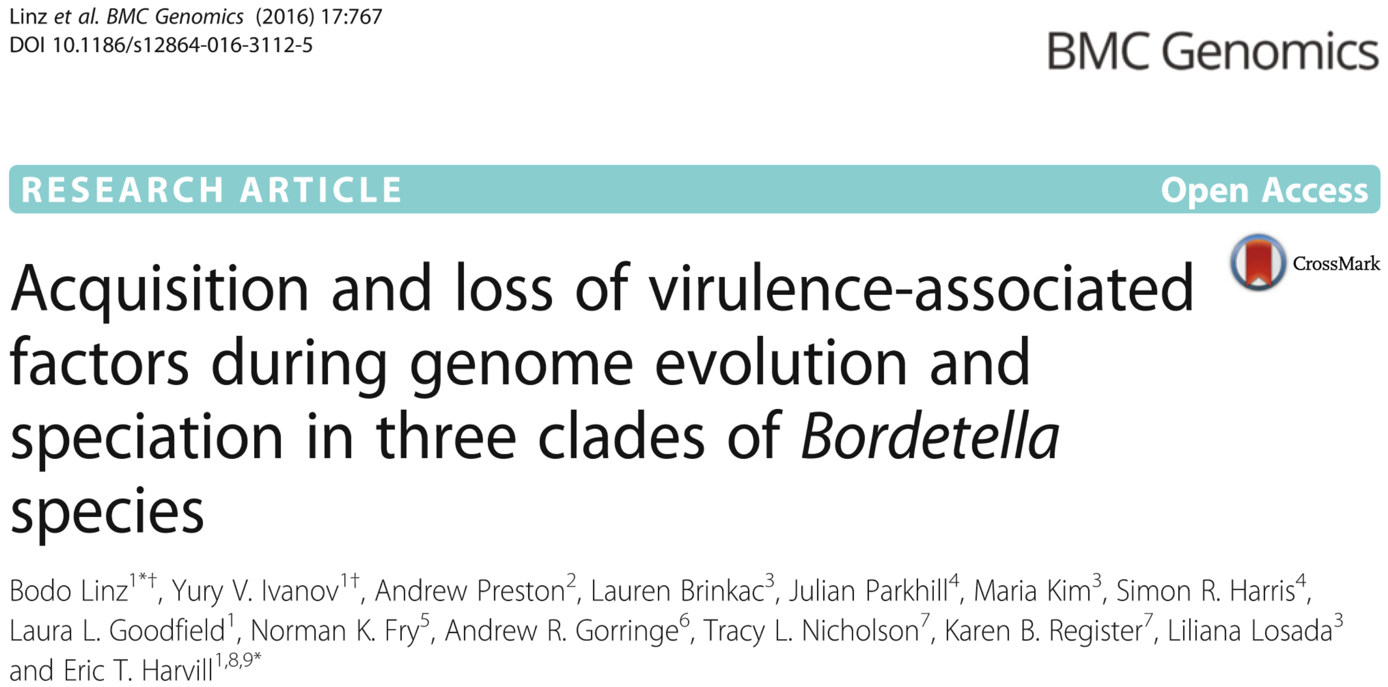
Guest Lecture

Bodo Linz
02/11/20

Comparative genomics of *Bordetella*



Linz *et al. BMC Genomics* (2016) 17:767
DOI 10.1186/s12864-016-3112-5

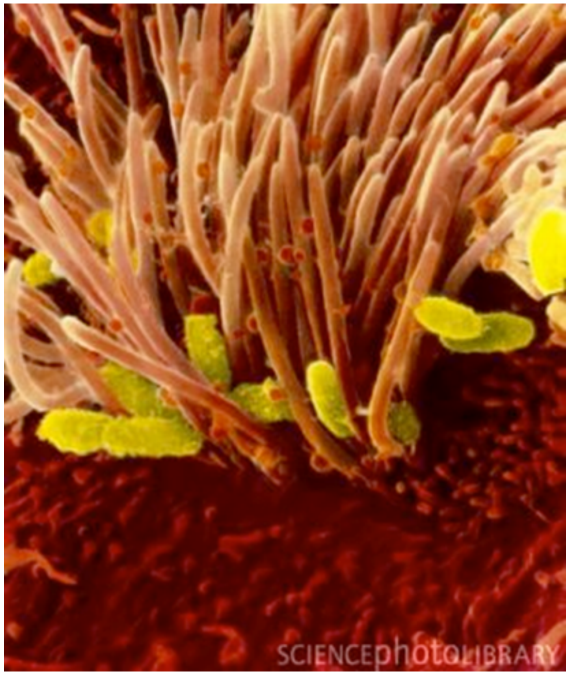
BMC Genomics

RESEARCH ARTICLE Open Access

Acquisition and loss of virulence-associated factors during genome evolution and speciation in three clades of *Bordetella* species

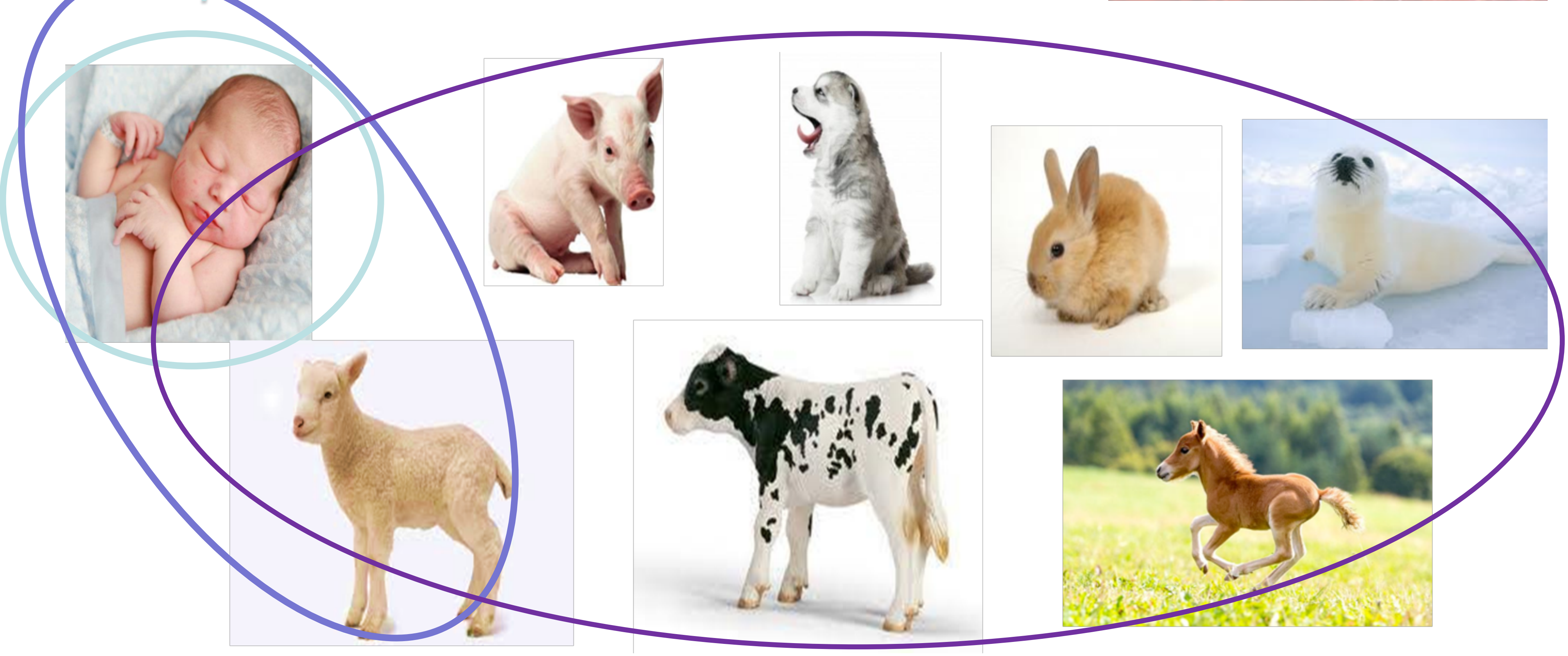
Bodo Linz[1*†], Yury V. Ivanov[1†], Andrew Preston[2], Lauren Brinkac[3], Julian Parkhill[4], Maria Kim[3], Simon R. Harris[4], Laura L. Goodfield[1], Norman K. Fry[5], Andrew R. Gorringe[6], Tracy L. Nicholson[7], Karen B. Register[7], Liliana Losada[3] and Eric T. Harvill[1,8,9*]

The Bordetellae

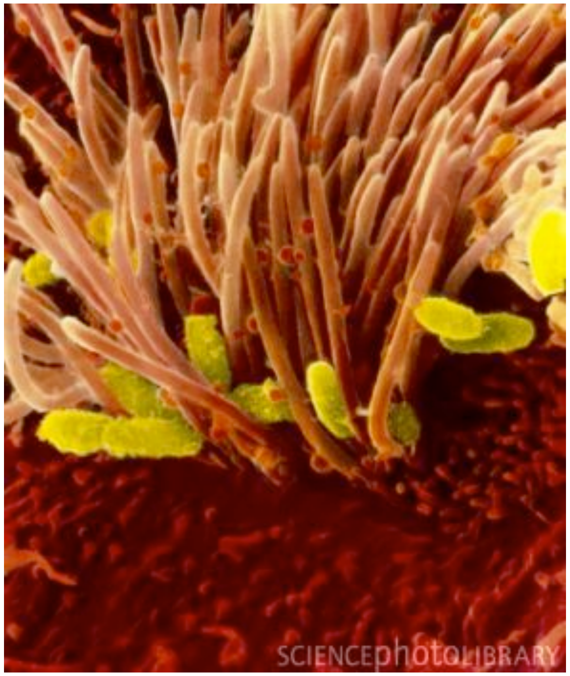


Beta-Proteobacteria
Include the classical bordetellae:

