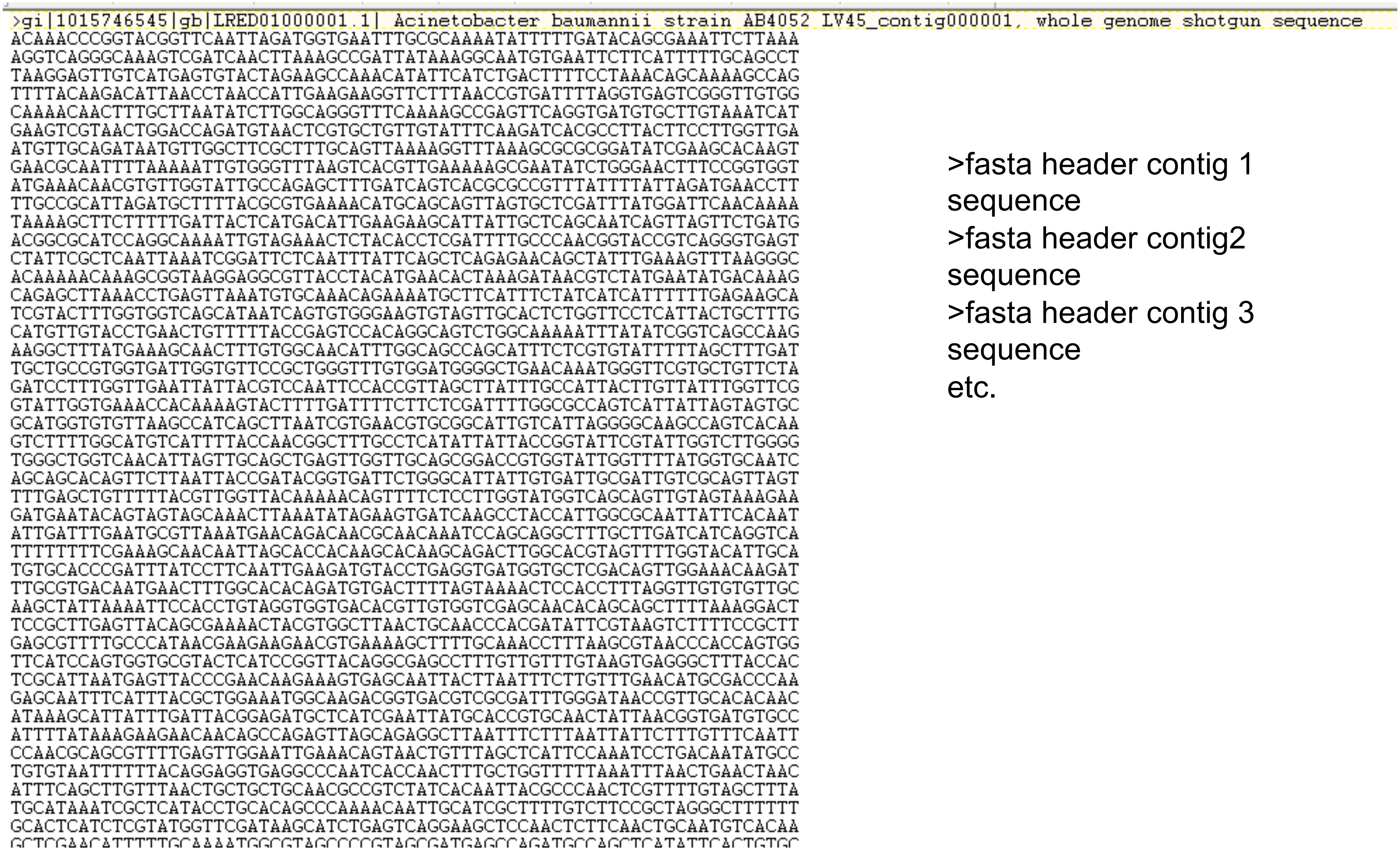


Project: LRED01

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

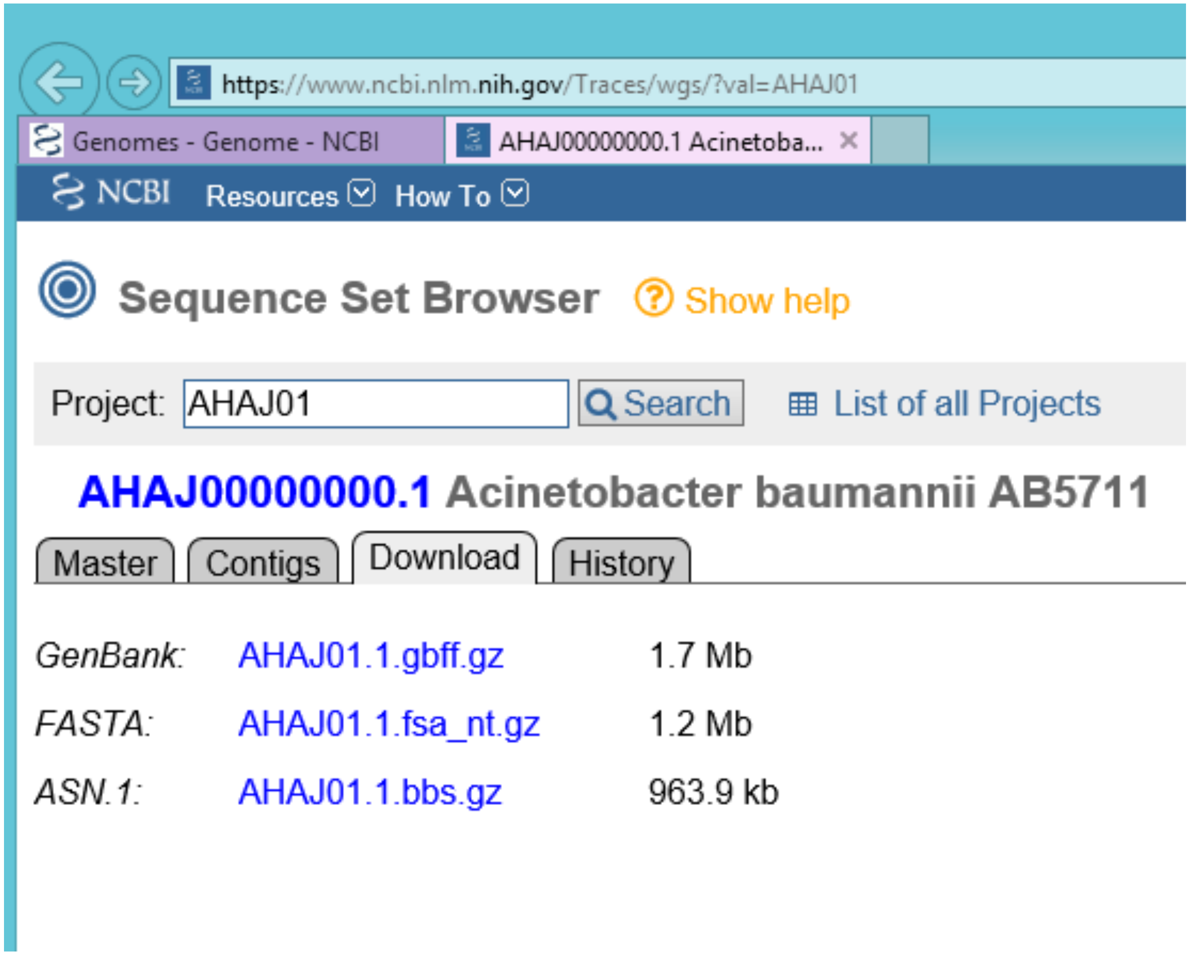


```
>gi|1015746545|gb|LRED01000001.1| Acinetobacter baumannii strain AB4052 LV45_contig000001, whole genome shotgun sequence
ACAAACCCGGTACGGTTCAATTAGATGGTGAATTTGCGCAAAATATTTTTGATACAGCGAAATTCTTAAA
AGGTCAGGGCAAAGTCGATCAACTTAAAGCCGATTATAAAGGCAATGTGAATTCTTCATTTTTGCAGCCT
TAAGGAGTTGTCATGAGTGTACTAGAAGCCAAACATATTCATCTGACTTTTCCTAAACAGCAAAAGCCAG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

do the same for strain AB5711



https://www.ncbi.nlm.nih.gov/Traces/wgs/?val=AHAJ01
Genomes - Genome - NCBI
AHAJ00000000.1 Acinetoba...
NCBI
Resources
How To

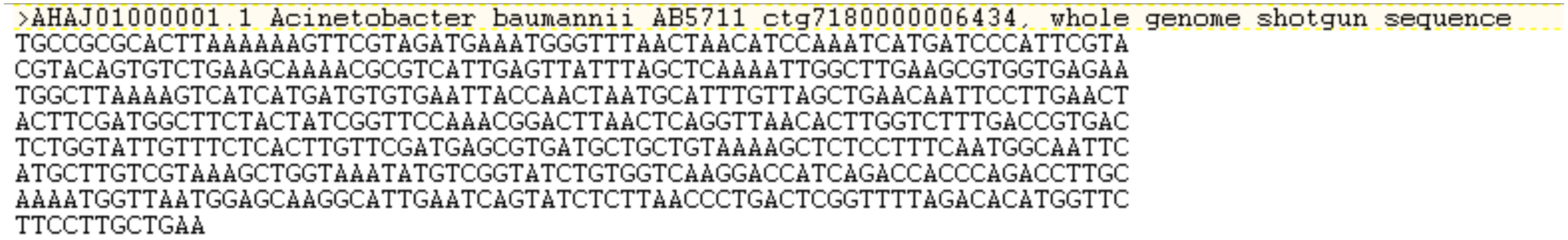
Sequence Set Browser

Project: AHAJ01

AHAJ00000000.1 Acinetobacter baumannii AB5711

GenBank:	AHAJ01.1.gbff.gz	1.7 Mb
FASTA:	AHAJ01.1.fsa_nt.gz	1.2 Mb
ASN.1:	AHAJ01.1.bbs.gz	963.9 kb

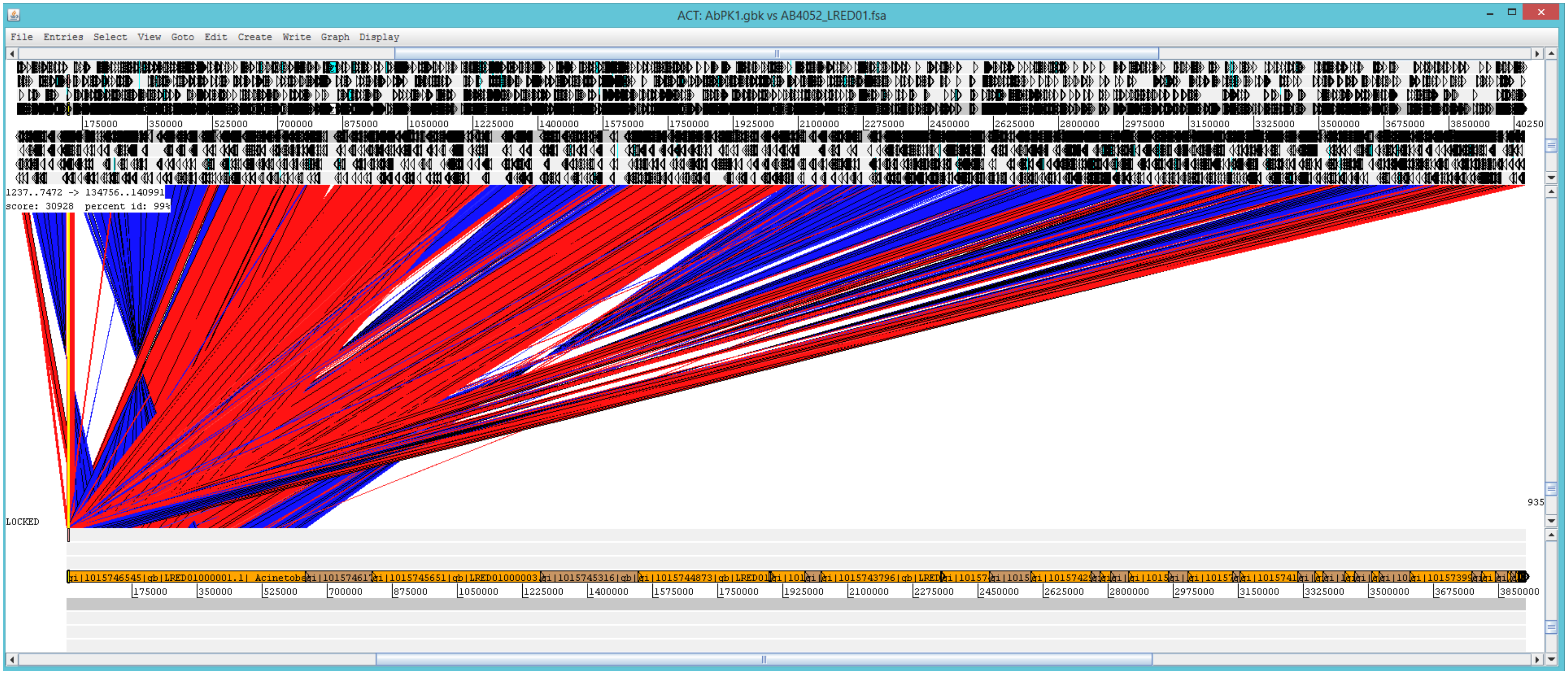
We get