- *B. bronchiseptica*
- *B. parapertussis*
- *B. pertussis*



The Bordetellae



- Include the classical bordetellae:
  - *B. bronchiseptica*
  - *B. parapertussis*
  - *B. pertussis*

- Non-classical:
  - *B. holmesii*
  - *B. hinzii*
  - *B. avium*

  respiratory pathogens in animals and
  in immuno-compromized humans

  - *B. trematum*
  - *B. ansorpii*

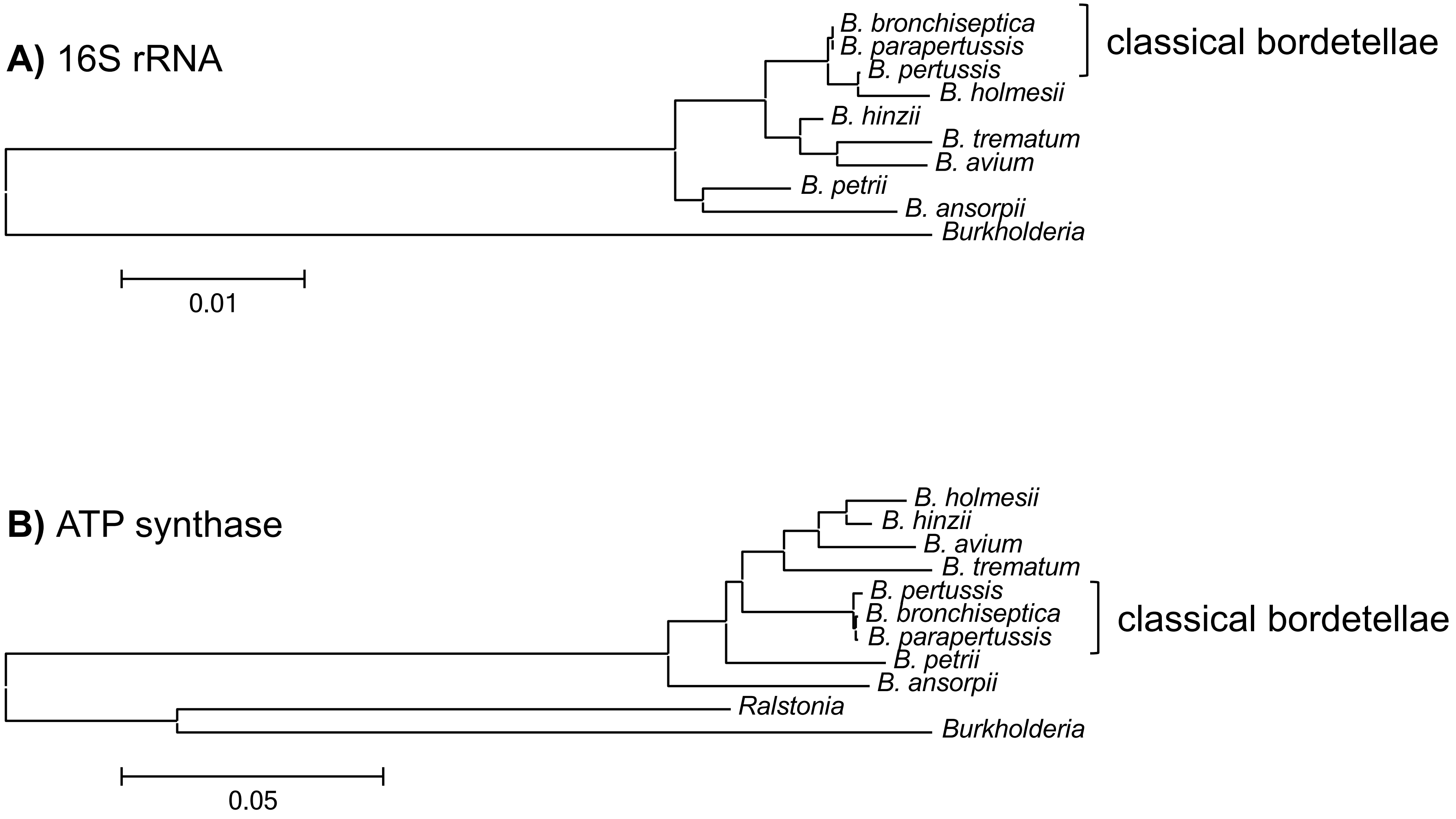
  wound and ear infection in humans

  - *B. petrii*

  environmental / ear infection in humans

+ several other recently described species

Neighbor-joining trees of 16S rRNA gene sequences and 8 concatenated ATP synthase proteins from *Bordetella*



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)

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

Approach

genome-wide SNP-based phylogenetic tree

- align genomes
  - align short reads against reference genome
  - call SNPs
  - generate consensus sequence
  - alignment of multiple genomes
- generate phylogenetic tree

next
week’s
lecture

Approach

data format: Sequence alignment in rows
Name SEQUENCE

```
SAMPLE01C  CGTTGCTGGCCGGATTTGCGCAGCAGGCGCGCGATCTCGTGGTCGTGCGCATTGACGCCCGCCCGCGCATCGACCAGGAACACCAC
SAMPLE02A  CGCTGCTGGCCGGATTTGCGCAGCAGGCGCGCGATCTCGTGGTCGTGCGCATTGACGCCCGCCCGCGCATCGACCAGGAACACCAC
SAMPLE03T  CGCTGCTGGCCGGACTTGCGCAGCAGGCGCGCGATCTCGTGGTCGTGCGCATTGACGCCCGCCCGCGCATCGACCAGGAACACCAC
SAMPLE-04  CGCTGCTGGCCAGATTTACGGAGC----------TTTCGTGGTCGTGCGCGTTGACGCCGGCGCGCGCGTCGACCAGGAACACCAC
SAMPLE05G  CGCTGCTGGCCGGATTTGCGCAGCAGGCGGGCGATTTCGTGGTCGTGCGCGTTGATGCCGGCACGGGCATCGACCAGGAACACGAC
SAMPLE06   CGCTGCTGGCCGGACTTGCGCAGCAGGCGGGCGATCTCGTGGTCATGCGCGTTGATCCCCGCCCGCGCGTCGACCAGGAAGACCAC
SAMPLE-7A  CGCTGCTGACCGGACTTACGCAG---------------------------------------------------------------
SAMPLE08B  CGCTGCTGGCCGGACTTGCGCAACAAGCGGGCGAT-----------------------CGGCCCGGGCGTCGACCAGGAACACCAC
SAMPLE09   CGCTGCTGCCCGGACTTGCGCAACAGGCGGGCGAT--------------------------------------------ACACCAC
```

Data format: 1 reference genome (5.3 MB), all other genomes aligned against it

Problem: missing data (dashes)

- gene not present
- gene so divergent that the sequence did not align
- multiple copies of a gene

Solution: remove all positions with missing data in any of the genomes

Approach

data format: Sequence alignment in rows
Name SEQUENCE
$1 $2 $1 = field 1; $2 = field 2

```
SAMPLE01C  CGTTGCTGGCCGGATTTGCGCAGCAGGCGCGCGATCTCGTGGTCGTGCGCATTGACGCCCGCCCGCGCATCGACCAGGAACACCAC
SAMPLE-04  CGCTGCTGGCCAGATTTACGGAGC----------TTTCGTGGTCGTGCGCGTTGACGCCGGCGCGCGCGTCGACCAGGAACACCAC
SAMPLE05G  CGCTGCTGGCCGGATTTGCGCAGCAGGCGGGCGATTTCGTGGTCGTGCGCGTTGATGCCGGCACGGGCATCGACCAGGAACACGAC
SAMPLE-7A  CGCTGCTGACCGGACTTACGCAG---------------------------------------------------------------
```

- **`awk`: change strain names to lower case and replace '-' by '_'**
- **`python`: replace nucleotides by nucleotides plus tab**
- **`awk`: remove extra tab at the end of each line**
- **`python`: transpose rows to columns**
- **`awk`: select only core loci**
- **`grep | wc`: determine the number of loci in the resulting file**
- **`python`: replace nucleotides by numbers**
- **`R:` calculate matrix**
- **`python`: transpose columns to rows**
- **`awk`: add extra tab at the end of each line**
- **`python`: replace nucleotides plus tab by nucleotides**

Approach

data format: Sequence alignment in rows

Name   SEQUENCE

$1 $2 $1 = field 1; $2 = field 2

```
SAMPLE01C  CGTTGCTGGCCGGATTTGCGCAGCAGGCGCGCGATCTCGTGGTCGTGCGCATTGACGCCCGCCCGCGCATCGACCAGGAACACCAC
SAMPLE-04  CGCTGCTGGCCAGATTTACGGAGC----------TTTCGTGGTCGTGCGCGTTGACGCCGGCGCGCGCGTCGACCAGGAACACCAC
SAMPLE05G  CGCTGCTGGCCGGATTTGCGCAGCAGGCGGGCGATTTCGTGGTCGTGCGCGTTGATGCCGGCACGGGCATCGACCAGGAACACGAC
SAMPLE-7A  CGCTGCTGACCGGACTTACGCAG---------------------------------------------------------------
```

- **need to manipulate nucleotide sequence in all rows**
- **problem: same letters in sequence names**
- **solution: sequence name lower case, sequence upper case,**
  dashes in names as underline
- **awk: change strain names to lower case and replace '-' by '_'**

MAKE THE SCRIPT USER FRIENDLY!!!