```
>AHAJ01000001.1 Acinetobacter baumannii AB5711 ctg7180000006434, whole genome shotgun sequence
TGCCGCGCACTTAAAAAAGTTCGTAGATGAAATGGGTTTAACTAACATCCAAATCATGATCCCATTCGTA
CGTACAGTGTCTGAAGCAAAACGCGTCATTGAGTTATTTAGCTCAAAATTGGCTTGAAGCGTGGTGAGAA
TGGCTTAAAAGTCATCATGATGTGTGAATTACCAACTAATGCATTTGTTAGCTGAACAATTCCTTGAACT
ACTTCGATGGCTTCTACTATCGGTTCCAAACGGACTTAACTCAGGTTAACACTTGGTCTTTGACCGTGAC
TCTGGTATTGTTTCTCACTTGTTCGATGAGCGTGATGCTGCTGTAAAAGCTCTCCTTTCAATGGCAATTC
ATGCTTGTCGTAAAGCTGGTAAATATGTCGGTATCTGTGGTCAAGGACCATCAGACCACCCAGACCTTGC
AAAATGGTTAATGGAGCAAGGCATTGAATCAGTATCTCTTAACCCTGACTCGGTTTTAGACACATGGTTC
TTCCTTGCTGAA
```

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

This is what we get:

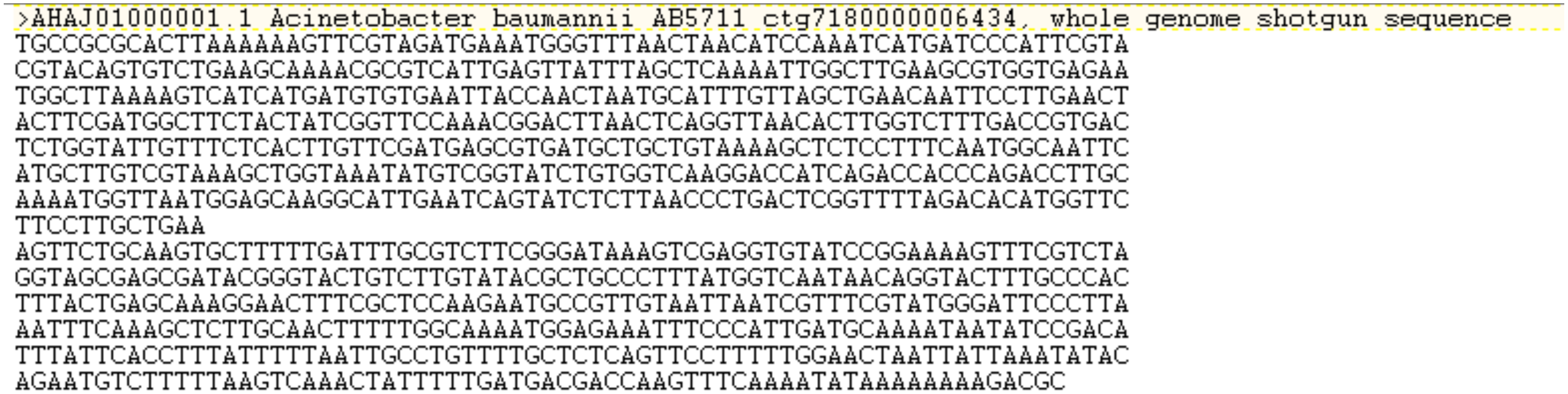


```
1237..7472 -> 134756..140991
score: 30928  percent id: 99%
```

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
AGAATGTCTTTTTAAGTCAAACTATTTTTGATGACGACCAAGTTTCAAAATATAAAAAAAAAGACGC
```

```
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 will 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
```