- write instructions to yourself
- let the computer display what it's currently doing

- awk: change strain names to lower case and replace '-' by '_‘

```
#!/bin/bash
# PhyGenome_Align_remove_missing_data.sh
# remove variably present loci, keep only core loci

# enter file names as needed
FILESNP="128genomes.phy"
NAMESNP=${FILESNP%%".phy"}
```

write instructions to yourself

you can either define the input file once or enter it again and again throughout the script

```
echo ""
echo "loading input file $NAMESNP"
echo ""
echo "awk: change strain names to lower case and '-' to '_'"
echo "----------------------------------------------------"
```

echo “ ” - let the computer display to the user what it is currently doing

```
# make sequence name lower case
cat $FILESNP | awk -v FS="\t" -v OFS="\t" '{$1=tolower($1);
print $0}' > fake
```

Let’s go through this command →

- awk: change strain names to lower case and replace '-' by '_‘

```
# make sequence name lower case
cat $FILESNP | awk -v FS="\t" -v OFS="\t" '{$1=tolower($1);
print $0}' > fake

# cat - concatenate
# open 1 file, open and combine (=concatenate) several files

# | pipe - string several commands together into a pipeline
#        - input from memory, output into memory

# FS="\t" - Field Separator is tab: $1   $2
# OFS="\t" - Output Field Separator is tab

# '{}' - what to do
# $1=tolower($1) - new field $1 is lower case of current $1
# print $0 - print all fields

# > save as
```

- **awk: change strain names to lower case and replace '-' by '_‘**

```
# make sequence name lower case
cat $FILESNP | awk -v FS="\t" -v OFS="\t" '{$1=tolower($1);
print $0}' > fake


# replace (substitute) "-" to "_" in strain names
cat $FILESNP | awk -v FS="\t" -v OFS="\t" '{gsub(/-/,"_",$1);
print $0}' > fake


# Why "gsub" and not "sub"?  assume strain name: M1989-03-14

awk '{sub(/-/"_",$1);print $0}'
# replaces only 1st instance: M1989_03-14

awk '{gsub(/-/,"_",$1);print $0}'
# replaces ALL instances in a line: M1989_03_14
```

- awk: change strain names to lower case and replace '-' by '_‘

```
# make sequence name lower case
cat $FILESNP | awk -v FS="\t" -v OFS="\t" '{$1=tolower($1);
print $0}' > fake
```

```
# replace (substitute) "-" to "_" in strain names
cat $FILESNP | awk -v FS="\t" -v OFS="\t" '{gsub(/-/,"_",$1);
print $0}' > fake
```

Let’s pipe it:

```
# replace "-" to "_" in strain names and lower case
cat $FILESNP | awk -v FS="\t" -v OFS="\t" '{$1=tolower($1);
print $0}' | awk -v FS="\t" -v OFS="\t" '{gsub(/-/,"_",$1);
print $0}' > fake
```

```
SAMPLE01C  CGTTGCTGGCCGGATTTGCGCAGCAGGCGCGATCTCGTGGTCGTGCGCATTGACGCCCGCCCGCGCATCGACCAGGAACACCAC
SAMPLE-04  CGCTGCTGGCCAGATTTACGGAGC----------TTTCGTGGTCGTGCGCGTTGACGCCGGCGCGCGCGTCGACCAGGAACACCAC
```

```
sample01c  CGTTGCTGGCCGGATTTGCGCAGCAGGCGCGATCTCGTGGTCGTGCGCATTGACGCCCGCCCGCGCATCGACCAGGAACACCAC
sample_04  CGCTGCTGGCCAGATTTACGGAGC----------TTTCGTGGTCGTGCGCGTTGACGCCGGCGCGCGCGTCGACCAGGAACACCAC
```

- python: change nucleotides to nucleotides plus tab

```
# insert tab after each nucleotide to get independent loci,
input_file "fake", output_file "fake2"
echo ""
echo "python: replace nucleotides by numbers plus tab"
echo "-----------------------------------------------------"
python2.6 ../../bin/replace_nucs_to_nucsplustab_in_file.py
```

# call python v2.6   # where is the script

```
sample01c  CGTTGCTGG...
sample_04  CGCTGCTGG...

sample01c  C          G          T          T          G          C          T          G          G
sample_04  C          G          C          T          G          C          T          G          G
```

Python script: `replace_nucs_to_nucsplustab_in_file.py`

```
#!/usr/bin/env python
input = open('fake', "r")
output = open('fake2', "w")


stext1 = 'A'  rtext1 = 'A\t'
stext2 = 'C'  rtext2 = 'C\t'
stext3 = 'G'  rtext3 = 'G\t'
stext4 = 'T'  rtext4 = 'T\t'
stext5 = '-'  rtext5 = 'Z\t'     # why Z? Any letter not A C G T or N will do
                                 (or not IUPAC depending on what you wanna do)
stext6 = 'N'  rtext6 = 'Z\t'


output.write(input.read().replace(stext1,
rtext1).replace(stext2, rtext2).replace(stext3,
rtext3).replace(stext4, rtext4).replace(stext5,
rtext5).replace(stext6, rtext6))
```

- awk: remove extra tab at the end of the line

```
# remove extra tab at the end of each line
echo ""
echo "awk: remove extra tab at the end of each line"
echo "-----------------------------------------------------"
cat fake2 | awk -v FS="\t" -v OFS="\t" '{sub(/[ \t]+$/, "");
print $0}' > fake3
```

- python: transpose rows to columns

```
# transform rows to columns

echo ""

echo "python: transpose rows to columns"

echo "----------------------------------------------------"

cat fake3 | python2.6 ../../bin/rows2columns_transposition.py
> fake4
```

# This time we pipe python. Input from memory, output to memory.

Python script: `rows2columns_transposition.py`

```
#!/usr/bin/env python

"""

rows_to_colums_transposition.py

input(sys.stdin) : A file with strains and tab separated
loci in rows

output (sys.stdout): A file with strains and loci in
columns

"""

import sys


for c in zip(*(l.strip().split() for l in
sys.stdin.readlines() if l.strip())):

    print('\t'.join(c))
```

- awk: select core loci (no missing data)

The story so far:

- we renamed $1 to lower case and changed “–” to “_”
- we replaced missing data (“-”, “N”) with “Z”
- we transposed rows to columns

sample1c	sample_04	sample05g	sample_7a
A	G	A	A
A	G	T	T
A	G	Z	Z
C	C	C	T

```
# select only rows that do not contain "Z" (=core loci only)
echo ""
echo "selecting core loci"
cat fake4 | grep -v "Z"  > fake5
cat fake5 > fake5_$NAMESNP.txt
# grep – global regular expression print – (“grab”)
# -v --invert-match (select all lines that do not contain Z)
```

- awk: select core loci (no missing data)

The story so far:

- we renamed $1 to lower case and changed “–” to “_”
- we replaced missing data (“-”, “N”) with “Z”
- we transposed rows to columns
- we selected core loci

sample1c	sample_04	sample05g	sample_7a
A	G	A	A
A	G	T	T
C	C	C	T

How many loci did we end up with?

```
# determine the number of loci in the resulting file

cat fake5 | grep -v s | wc -l > fake5a

echo "The dataset from file '$NAMESNP' consists of $(cat 
fake5a) core loci. "


# grep -v s - select all lines that do not contain “s”

# wc -l - word count, count the number of lines (-l)

# cat fake5a - open file fake5a, which is just a number
```

- awk: select core loci (no missing data)

The story so far:

- we renamed $1 to lower case and changed “–” to “_”
- we replaced missing data (“-”, “N”) with “Z”
- we transposed rows to columns
- we selected core loci

sample1c	**sample_04**	**sample05g**	**sample_7a**
A	**G**	**A**	**A**
A	**G**	**T**	**T**
C	**C**	**C**	**T**

How many loci did we end up with?

```
# determine the number of loci in the resulting file

# grep -v s - requires a common character (“s”)in all names

# alternatively:

cat fake5 | awk 'NR>1' | wc -l > fake5a


# awk 'NR>1' - select all lines (=rows) after the first
```

- **python: replace nucs by numbers (fake5 > fake6) as before (stext and rtext)**

- **python: transpose columns to rows**

```
# transform columns to rows

echo "python: transpose columns to rows"

echo "----------------------------------------------------"

cat fake6 | python2.6 ../../bin/rows2columns_transposition.py >
fake7
```

-awk: add extra tab at the end of each line

```
cat fake7 | awk ‘{print $0”\t”}’ > fake 8
```

python: replace nucleotides plus tab by nucleotides

```
cat fake8 | python2.6
../../bin/replace_nucs_plus_tab_by_nucs.py > fake9
```

- write final output file

```
echo ""
echo "awk: writing output file"
echo "----------------------------------------------------"
cat fake9 | awk -v FS="\t" -v OFS="\t" '{print $1,$2}' >
$NAMESNP-no-gaps.phy
```

–R: Calculate Distance matrix

```
echo "R: Calculate Distance matrix."
echo "-----------------------------------------------------"
# Run R in '--slave' mode to incorporate in bash script
R --slave -f Dist_mat_Genomes.R
```

R:

- **another scripting language**
- **awesome for calculations**
- **syntax different from bash or python**

Syntax: R vs Python

R: read file

```
a <-read.table(“fake6”, header=TRUE, sep=“\t”)
```

Python: read file

```
input = open('fake6', "r")
```

R: transpose rows to columns

```
y = t(x)
```

Python: transpose rows to columns

```
for c in zip(*(l.strip().split() for l in
sys.stdin.readlines() if l.strip())):
    print('\t'.join(c))
```