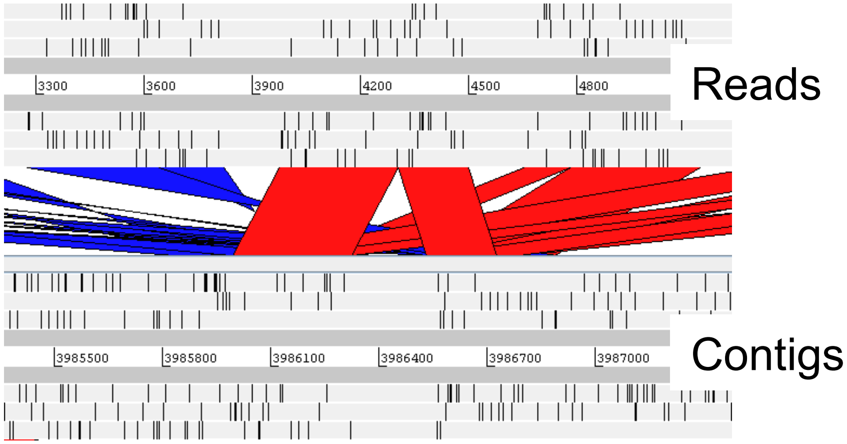
```
# substr: substring
# if $1 at position 1 for 1 character = ">", print nothing
# else print
# printf "%s" - take the first of the following arguments ($1) and
print it as a string (s), "%d" - as a number (decimal)
# then print "\n"
# >> add to file AHAJ01.fa
```

Solution: modify the genome format

- What we get: very simple, One fasta header, followed by sequence
- Can be used for genome comparison (blastall and MSPcrunch)
- useful for having a quick look
- e.g. make comparison to find a gene in target genome
- what if: want to keep tract of contig numbers to
  - order and orient contigs based on reference genome
  - close the genome

- concatenate contigs into a single supercontig

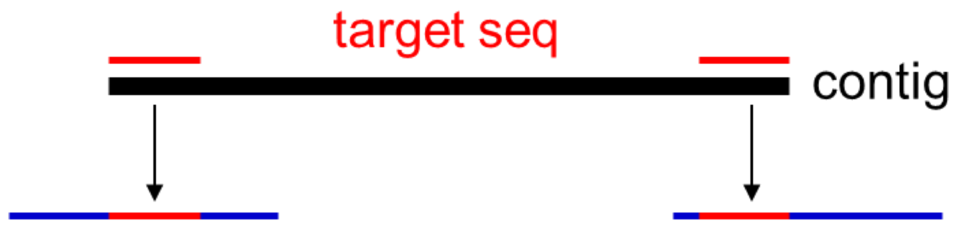
- want to keep contig numbers



Closing the genome

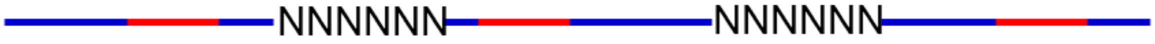
165 contigs

target sequences near both ends of the contig



extract seq reads containing the target sequence

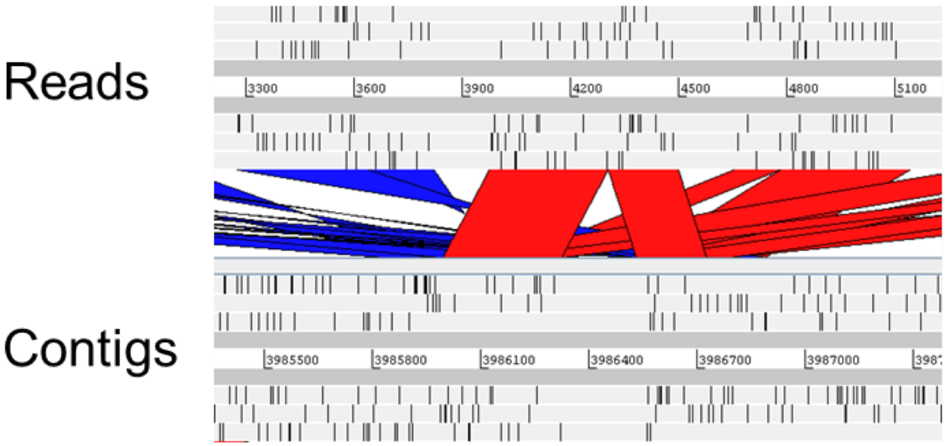
concatenate extracted reads



concatenate contigs

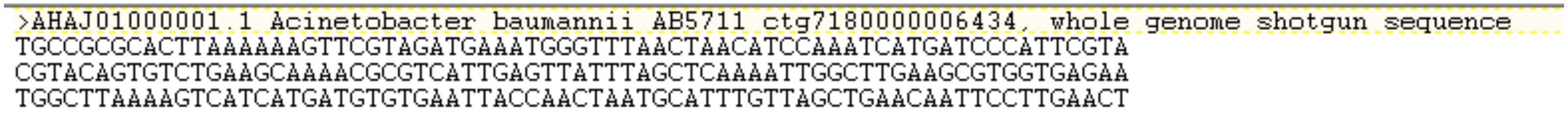
run genome comparison

visualize in ACT

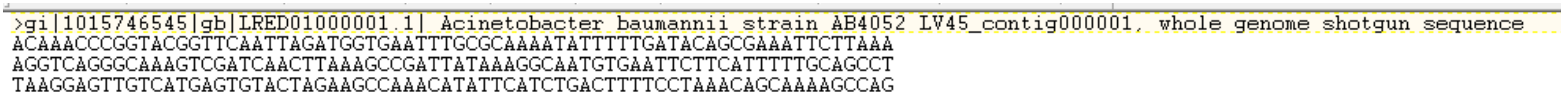


Modify the genome format

- two different formats of genome headers: make same format



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



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

- problem: ACT doesn’t take numbers
- solution: letter code for contig numbers, IUPAC
- start the contig with a 5 letter contig code
- followed by nn to separate from sequence
- separate contigs by stretches of N’s (~300)

e.g. …NNNNNAAAGTnnACGTATGCAT…

IUPAC

0	–	A
1	–	G
2	–	C
3	–	T
4	–	R (G or A)
5	–	Y (C or T)
6	–	M (A or C)
7	–	K (G or T)
8	–	S (G or C)
9	–	W (A or T)

Solution: modify the genome format

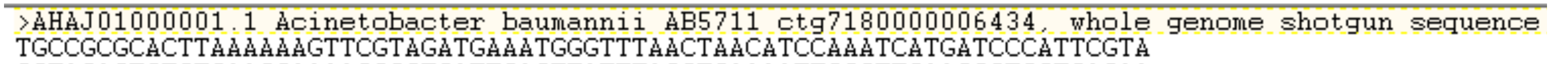
```
#!/bin/bash
# runContigstoACT.sh
# Author Bodo Linz
# run blast of *.fna or *.fsa file in the current directory
# against a specified reference sequence (database)
# generate the *.cmp file for ACT


BLASTALL=~/bin/blastall
MSPCRUNCH=~/bin/MSPcrunch
DATABASE=AbPK1.fasta
NAME1=${DATABASE%%".fasta"}
GENOME2=AB4052_LRED01.fsa
NAME2=${GENOME2%%"_LRED01.fsa"}
```

Completed reference genome

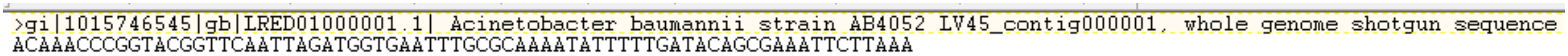
Target genome as contigs

```
# Based on the previous lecture:
# What is NAME1?
# What is NAME2?
```

Note the different headers



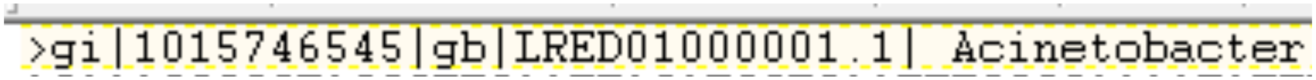
```
>AHAJ01000001.1 Acinetobacter baumannii AB5711 ctg7180000006434, whole genome shotgun sequence
TGCCGCGCACTTAAAAAAGTTCGTAGATGAAATGGGTTTAACTAACATCCAAATCATGATCCCATTCGTA
```



```
>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"
       }
      }' \
> tempgenome.fsa
```

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”
       }
      }' \
>tempgenome.fsa
```

```
We Get: >LRED01000001.1
        >AHAJ01000001.1 Acinetobacter baumannii AB5711 ctg7180000…
        → We took care of the different headers
```

```
# generate one large contig fasta, contigs separated by N's
printf ">"$NAME2" Contigs\n" > $NAME2.fa

cat tempgenome.fsa | awk -v FS="\n" -v OFS="" '{print $1}' | awk -v
FS=" " -v OFS="\t" '{print $1}' \
 | tr "0" "A" | tr "1" "G" | tr "2" "C" | tr "3" "T" | tr "4" "R" | tr
"5" "Y" | tr "6" "M" | tr "7" "K" | tr "8" "S" | tr "9" "W" \
 | awk '{
       if(substr($1,1,1) == ">"){
printf"NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN";
               printf substr($1,9,5);
               printf"nn";
        }else{
               printf"%s",$1;
        }
     } END{printf"\n"}' \
>> $NAME2.fa
```

Let’s walk through

```
printf ">"$NAME2" Contigs\n" > $NAME2.fa
# print “>” and $NAME2 (=AB4052_LRED01) Contigs followed by “\n”
# and save this as file fake
# >AB4052_LRED01 Contigs

cat tempgenome.fsa | awk -v FS="\n" -v OFS="" '{print $1}' | awk -v
FS=" " -v OFS="\t" '{print $1}' \
 | tr "0" "A" | tr "1" "G" | tr "2" "C" | tr "3" "T" | tr "4" "R" | tr
"5" "Y" | tr "6" "M" | tr "7" "K" | tr "8" "S" | tr "9" "W" \
# FS=" " -v OFS="\t" replaces space in header by tab
# tr – translate/transliterate, replace or remove specific characters
# syntax: tr “what to search for” “what to replace with”
# here: replace the numbers by IUPAC letters
```

Let’s walk through

```
awk '{
        if(substr($1,1,1) == ">"){
# if the $1 at character 1 equals “>”
printf"NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN";
# then print a stretch of N’s
               printf substr($1,9,5);
               printf"nn";
# print 5 character substring starting at character 9: >LRAD01000001.1
                                                       >LRADAGAAAAAG.G
         }else{
               printf"%s",$1;
         }
     } END{printf"\n"}' \
# else print everything on the line, add a “\n” at the end
>> $NAME2.fa
# attach (add) everything to file $NAME2.fa (AB4052.fa)
```

```
echo ""
echo "Done generating the contig file"
echo "------------------------------------------"
echo ""