R: write file

```
write.table(m5, file = “SEQ1.dist”, sep = “\t”, row.names =
FALSE, column.names = FALSE)
```

Python: write file

```
output = open('fake7', "w")
```

–R: Calculate Distance matrices of SNPs and Genes

```
#!usr/bin/R

#delete all objects

rm(list = ls())

#load packages

library(ade4)

library(MASS)

a <-read.table(“fake6”, header=TRUE, sep=“\t”) ## load data

x = t(a) ## transform data to genomes by row and SNPs by col

SEQ1.dist <- as.dist(dist(x, “manhattan”)) ## calc matrix

m5 <- as.matrix(SEQ1.dist) ## write as matrix

write.table(m5, file = “SEQ1.dist”, sep = “\t”, row.names =
FALSE, column.names = FALSE)
```

- transfer distance matrix
- change to MEGA format
- MEGA – Molecular Evolutionary Genetics Analysis
- load matrix and display tree

https://www.megasoftware.net/

MEGA format:

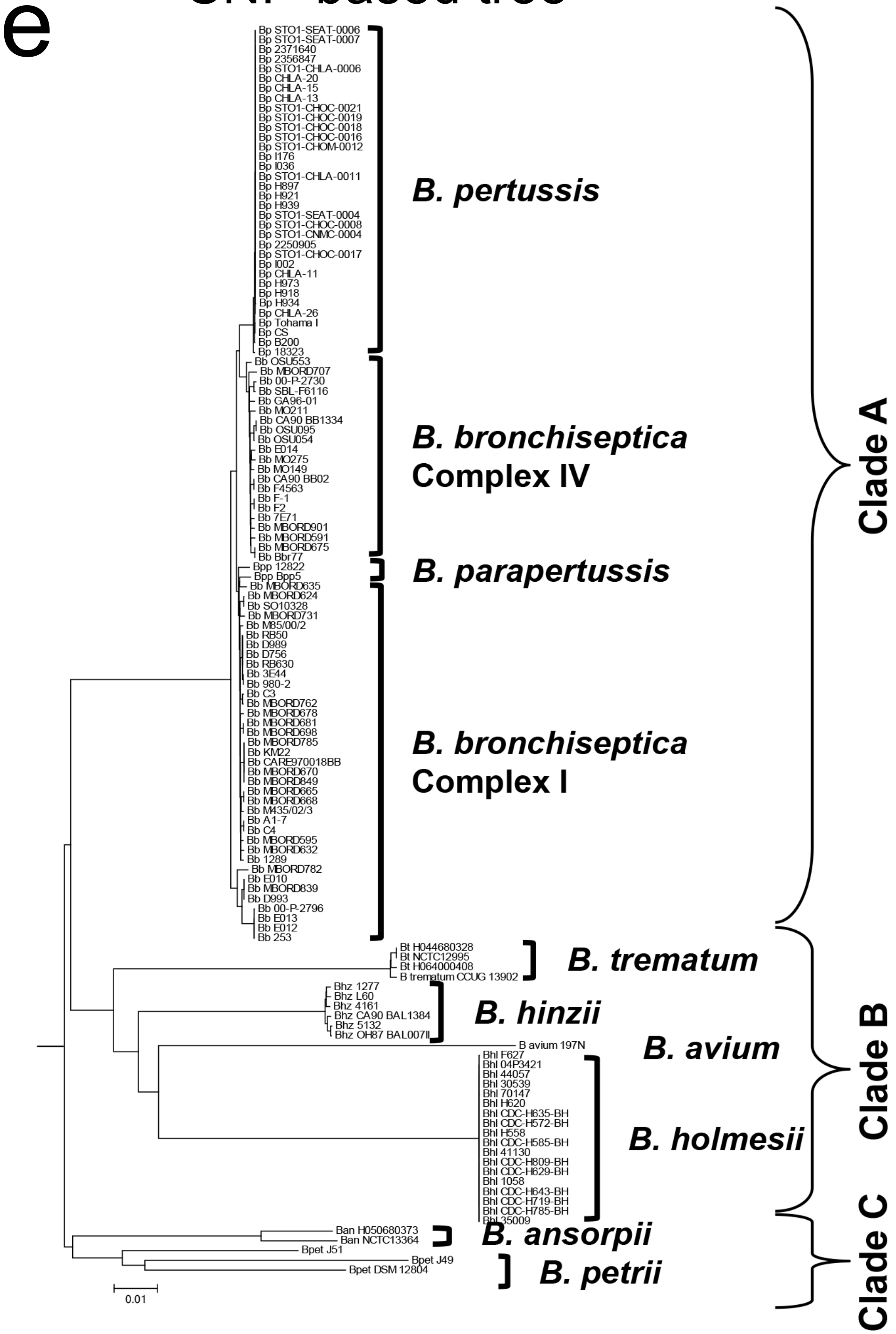
```
#mega
Title: distance matrix genome-wide SNPs in 128 Bordetella genomes;


[ 1]  # sample_1a
[ 2]  # sample02
[ 3]  #  sample3a
[ 4]  # sample4c

[               1               2               3               4   ]

[ 1]

[ 2]            0.007695584

[ 3]            0.000200096     0.007495488

[ 4]            0.00021632      0.007511712     0.000016224
```

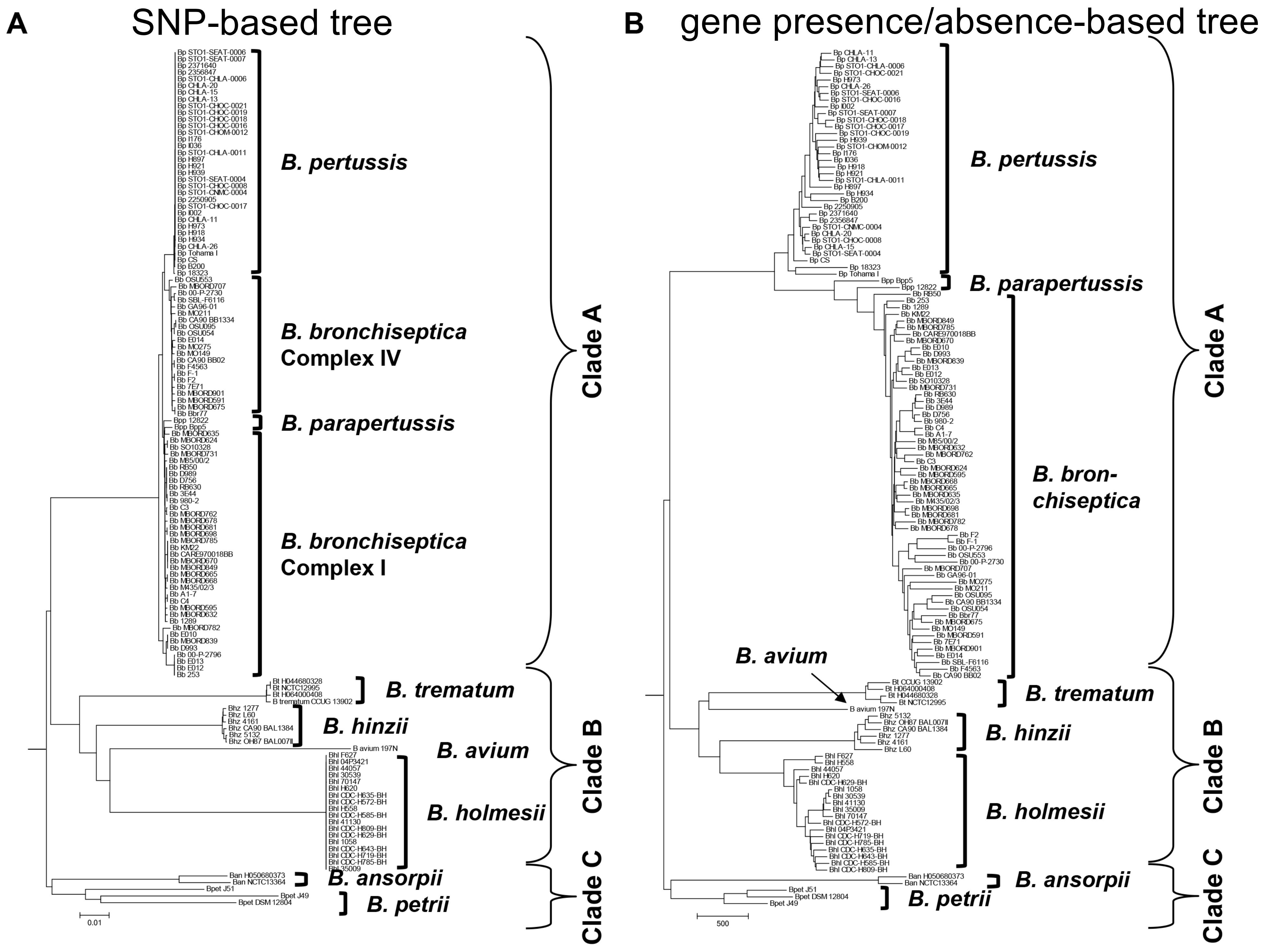
Change matrix to MEGA format: either by hand in text editor or by scripting

```
echo "Writing output file."
echo ""

printf "#mega\nTitle distance matrix of genome sequences from 10 Bordetella species;\n\n" > 10gen.meg
cat 10gen.phy | awk 'NR==1' | awk -v FS="\t" -v OFS="" '{print "[ 1]  #",$1}' >> 10gen.meg
cat 10gen.phy | awk 'NR==2' | awk -v FS="\t" -v OFS="" '{print "[ 2]  #",$1}' >> 10gen.meg
cat 10gen.phy | awk 'NR==3' | awk -v FS="\t" -v OFS="" '{print "[ 3]  #",$1}' >> 10gen.meg
cat 10gen.phy | awk 'NR==4' | awk -v FS="\t" -v OFS="" '{print "[ 4]  #",$1}' >> 10gen.meg
cat 10gen.phy | awk 'NR==5' | awk -v FS="\t" -v OFS="" '{print "[ 5]  #",$1}' >> 10gen.meg
cat 10gen.phy | awk 'NR==6' | awk -v FS="\t" -v OFS="" '{print "[ 6]  #",$1}' >> 10gen.meg
cat 10gen.phy | awk 'NR==7' | awk -v FS="\t" -v OFS="" '{print "[ 7]  #",$1}' >> 10gen.meg
cat 10gen.phy | awk 'NR==8' | awk -v FS="\t" -v OFS="" '{print "[ 8]  #",$1}' >> 10gen.meg
cat 10gen.phy | awk 'NR==9' | awk -v FS="\t" -v OFS="" '{print "[ 9]  #",$1}' >> 10gen.meg
cat 10gen.phy | awk 'NR==10' | awk -v FS="\t" -v OFS="" '{print "[10]  #",$1,"\n"}' >> 10gen.meg

printf "[\t1\t2\t3\t4\t5\t6\t7\t8\t9\t10  ]\n" >> 10gen.meg

printf "[ 1]\n" >> 10gen.meg
cat 10gens.dist | awk 'NR==2' | awk -v FS="\t" -v OFS="" '{print "[ 2]\t",$1}' >> 10gen.meg
cat 10gens.dist | awk 'NR==3' | awk -v FS="\t" -v OFS="" '{print "[ 3]\t",$1,"\t",$2}' >> 10gen.meg
cat 10gens.dist | awk 'NR==4' | awk -v FS="\t" -v OFS="" '{print "[ 4]\t",$1,"\t",$2,"\t",$3}' >> 10gen.meg
cat 10gens.dist | awk 'NR==5' | awk -v FS="\t" -v OFS="" '{print "[ 5]\t",$1,"\t",$2,"\t",$3,"\t",$4}' >> 10gen.meg
cat 10gens.dist | awk 'NR==6' | awk -v FS="\t" -v OFS="" '{print "[ 6]\t",$1,"\t",$2,"\t",$3,"\t",$4,"\t",$5}' >> 10gen.meg
cat 10gens.dist | awk 'NR==7' | awk -v FS="\t" -v OFS="" '{print "[ 7]\t",$1,"\t",$2,"\t",$3,"\t",$4,"\t",$5,"\t",$6}' >> 10gen.meg
cat 10gens.dist | awk 'NR==8' | awk -v FS="\t" -v OFS="" '{print "[ 8]\t",$1,"\t",$2,"\t",$3,"\t",$4,"\t",$5,"\t",$6,"\t",$7}' >> 10gen.meg
cat 10gens.dist | awk 'NR==9' | awk -v FS="\t" -v OFS="" '{print "[ 9]\t",$1,"\t",$2,"\t",$3,"\t",$4,"\t",$5,"\t",$6,"\t",$7,"\t",$8}' >> 10gen.meg
cat 10gens.dist | awk 'NR==10' | awk -v FS="\t" -v OFS="" '{print "[10]\t",$1,"\t",$2,"\t",$3,"\t",$4,"\t",$5,"\t",$6,"\t",$7,"\t",$8,"\t",$9,"\n"}' >> 10gen.meg

echo ""
echo "Done."
echo ""
```