# 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

# if -f(ile) ${DATABASE}.nhr -a(nd) ${DATABASE}.nin etc. exist
# then display "The database is already formatted"
# else run formatdb
```

Then run blastall and MSPcrunch as before (see last lecture)

128 Bordetella genomes

95 classical bordetellae:

- 58 *B. bronchiseptica*
- 2 *B. parapertussis*
- 34 *B. pertussis*

respiratory pathogens in
animals and humans

34 non-classical bordetellae:

- 18 *B. holmesii*
- 6 *B. hinzii*
- 1 *B. avium*

respiratory pathogens in animals and
in immuno-compromized humans

- 4 *B. trematum*
- 2 *B. ansorpii*

wound and ear infection in humans

- 3 *B. petrii*

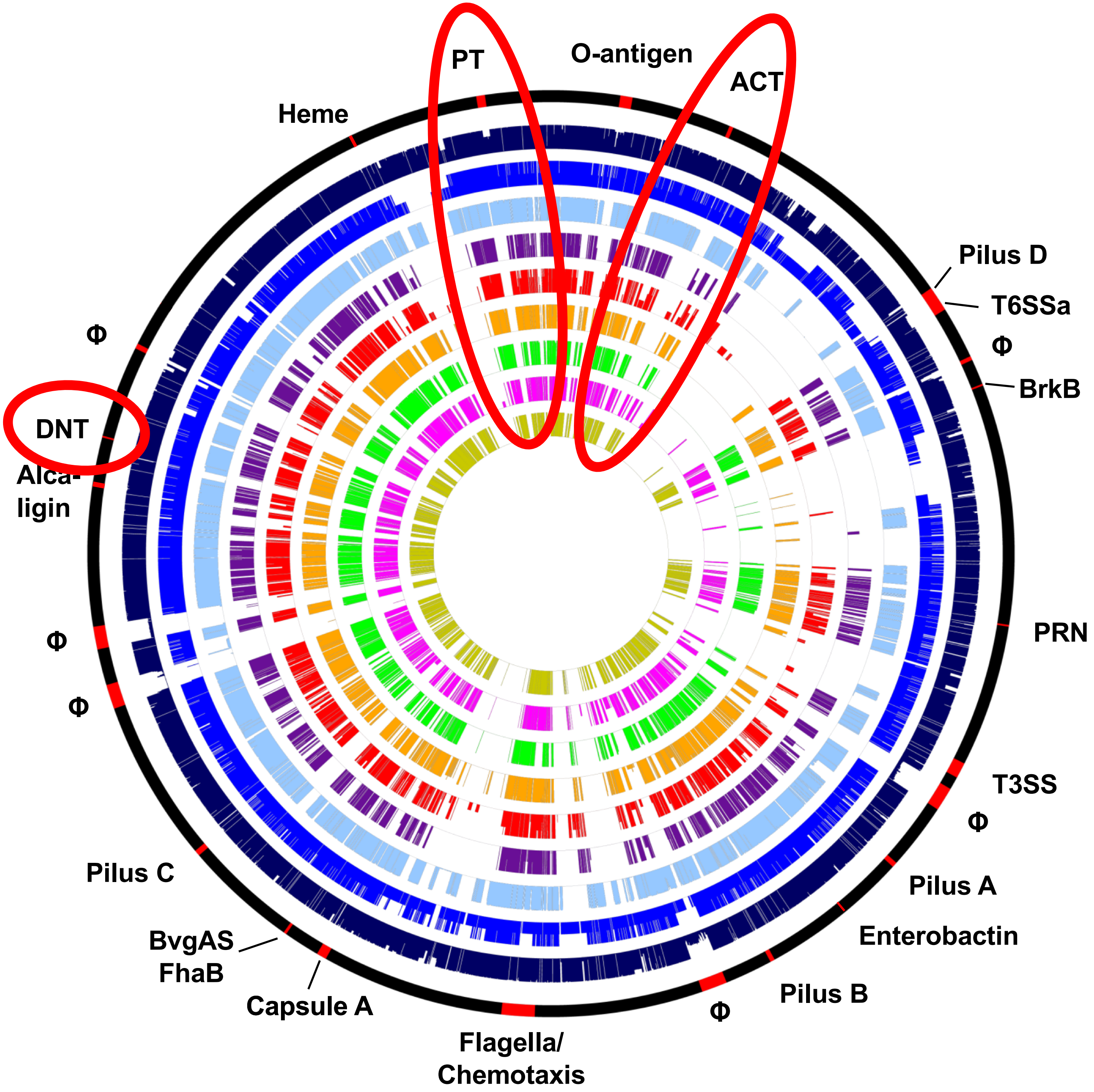
environmental / ear infection in humans

questions

- virulence-associated factors determining host specificity?
- virulence-associated factors determining disease outcome?

Approach

- genome-wide SNP-based phylogenetic tree
- genome-wide presence/absence of genes
  - similar evolutionary trends?
- Pairwise genome comparisons (ACT)
  (Artemis Comparison Tool)
- mapping of virulence-associated genes
- Principle Components Analysis (PCA)



Circles

1: Virtual chromosome of *B. bronchiseptica* RB50 with genes of interest;

2: *B. bronchiseptica* (based on 58 genomes);

3: *B. parapertussis* (2);

4: *B. pertussis* (34);

5: *B. ansorpii* (2);

6: *B. petrii* (3);

7: *B. hinzii* (6);

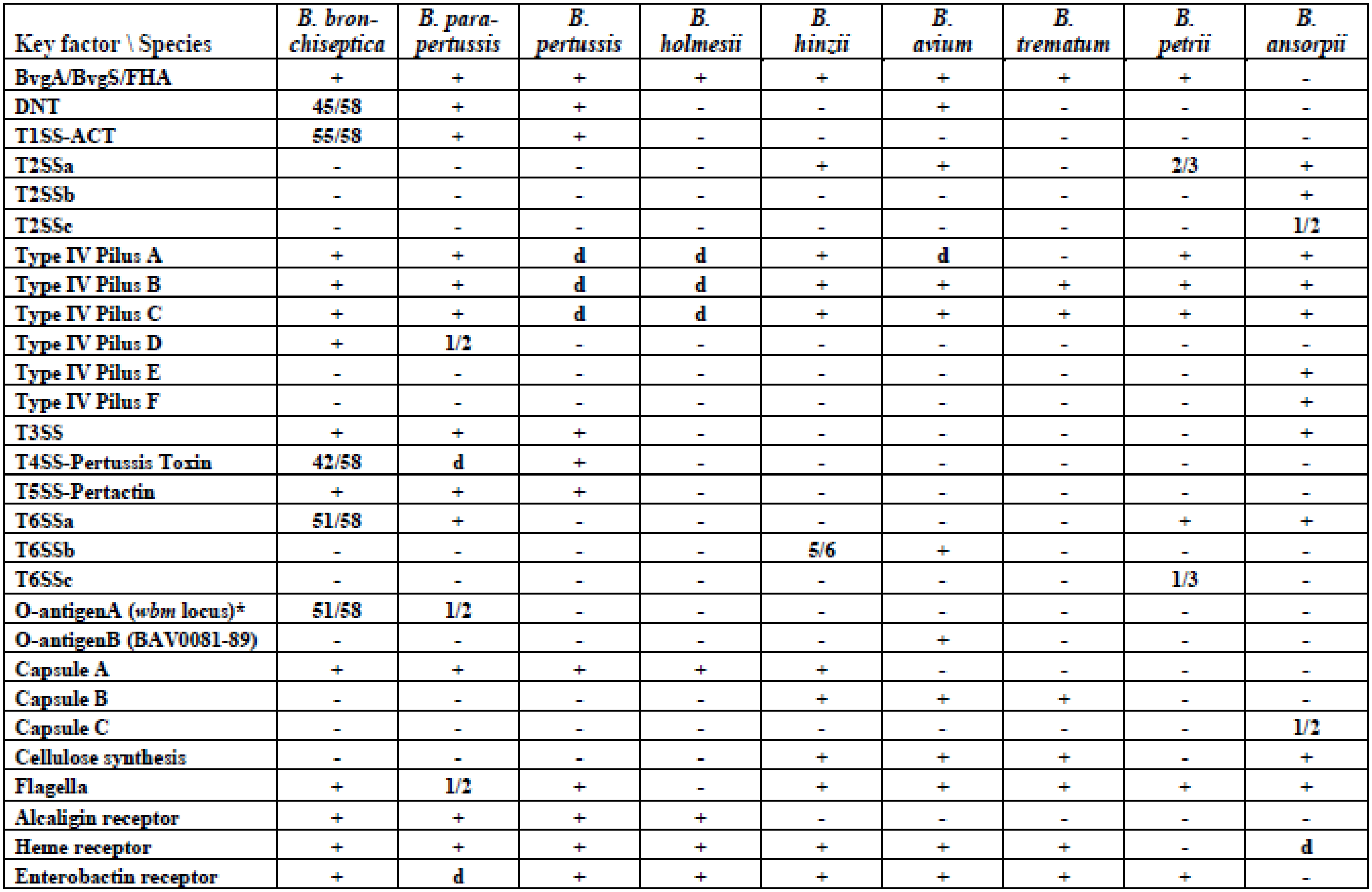
8: *B. holmesii* (18);

9: *B. trematum* (4);

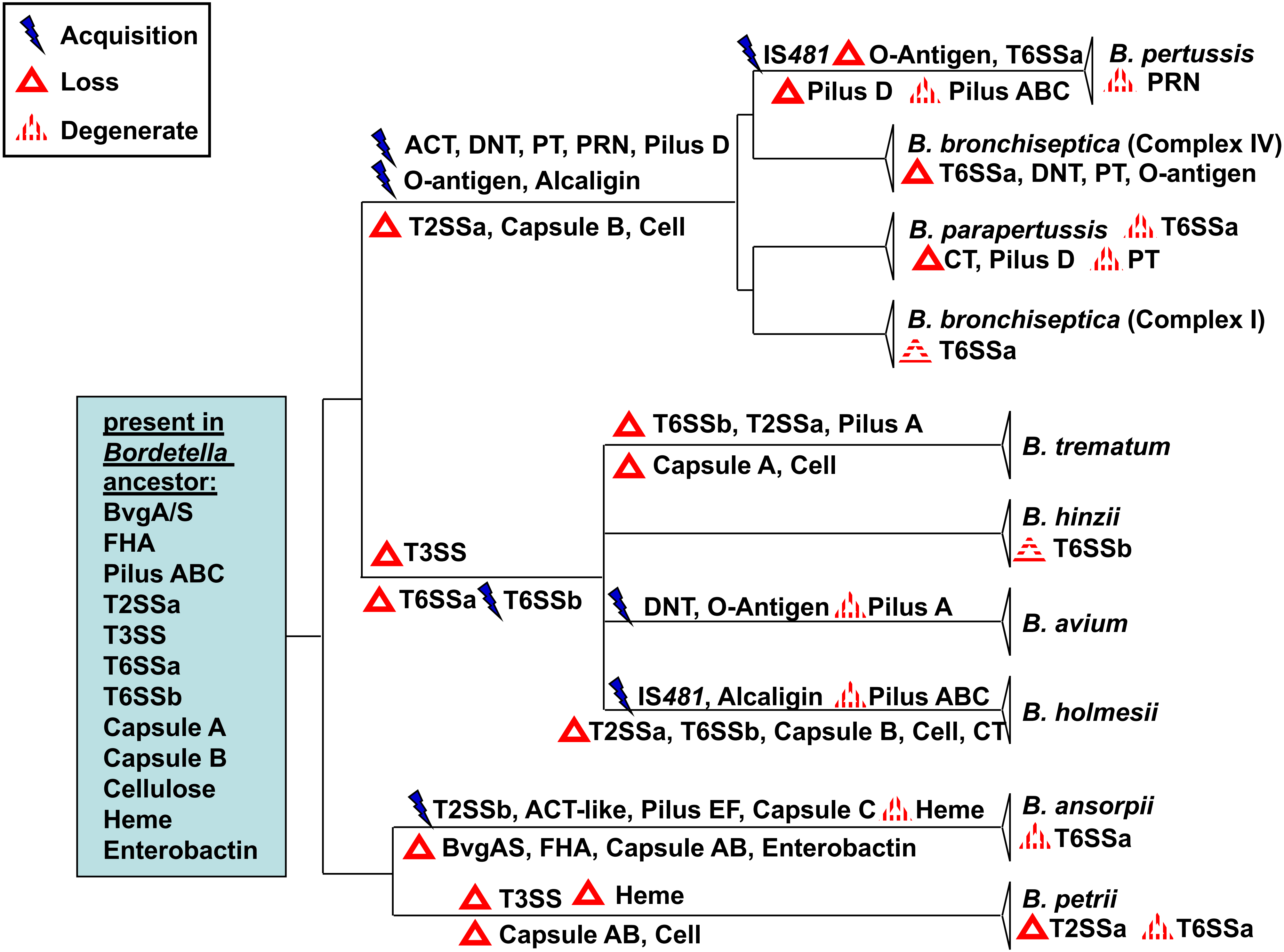
10: *B. avium* (1)

Non-classical species lack toxins of the classical species:
Pertussis Toxin (PT)
ACT
DNT

Presence and absence of virulence-associated key factors



Key factor \ Species	*B. bronchiseptica*	*B. parapertussis*	*B. pertussis*	*B. holmesii*	*B. hinzii*	*B. avium*	*B. trematum*	*B. petrii*	*B. ansorpii*
BvgA/BvgS/FHA	+	+	+	+	+	+	+	+	-
DNT	**45/58**	+	+	-	-	+	-	-	-
T1SS-ACT	**55/58**	+	+	-	-	-	-	-	-
T2SSa	-	-	-	-	+	+	-	**2/3**	+
T2SSb	-	-	-	-	-	-	-	-	+
T2SSc	-	-	-	-	-	-	-	-	**1/2**
Type IV Pilus A	+	+	**d**	**d**	+	**d**	-	+	+
Type IV Pilus B	+	+	**d**	**d**	+	+	+	+	+
Type IV Pilus C	+	+	**d**	**d**	+	+	+	+	+
Type IV Pilus D	+	**1/2**	-	-	-	-	-	-	-
Type IV Pilus E	-	-	-	-	-	-	-	-	+
Type IV Pilus F	-	-	-	-	-	-	-	-	+
T3SS	+	+	+	-	-	-	-	-	+
T4SS-Pertussis Toxin	**42/58**	**d**	+	-	-	-	-	-	-
T5SS-Pertactin	+	+	+	-	-	-	-	-	-
T6SSa	**51/58**	+	-	-	-	-	-	+	+
T6SSb	-	-	-	-	**5/6**	+	-	-	-
T6SSc	-	-	-	-	-	-	-	**1/3**	-
O-antigenA (*wbm* locus)*	**51/58**	**1/2**	-	-	-	-	-	-	-
O-antigenB (BAV0081-89)	-	-	-	-	-	+	-	-	-
Capsule A	+	+	+	+	+	-	-	-	-
Capsule B	-	-	-	-	+	+	+	-	-
Capsule C	-	-	-	-	-	-	-	-	**1/2**
Cellulose synthesis	-	-	-	-	+	+	+	-	+
Flagella	+	**1/2**	+	-	+	+	+	+	+
Alcaligin receptor	+	+	+	+	-	-	-	-	-
Heme receptor	+	+	+	+	+	+	+	-	**d**
Enterobactin receptor	+	**d**	+	+	+	+	+	+	-



⚡ Acquisition
△ Loss
◬ Degenerate

present in *Bordetella* ancestor:
BvgA/S
FHA
Pilus ABC
T2SSa
T3SS
T6SSa
T6SSb
Capsule A
Capsule B
Cellulose
Heme
Enterobactin

⚡ ACT, DNT, PT, PRN, Pilus D
⚡ O-antigen, Alcaligin
△ T2SSa, Capsule B, Cell

⚡ IS*481* △ O-Antigen, T6SSa
△ Pilus D ◬ Pilus ABC
B. pertussis ◬ PRN

B. bronchiseptica (Complex IV)
△ T6SSa, DNT, PT, O-antigen

B. parapertussis ◬ T6SSa
△CT, Pilus D ◬ PT

B. bronchiseptica (Complex I)
◭ T6SSa

△T3SS
△T6SSa ⚡ T6SSb

△ T6SSb, T2SSa, Pilus A
△ Capsule A, Cell
B. trematum

B. hinzii
◭ T6SSb

⚡ DNT, O-Antigen ◬Pilus A
B. avium

⚡ IS*481*, Alcaligin ◬Pilus ABC
△T2SSa, T6SSb, Capsule B, Cell, CT
B. holmesii

⚡T2SSb, ACT-like, Pilus EF, Capsule C ◬ Heme
△ BvgAS, FHA, Capsule AB, Enterobactin
B. ansorpii
◬ T6SSa

△ T3SS △ Heme
△ Capsule AB, Cell
B. petrii
△T2SSa ◬ T6SSa

Presence and absence of virulence-associated key factors:

Are there similarities or trends to explain:

- host spectrum?
- infected organs?
- disease outcome?

Principal Component Analysis (PCA)

- invented in 1901 by Karl Pearson
- statistical procedure that converts a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components (PCs)
- Principal Components are the underlying structure in the data
- PCA mostly used as a tool in exploratory data analysis
- it reveals the internal structure of the data
- in a way that best explains the variance in the data
- PC1 has the largest possible variance
    - accounts for as much of the variability in the data as possible
- PC2 second largest variance in the data
- PC3 third largest
- resulting PCs are uncorrelated

Input

- based on numbers
- change nucleotides to allele numbers (e.g. A=1, C=2, G=3, T=4)
- here presence and absence of genes as 1 and 0
- computation in R using libraries `gplots`, `gdata`, and `gtools`

Species/factor	BvgAS	DNT	ACT	T2SSa	T2SSb	T2SSc	PilA	PilB	PilC	PilD	PilE	PilF	T3SS	PT	PRN	T6SSa	T6SSb
B.bronch1	1	1	1	0	0	0	1	1	1	1	0	0	1	1	1	1	0
B.bronch2	1	1	0	0	0	0	1	1	1	1	0	0	1	1	1	1	0
B.bronch3	1	0	0	0	0	0	1	1	1	1	0	0	1	0	1	1	0
B.bronch4	1	0	0	0	0	0	1	1	1	1	0	0	1	0	1	0	0
B.bronch5	1	0	0	0	0	0	1	1	1	1	0	0	1	0	1	1	0
B.bronch6	1	1	1	0	0	0	1	1	1	1	0	0	1	0	1	0	0
B.bronch7	1	0	1	0	0	0	1	1	1	1	0	0	1	1	1	0	0
B.bronch8	1	1	1	0	0	0	1	1	1	1	0	0	1	0	1	1	0
B.parahu	1	1	1	0	0	0	1	1	1	1	0	0	1	0	1	1	0
B.paraov	1	1	1	0	0	0	1	1	1	1	0	0	1	0	1	1	0
B.pertussis1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0
B.pertussis2	1	1	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0
B.holmesii	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B.hinzii1	1	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	1
B.hinzii2	1	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0
B.avium197N	1	1	0	1	0	0	1	1	1	0	0	0	0	0	0	0	1
B.trematum	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
B.petriiJ49	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	1	0
B.petriiJ51	1	0	0	1	0	0	1	1	1	0	0	0	0	0	0	1	0
B.petriiDSM	1	0	0	1	0	0	1	1	1	0	0	0	0	0	0	1	0
B.ansorpii1	0	0	0	1	1	1	1	1	1	0	1	1	1	0	0	1	0
B.ansorpii2	0	0	0	1	1	1	1	1	1	0	1	1	1	0	0	1	0

# computation of PCA

```
library(gplots)

library(gdata)

library(gtools)


rm(list = ls())

g<-as.matrix(read.table("D:/Data/Virulence.txt", 
row.names=1,header=TRUE,check.names=TRUE, sep = "\t") )

h <- as.matrix(dist(g))

print(summary(pc<- princomp(h, cor=T)))

pc$loadings

pc$scores

ghi1 <- as.table(pc$scores)

ghi2 <- as.table(pc$loadings)

write.table(ghi1, file="D:/Data/PCA_scores.txt", sep="\t", 
row.names=T, col.names=T)

write.table(ghi2, file="D:/Data/PCA_loadings.txt", sep="\t", 
row.names=T, col.names=T)
```

# Let’s walk through:

```
library(gplots) # load library (gplots)

library(gdata)  # load library (gdata)

library(gtools) # load library (gtools)


rm(list = ls())  # empty memory, optional

g<-as.matrix(read.table("D:/Data/Virulence.txt",
row.names=1,header=TRUE,check.names=TRUE, sep = "\t") )

# read table "D:/Data/Virulence.txt" in matrix format into file "g"

# row.names=1  - table has 1 row name

(you can have several such as strain, year, country, etc)

# header=TRUE,check.names=TRUE - table has headers, check that
column headers are unique

# sep = “\t” -  columns are separated by tab

h <- as.matrix(dist(g))

# make distance matrix of file g
```

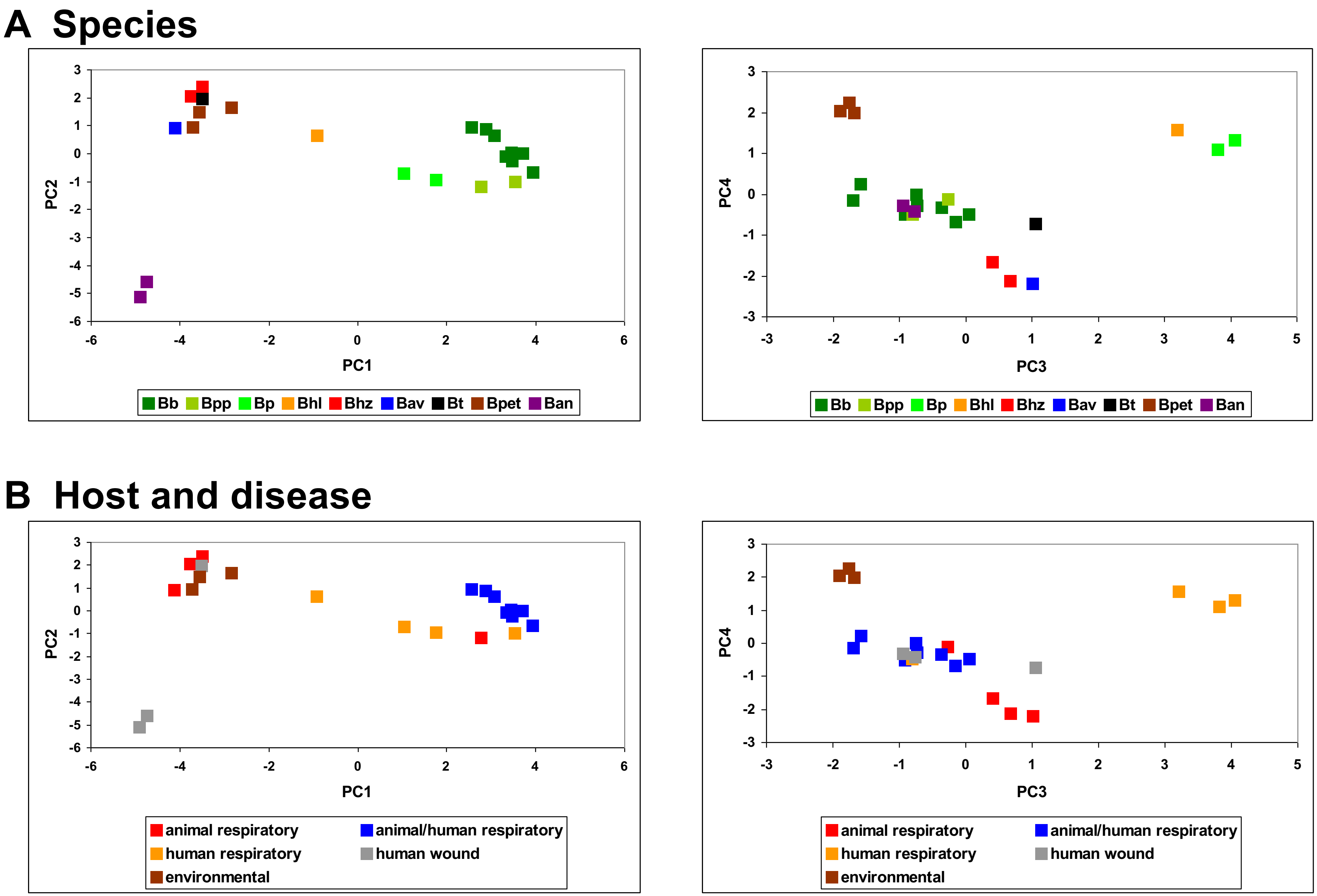
# Let’s walk through:

```
print(summary(pc<- princomp(h, cor=T)))

pc$loadings

pc$scores

# run principal component analysis of file h, save as pc

# print summary of data: pc$loadings and pc$scores

ghi1 <- as.table(pc$scores)

ghi2 <- as.table(pc$loadings)