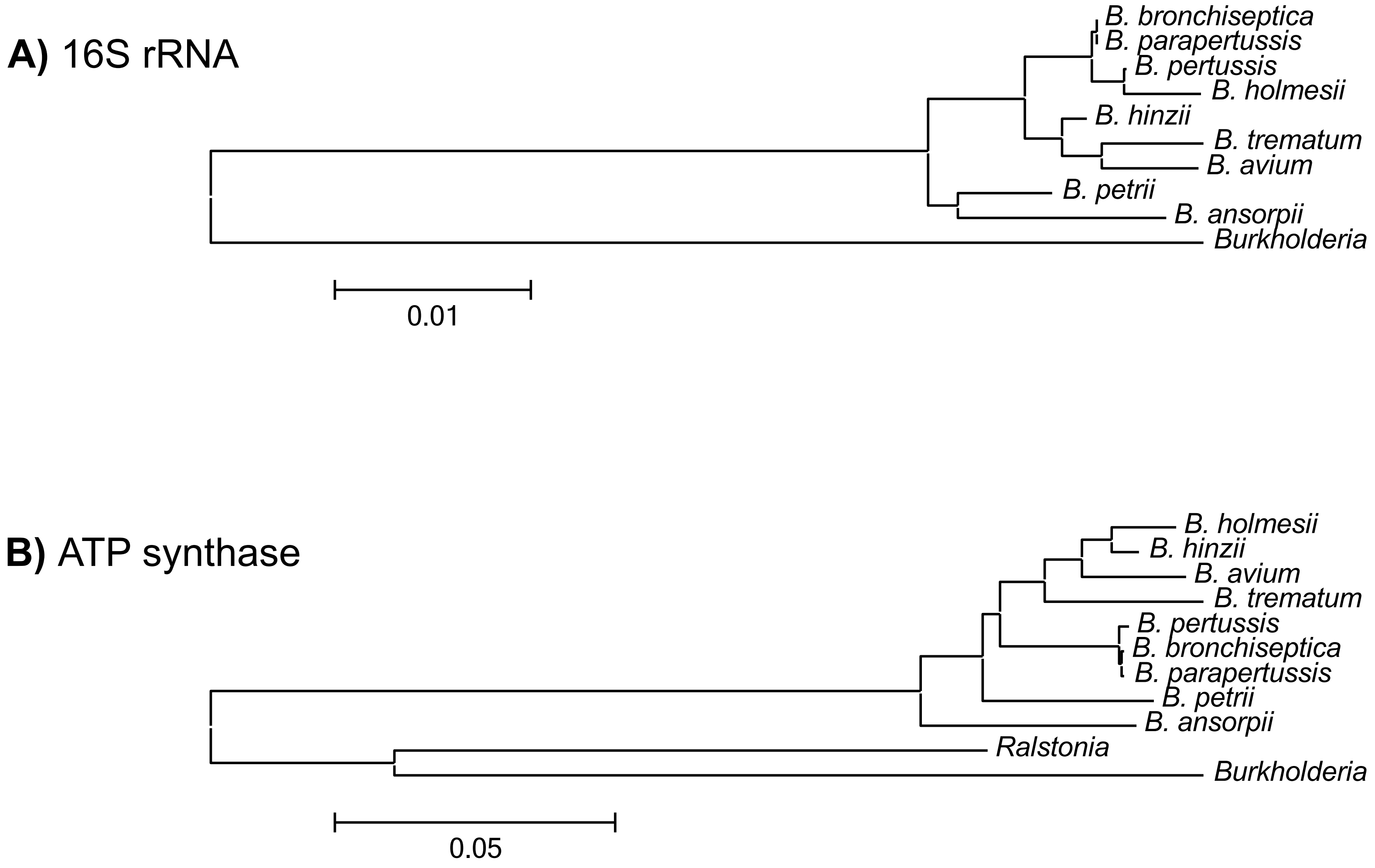
Display tree

SNP-based tree




A
SNP-based tree
Bp STO1-SEAT-0006
Bp STO1-SEAT-0007
Bp 2371640
Bp 2356847
Bp STO1-CHLA-0006
Bp CHLA-20
Bp CHLA-15
Bp CHLA-13
Bp STO1-CHOC-0021
Bp STO1-CHOC-0019
Bp STO1-CHOC-0018
Bp STO1-CHOC-0016
Bp STO1-CHOM-0012
Bp I176
Bp I036
Bp STO1-CHLA-0011
Bp H897
Bp H921
Bp H939
Bp STO1-SEAT-0004
Bp STO1-CHOC-0008
Bp STO1-CNMC-0004
Bp 2250905
Bp STO1-CHOC-0017
Bp I002
Bp CHLA-11
Bp H973
Bp H918
Bp H934
Bp CHLA-26
Bp Tohama I
Bp CS
Bp B200
Bp 18323
B. pertussis
Bb OSU553
Bb MBORD707
Bb 00-P-2730
Bb SBL-F6116
Bb GA96-01
Bb MO211
Bb CA90 BB1334
Bb OSU095
Bb OSU054
Bb E014
Bb MO275
Bb MO149
Bb CA90 BB02
Bb F4563
Bb F-1
Bb F2
Bb 7E71
Bb MBORD901
Bb MBORD591
Bb MBORD675
Bb Bbr77
B. bronchiseptica
Complex IV
Bpp 12822
Bpp Bpp5
B. parapertussis
Bb MBORD635
Bb MBORD624
Bb SO10328
Bb MBORD731
Bb M85/00/2
Bb RB50
Bb D989
Bb D756
Bb RB630
Bb 3E44
Bb 980-2
Bb C3
Bb MBORD762
Bb MBORD678
Bb MBORD681
Bb MBORD698
Bb MBORD785
Bb KM22
Bb CARE970018BB
Bb MBORD670
Bb MBORD849
Bb MBORD665
Bb MBORD668
Bb M435/02/3
Bb A1-7
Bb C4
Bb MBORD595
Bb MBORD632
Bb 1289
Bb MBORD782
Bb E010
Bb MBORD839
Bb D993
Bb 00-P-2796
Bb E013
Bb E012
Bb 253
B. bronchiseptica
Complex I
Clade A
Bt H044680328
Bt NCTC12995
Bt H064000408
B trematum CCUG 13902
B. trematum
Bhz 1277
Bhz L60
Bhz 4161
Bhz CA90 BAL1384
Bhz 5132
Bhz OH87 BAL007II
B. hinzii
B avium 197N
B. avium
Bhl F627
Bhl 04P3421
Bhl 44057
Bhl 30539
Bhl 70147
Bhl H620
Bhl CDC-H635-BH
Bhl CDC-H572-BH
Bhl H558
Bhl CDC-H585-BH
Bhl 41130
Bhl CDC-H809-BH
Bhl CDC-H629-BH
Bhl 1058
Bhl CDC-H643-BH
Bhl CDC-H719-BH
Bhl CDC-H785-BH
Bhl 35009
B. holmesii
Clade B
Ban H050680373
Ban NCTC13364
B. ansorpii
Bpet J51
Bpet J49
Bpet DSM 12804
B. petrii
Clade C
0.01
B
gene presence/absence-based tree
Bp CHLA-11
Bp CHLA-13
Bp STO1-CHLA-0006
Bp STO1-CHOC-0021
Bp H973
Bp CHLA-26
Bp STO1-SEAT-0006
Bp STO1-CHOC-0016
Bp I002
Bp STO1-SEAT-0007
Bp STO1-CHOC-0018
Bp STO1-CHOC-0017
Bp STO1-CHOC-0019
Bp H939
Bp STO1-CHOM-0012
Bp I176
Bp I036
Bp H918
Bp H921
Bp STO1-CHLA-0011
Bp H897
Bp H934
Bp B200
Bp 2250905
Bp 2371640
Bp 2356847
Bp STO1-CNMC-0004
Bp CHLA-20
Bp STO1-CHOC-0008
Bp CHLA-15
Bp STO1-SEAT-0004
Bp CS
Bp 18323
Bp Tohama I
B. pertussis
Bpp Bpp5
Bpp 12822
B. parapertussis
Bb RB50
Bb 253
Bb 1289
Bb KM22
Bb MBORD849
Bb MBORD785
Bb CARE970018BB
Bb MBORD670
Bb E010
Bb D993
Bb MBORD839
Bb E013
Bb E012
Bb SO10328
Bb MBORD731
Bb RB630
Bb 3E44
Bb D989
Bb D756
Bb 980-2
Bb C4
Bb A1-7
Bb M85/00/2
Bb MBORD632
Bb MBORD762
Bb C3
Bb MBORD624
Bb MBORD595
Bb MBORD668
Bb MBORD665
Bb MBORD635
Bb M435/02/3
Bb MBORD698
Bb MBORD681
Bb MBORD782
Bb MBORD678
Bb F2
Bb F-1
Bb 00-P-2796
Bb OSU553
Bb 00-P-2730
Bb MBORD707
Bb GA96-01
Bb MO275