# output of pc$scores in table format into file ghi1

# output of pc$loadings in table format into file ghi2

write.table(ghi1, file="D:/Data/PCA_scores.txt", sep="\t",
row.names=T, col.names=T)

write.table(ghi2, file="D:/Data/PCA_loadings.txt", sep="\t",
row.names=T, col.names=T)

# save ghi1 in table format as file “D:/Data/PCA_scores.txt”

# fields separated by tab, file has row names and column names

# save ghi2 in table format as file “D:/Data/PCA_loadings.txt”
```

Output PCA_scores

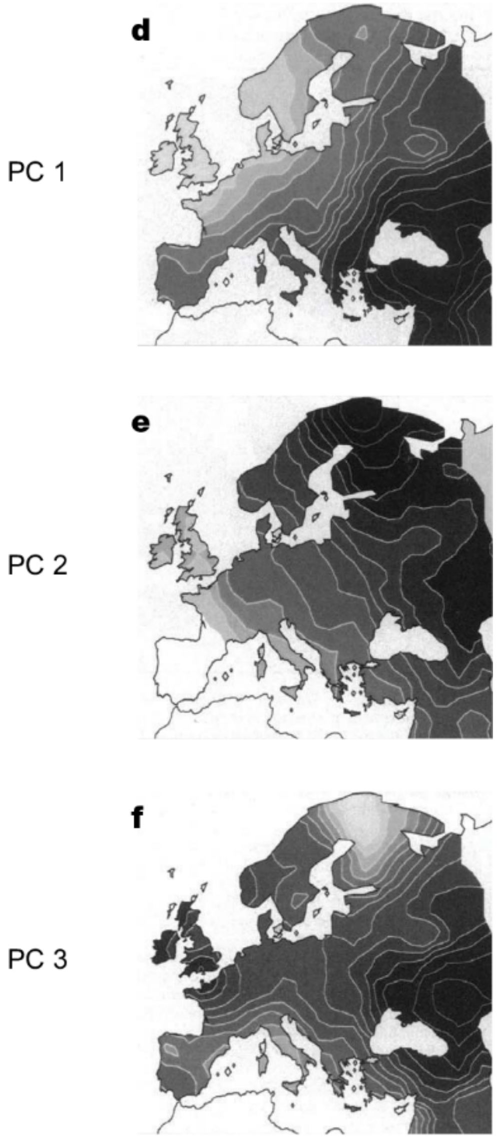
	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8	Comp.9	Comp.10	Comp.11	Comp.12	Comp.13	Comp.14	Comp.15	Comp.16	Comp.17
B.bronch1	3.940976	-0.65934	-0.35932	-0.33097	-0.78523	-0.63582	0.106812	-0.33411	0.251795	-0.83729	0.111922	-0.15431	0.170636	-0.08216	0.037813	-0.00413	0.001747
B.bronch2	3.467985	-0.26221	-0.73372	-0.2848	-0.10144	-0.71256	0.308428	-0.22728	-0.31109	-1.24364	-0.05382	0.083955	-0.1414	0.356394	-0.19073	0.076178	0.032616
B.bronch3	3.0684	0.631039	-1.6963	-0.13845	1.265976	-0.1194	0.149705	0.190226	-0.5807	-0.05045	-0.21447	0.205404	-0.35658	-0.14436	0.076716	-0.01255	-0.0154
B.bronch4	2.877919	0.864665	-0.92187	-0.50047	1.548399	-0.52757	0.272852	-0.06821	-0.03708	0.741385	-0.115	0.200025	-0.28327	0.32901	-0.18223	0.105868	0.03109
B.bronch5	2.558964	0.94425	-1.57696	0.238629	1.058568	0.560872	-0.33912	0.777675	-1.01252	0.00307	-0.06791	-0.1346	0.360272	-0.21061	0.152336	-0.06685	-0.03566
B.bronch6	3.703721	0.005205	-0.15197	-0.67054	-0.25434	-0.31372	0.073549	-0.37075	0.572002	0.745596	0.348163	-0.55449	-0.29786	0.186175	-0.22708	0.059994	-0.009
B.bronch7	3.338116	-0.09097	0.052605	-0.49044	0.440996	-1.20112	0.187546	-0.36444	0.738305	0.354975	-0.02958	0.271254	0.893447	-0.22867	0.262911	-0.03553	-0.00597
B.bronch8	3.44944	0.046542	-0.74398	-0.01318	-0.81557	0.840945	-0.51252	0.391626	-0.2547	0.098619	0.353441	-0.7869	0.291786	-0.00693	-0.08847	0.063754	-0.00111
B.parahu	3.535931	-0.999	-0.80005	-0.49297	-0.86969	0.71525	0.003884	-0.33116	0.424089	0.051217	-0.07841	0.168235	-0.73995	-0.52358	0.315321	-0.20105	-0.02009
B.paraov	2.777047	-1.18401	-0.26294	-0.11987	-1.06511	1.975882	-0.06008	0.00801	0.238236	0.190538	-0.36508	0.660132	0.324452	0.363885	-0.12001	0.127765	0.026893
B.pertussis1	1.766612	-0.93116	3.810397	1.092294	-0.48526	-0.66592	-0.37389	0.495592	-0.3159	0.197566	-0.16602	0.138258	-0.03243	0.03203	-0.30828	-0.64827	0.06748
B.pertussis2	1.042796	-0.71475	4.06178	1.310539	-0.4259	-0.61146	-0.36971	0.496295	-0.25876	0.112637	0.007457	0.10185	-0.22765	-0.03094	0.299929	0.635849	-0.06987
B.holmesii	-0.90844	0.633103	3.204297	1.568969	1.713535	1.408775	1.119641	-0.48406	0.37677	-0.36079	0.207976	-0.25971	0.060057	-0.04288	0.032629	-0.0665	0.014726
B.hinzii1	-3.76295	2.059499	0.678829	-2.13513	-0.04269	0.056194	0.172072	0.893481	0.445499	-0.20637	0.109606	0.198674	0.003395	-0.4445	-0.4889	0.187332	0.252445
B.hinzii2	-3.49032	2.403655	0.407988	-1.67139	0.238094	0.081278	-0.45688	0.867654	0.753546	-0.28505	0.032411	-0.05082	-0.12367	0.407449	0.390635	-0.18896	-0.30465
B.avium197N	-4.11968	0.903954	1.010648	-2.19459	-1.33046	-0.10379	1.159603	-0.52777	-1.18332	0.367323	0.133672	0.006217	0.094045	0.060943	0.179615	-0.05658	-0.0558
B.trematum	-3.5035	1.965244	1.057325	-0.72796	0.489283	0.188769	-1.50747	-1.4825	-0.21438	-0.10784	-0.40126	-0.10921	0.023846	-0.04558	-0.05005	0.025121	0.080502
B.petriiJ49	-2.83216	1.640384	-1.7567	2.252418	-0.33904	-0.112	-0.34345	-0.36313	-0.09977	0.03703	0.891021	0.506595	-0.00565	-0.09372	-0.19309	0.043179	-0.19803
B.petriiJ51	-3.55346	1.498028	-1.8962	2.036387	-0.63167	-0.30027	0.132559	0.176598	0.17416	0.084028	0.128442	0.011786	-0.05385	0.200129	0.255819	-0.17234	0.291509
B.petriiDSM	-3.71508	0.948995	-1.67945	1.984304	-0.75985	-0.37029	0.550464	0.119218	0.274471	0.107291	-1.00195	-0.38378	0.025422	-0.11243	-0.14713	0.119742	-0.11404
B.ansorpii1	-4.89809	-5.10786	-0.76678	-0.41503	0.589317	-0.08374	-0.06736	0.020922	0.006431	-0.01095	0.028432	-0.03379	0.036317	-0.07135	-0.1765	-0.02777	-0.55157
B.ansorpii2	-4.74422	-4.59526	-0.93764	-0.29775	0.562071	-0.0703	-0.20664	0.116131	0.01294	0.011119	0.140955	-0.08477	-0.02136	0.10169	0.168743	0.035767	0.582183

Load in Excel and plot pairwise



Supplementary Figure 4. Principal Component Analysis of presence/absence of virulence-associated factors in Bordetella genomes by A) *Bordetella* species; B) host and disease. The genomes from each species were grouped by presence/absence of individual factors, and any unique combination of factors was analyzed as separate data entry resulting in several data points per species. PC1 divides the classical from the non-classical species, PC2 isolates *B. ansorpii*, and PC3 separates the genomes of the human-restricted *B. pertussis* and *B. holmesii* from those of the other species. Bb *B. bronchiseptica*; Bpp *B. parapertussis*; Bp *B. pertussis*; Bhl *B. holmesii*; Bhz *B. hinzii*; Bav *B. avium*; Bt *B. trematum*; Bpet *B. petrii*; Ban *B. ansorpii*

Example from human genetics:
Allele frequencies of 95 allozymes in Europe and the Middle East

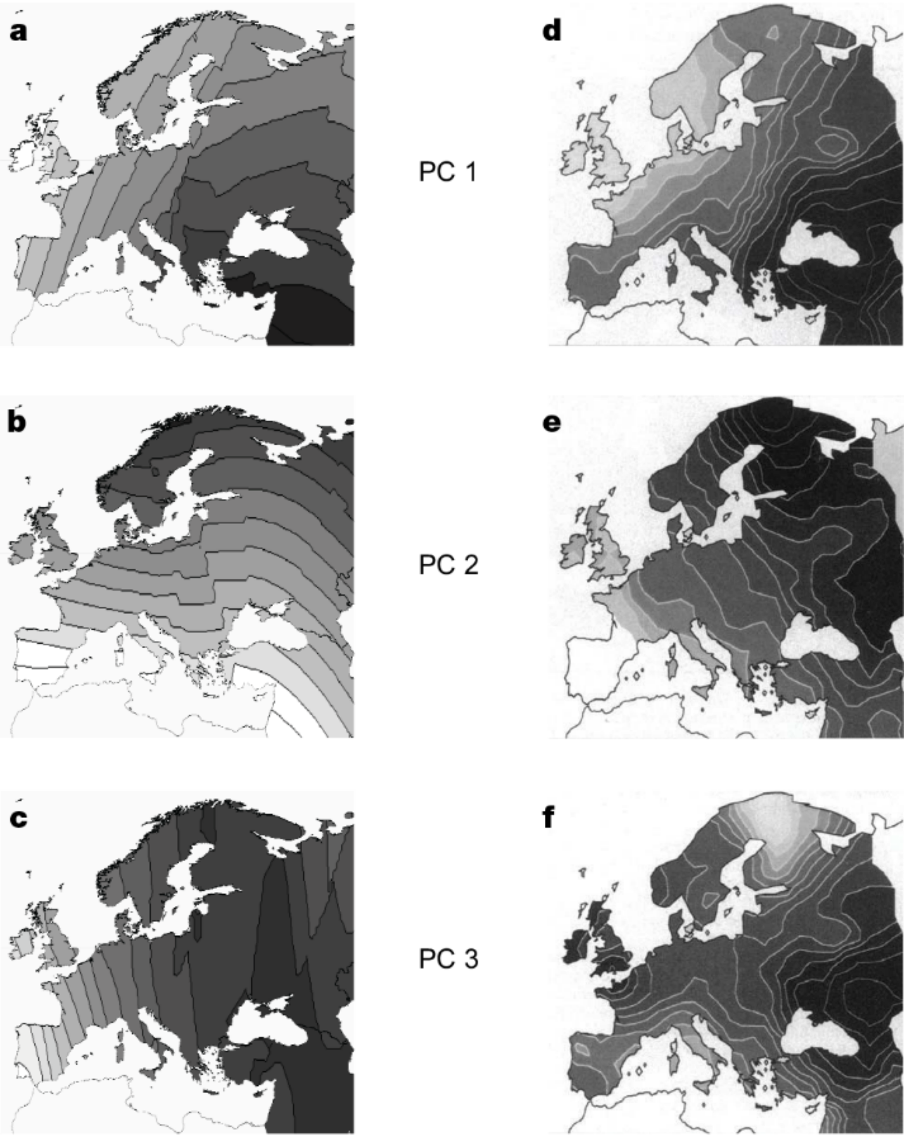


Clinal gradients in principal components 1–3 in allozyme allele frequencies in Europeans

Piazza et al., (1995).
Genetics and the origin
of European languages
Proc. Natl. Acad. Sci. USA
Vol. 92, pp. 5836-5840

Example from human genetics and the human stomach bacterium *Helicobacter pylori*: Allele frequencies of 95 allozymes and *H. pylori* gene sequences in Europe and the Middle East

Similar clinal gradients between principal components 1–3 in European *H. pylori* and humans

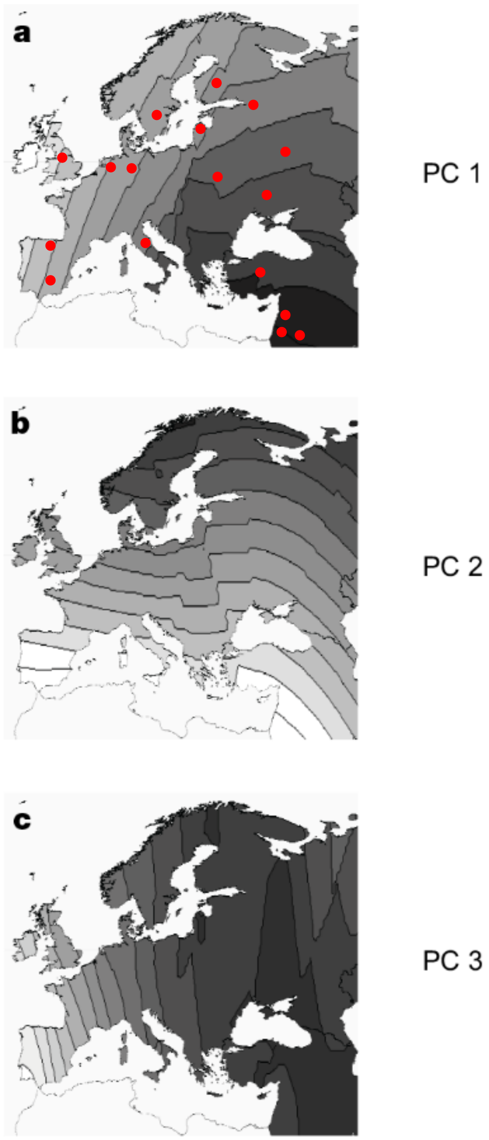


Clinal gradients in principal components 1–3 in allozyme allele frequencies in Europeans

Linz et al., (2007).
An African origin for the intimate association between humans and *Helicobacter pylori*
Nature Vol. 445, pp. 915-918

Piazza et al., (1995).
Genetics and the origin of European languages
Proc. Natl. Acad. Sci. USA
Vol. 92, pp. 5836-5840

PCA of gene sequences from H. *pylori* in Europe



- concatenated MLST sequences of *H. pylori* sampled from patients at multiple locations
- grouped by sampling location
- changed nucleotides to allele numbers
- ran PCA
- subjected data from each individual PC to spatial autocorrelation analysis in GS+ 7.0 (Geostatistics software for the Environmental Sciences)
- extrapolated data points throughout the grid
- plotted onto a synthetic map of Europe using arcGIS

- clines originally interpreted as genetic signatures of episodic migratory events:

  PC1: spread of agriculture from Middle East to Europe

  PC2: introgression of Uralic speaking peoples from northern Siberia into northern Europe (Lapps, Finns, Estonians, Hungarians)

  PC3: Spread of the Kurgan culture (pastoral nomads) from Eurasian steppes after domestication of the horse

Let’s change the topic:

How to get a specific gene sequence from a short read archive

We will:

Download a Short Read Archive (SRA) from NCBI

↓

extract reads for
a specific gene

↓

assemble the gene
sequence from the reads

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