Bb MO211
Bb OSU095
Bb CA90 BB1334
Bb OSU054
Bb Bbr77
Bb MBORD675
Bb MO149
Bb MBORD591
Bb 7E71
Bb MBORD901
Bb E014
Bb SBL-F6116
Bb F4563
Bb CA90 BB02
B. bron-
chiseptica
Clade A
B. avium
Bt CCUG 13902
Bt H064000408
Bt H044680328
Bt NCTC12995
B avium 197N
B. trematum
Bhz 5132
Bhz OH87 BAL007II
Bhz CA90 BAL1384
Bhz 1277
Bhz 4161
Bhz L60
B. hinzii
Bhl F627
Bhl H558
Bhl 44057
Bhl H620
Bhl CDC-H629-BH
Bhl 1058
Bhl 30539
Bhl 41130
Bhl 35009
Bhl 70147
Bhl CDC-H572-BH
Bhl 04P3421
Bhl CDC-H719-BH
Bhl CDC-H785-BH
Bhl CDC-H635-BH
Bhl CDC-H643-BH
Bhl CDC-H585-BH
Bhl CDC-H809-BH
B. holmesii
Clade B
Ban H050680373
Ban NCTC13364
B. ansorpii
Bpet J51
Bpet DSM 12804
Bpet J49
B. petrii
Clade C
500

Neighbor-joining trees of 16S rRNA gene sequences and 8 concatenated ATP synthase proteins from *Bordetella*



- **R: Calculate Distance matrices of SNPs and Genes**
- **R: Calculate Mantel correlation between 2 phylogenies**

```
a <-read.table(“fake5_gene1”, header = TRUE, sep = “\t”
## load data gene 1
x = t(a) ## transform data to genomes by row and SNPs by col
SEQ1.dist <- as.dist(dist(x, “manhattan”)) ## calc matrix
m1 <- as.table(SEQ1.dist) ## write as table
#############################################################
z <-read.table(“fake5_gene2”, header = TRUE, sep = “\t”
## load data gene 2
y = t(z) ## transform data to genomes by row and SNPs by col
SEQ2.dist <- as.dist(dist(y, “manhattan”)) ## calc matrix
m2 <- as.table(SEQ2.dist) ## write as table
```

- –**R: Calculate Distance matrices of SNPs and Genes**
- –**R: Calculate Mantel correlation between 2 phylogenies**

```
m3 <-mantel.rtest(SEQ1.dist, SEQ2.dist, nrepet = 99999)

fileConn <- file(“output.txt”)

write.lines(paste(m3[2:4], sep = “\t”), fileConn)

close fileConn
```

```
 cat output.txt
```

extract values from output.txt

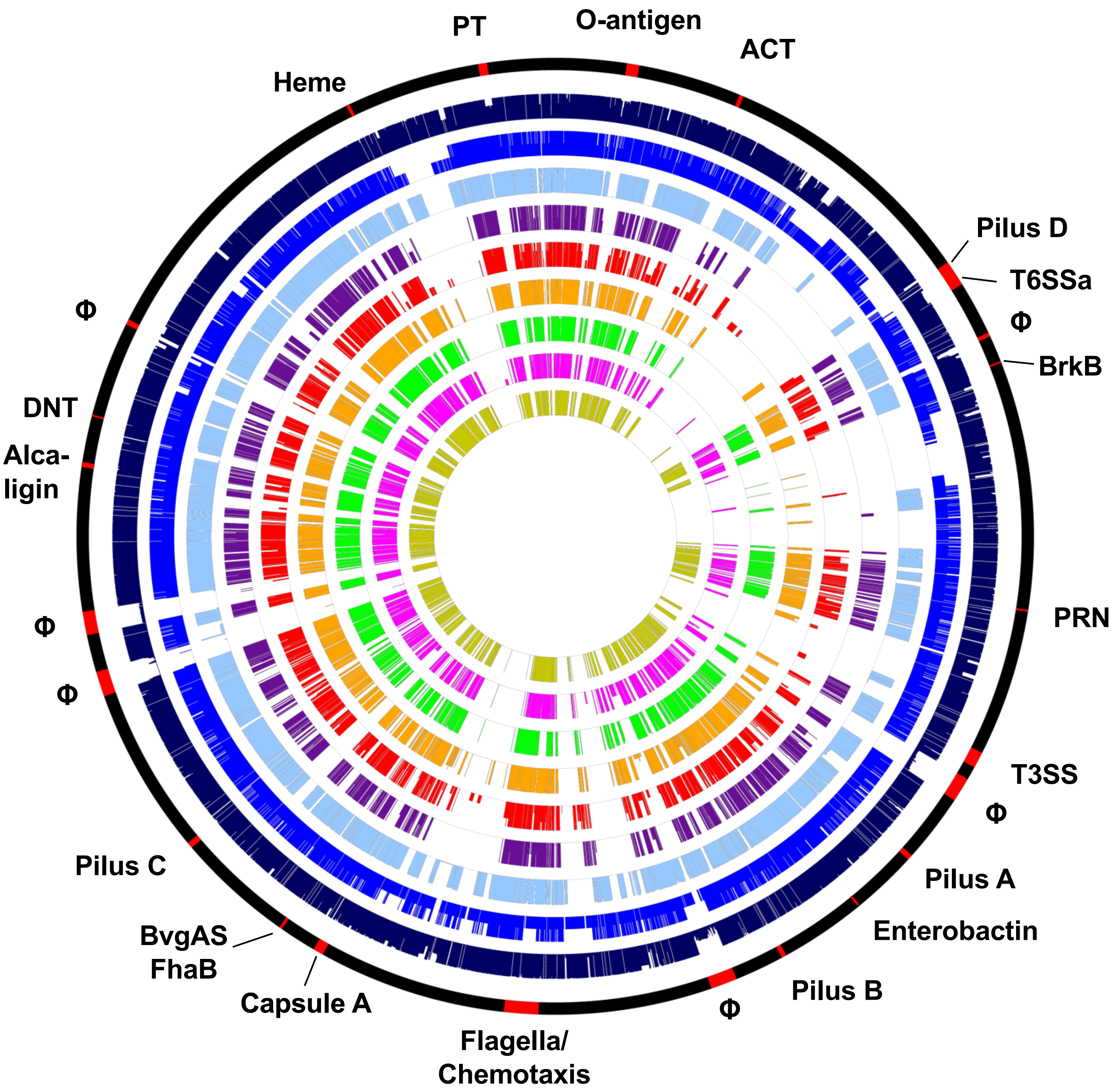
```
 cat output.txt | awk ‘NR==1’ > t1

 cat output.txt | awk ‘NR==2’ > t2

 cat output.txt | awk ‘NR==3’ > t3

 printf “r = $(cat t1) \n nrepet = $(cat t2) \n p-value = $(cat
 t3) \n” >> $NAMEGENE1-$NAMEGENE2.out
```

extract values from output.txt

```
cat output.txt | awk ‘NR==1’ > t1

cat output.txt | awk ‘NR==2’ > t2

cat output.txt | awk ‘NR==3’ > t3

printf “r = $(cat t1) \n nrepet = $(cat t2) \n p-value = $(cat t3) \n” >> $NAMEGENE1-$NAMEGENE2.out
```

```
cat $NAMEGENE1-$NAMEGENE2.out

Dataset from file ‘9BordetellaSNP’: 265372 loci.

Dataset from file ‘ATPsynthase_AA’: 2125 loci.

r = 0.65755                  # R^2 = 0.4324

nrepet = 99999

p-value = 0.00483
```

Presence and absence of genes in 128 genomes from 9 *Bordetella* species



Virtual chromosome of the *B. bronchiseptica* RB50 reference genome with key factor genes or gene clusters in red.

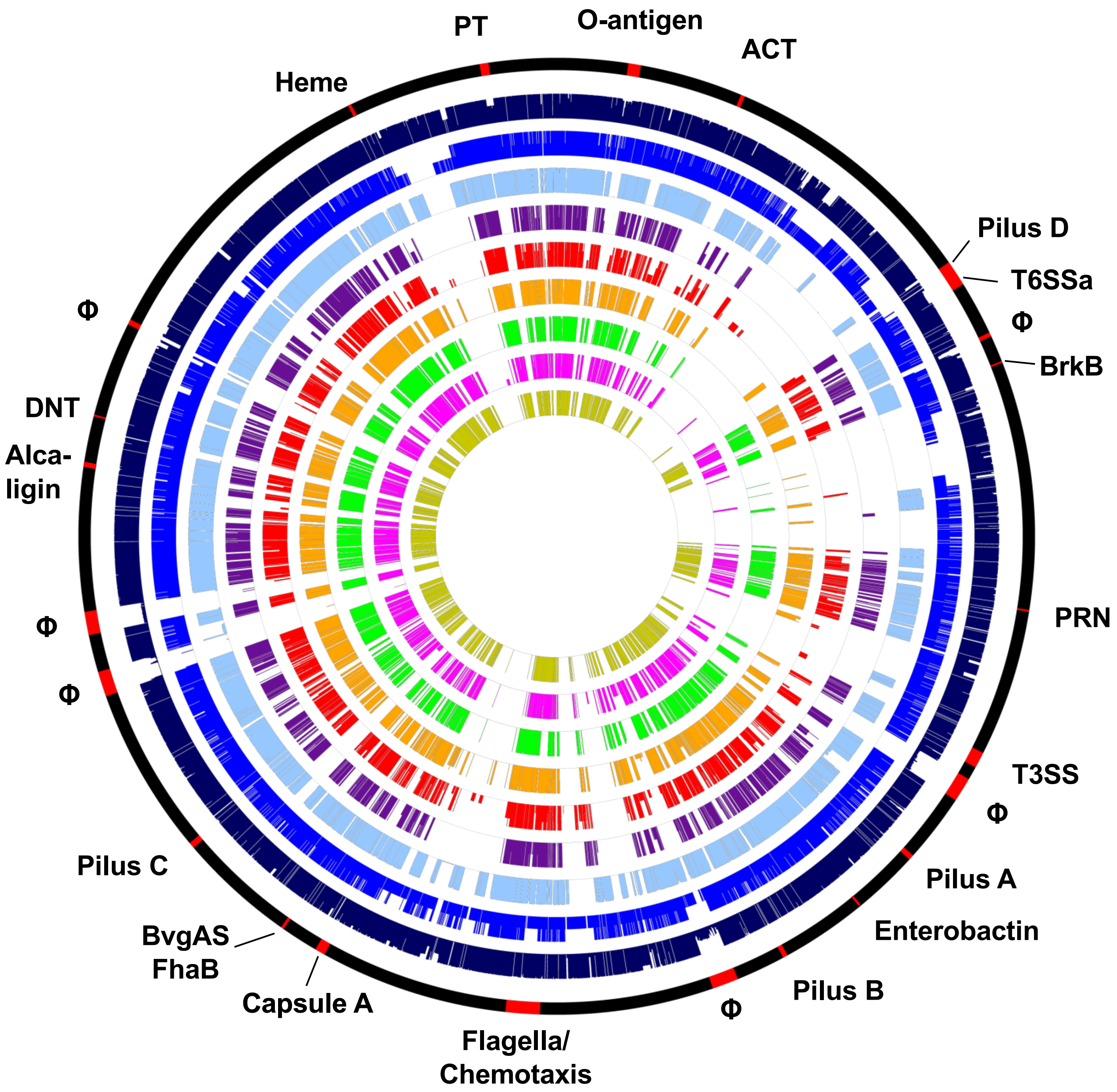
Proportion of genes present in individual genomes per species color-coded by species.

A thin line for each gene indicates the percentage of genomes in each species containing this gene.

colored: gene(s) present
white: gene(s) absent

Φ – prophage

Presence and absence of genes in 128 genomes from 9 *Bordetella* species



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);

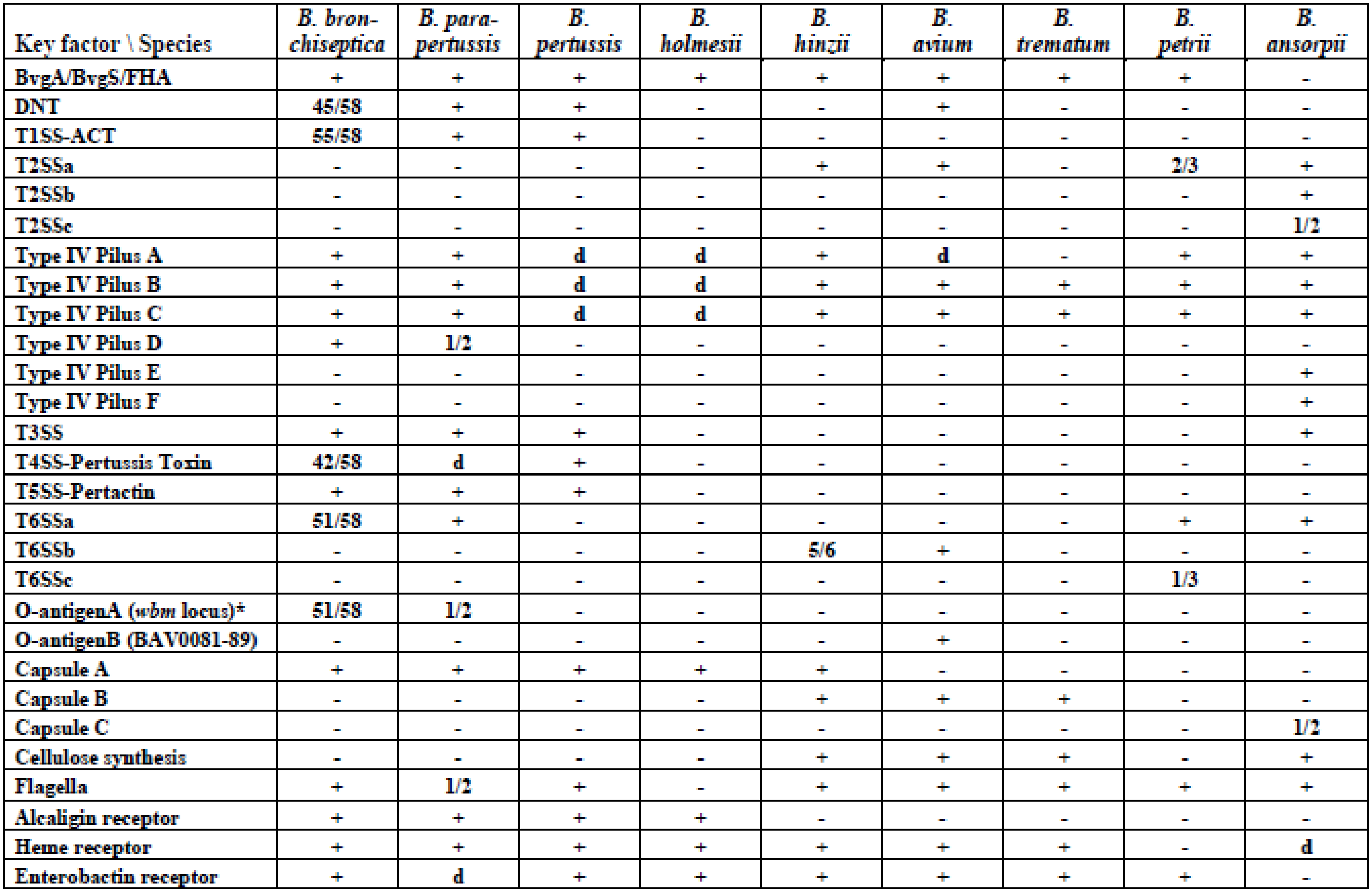
7: *B. hinzii* (6);

8: *B. holmesii* (18);