```
~/[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/Bhinzii/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 – Dump each read into separate file. Files will receive suffix corresponding to read number.

# --clip SRR_ID – change the SRR_ID to whatever the ID is, e.g. SRR942665

Download a short read archive (SRA) from NCBI

Let's assume we downloaded the 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
SEQUENCE
+ (sometimes with seqID again)
QUALITY_SCORES_FOR_ALL_NUCLEOTIDES
```

e.g.

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

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

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

Download a short read archive (SRA) from NCBI

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 more desired 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 100 –x 0.1
```

results in:

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

Download a short read archive (SRA) from NCBI

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

rename: `mv out.extendedFrags.fastq SRR942665_joined.fastq`

Let’s extract the reads for a certain membrane transporter gene (locus_tag BB1335 in *B. bronchiseptica* RB50) to check for a frameshift mutation in this *B. hinzii* genome using `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/)

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

Let’s dig in:

```
cat SRR942665_joined.fastq | …
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
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?
Well, you know what to do:

```
printf “Bhinzii5132 consensus\n” > Bhinzii5132_BB1335.fa
cat Bhinzii5132_BB1335_consensus.fa
| awk '{
       if(substr($1,1,1) == ">"){
             printf"";
        }else{
             printf"%s",$1;
        }
     } END{printf"\n"}' \
>> Bhinzii5132_BB1335.fa

cat Bhinzii5132_BB1335.fa | wc -L

# wc -L prints the length of the longest line
# Result: 1415
# That means, we are dealing with the frameshift gene variant
```

How many nucleotides?
Easier solution:

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

# 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
# → 0  1     1415 (0 lines, 1 word, 1415 characters)
```

or

```
cat Bhinzii5132_BB1335_consensus.fa \
| grep -v “>” | tr -d “\n” | wc -L
# returns number of characters in longest line: 1415
```

Thank you.