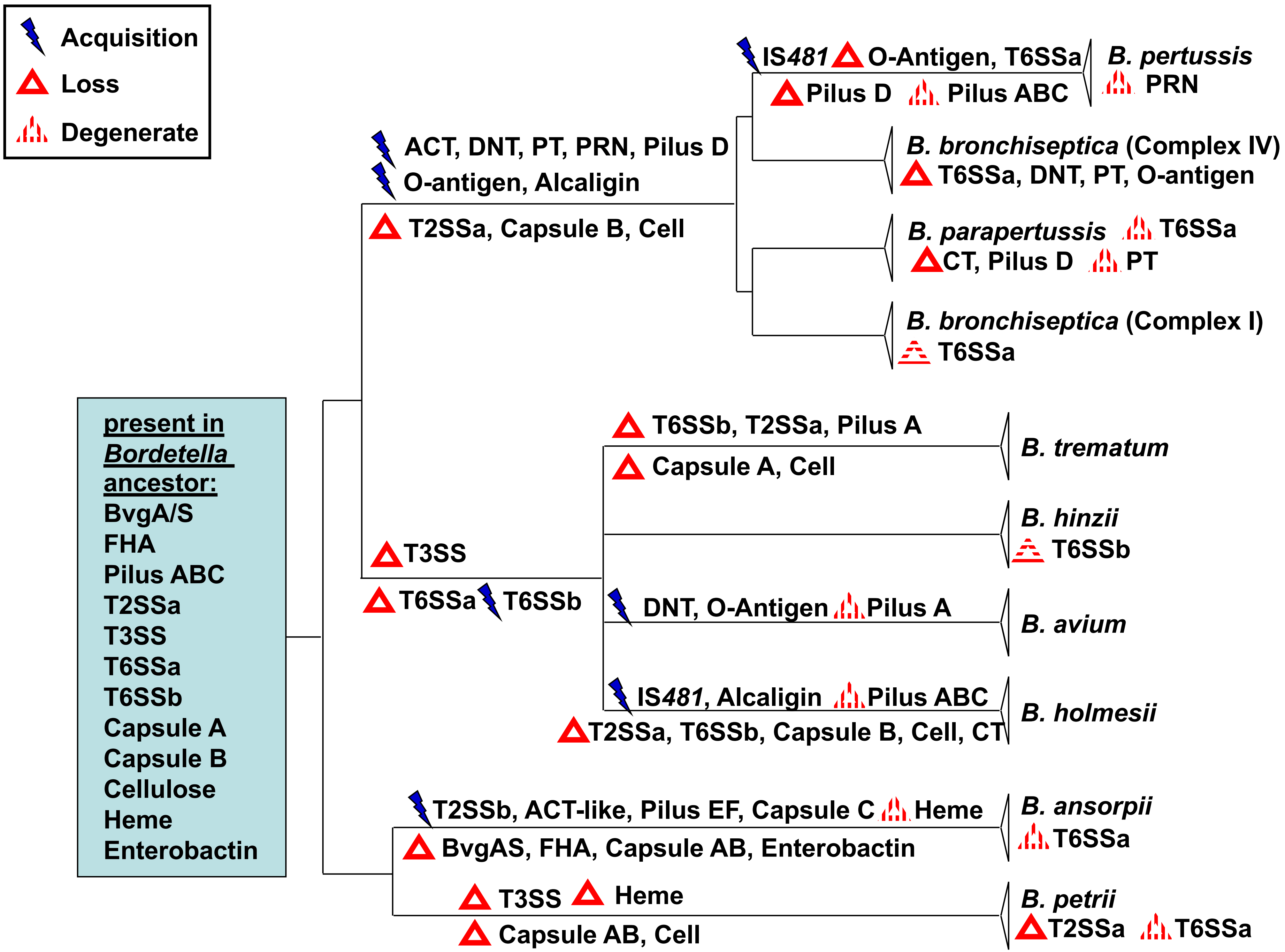
9: *B. trematum* (4);

10: *B. avium* (1)

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
⚡ IS481 △ 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
⚡ IS481, 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

```r
rm(list = ls())

library(gplots)

library(gdata)

library(gtools)


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
```

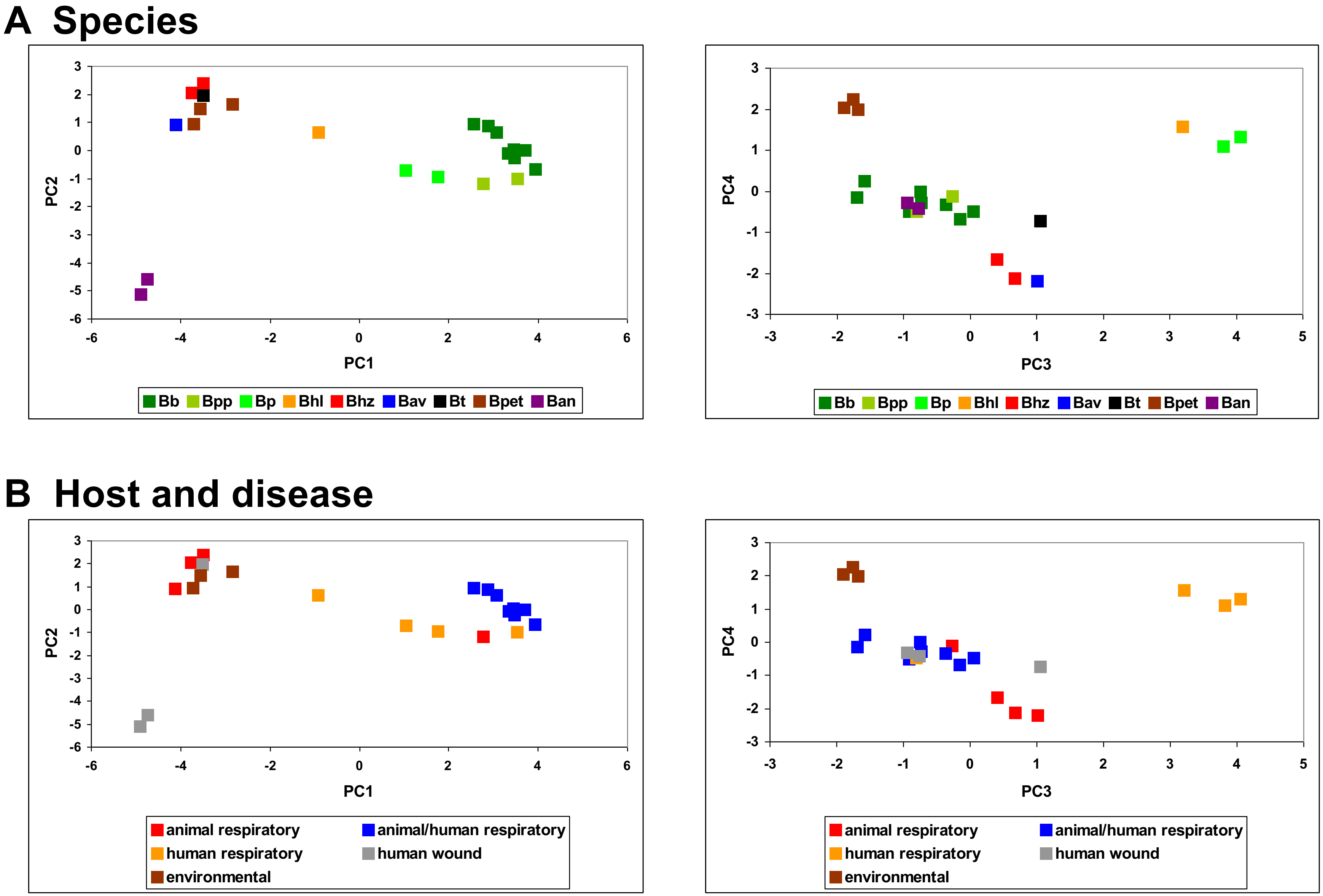
# Let’s walk through:

```
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

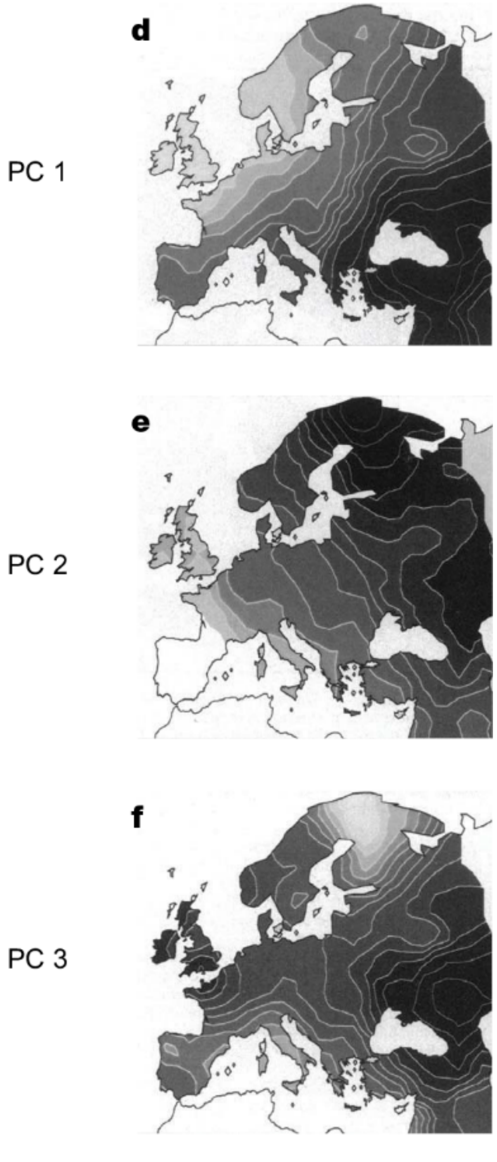
	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8	Comp.9	Comp.10	Comp.11	Comp.12	Comp.13	Comp.14	Comp.15	Comp.16	Comp.17
B.bronch1	3.940976	-0.65934	-0.35932	-0.33097	-0.78523	-0.63582	0.106812	-0.33411	0.251795	-0.83729	0.111922	-0.15431	0.170636	-0.08216	0.037813	-0.00413	0.001747
B.bronch2	3.467985	-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

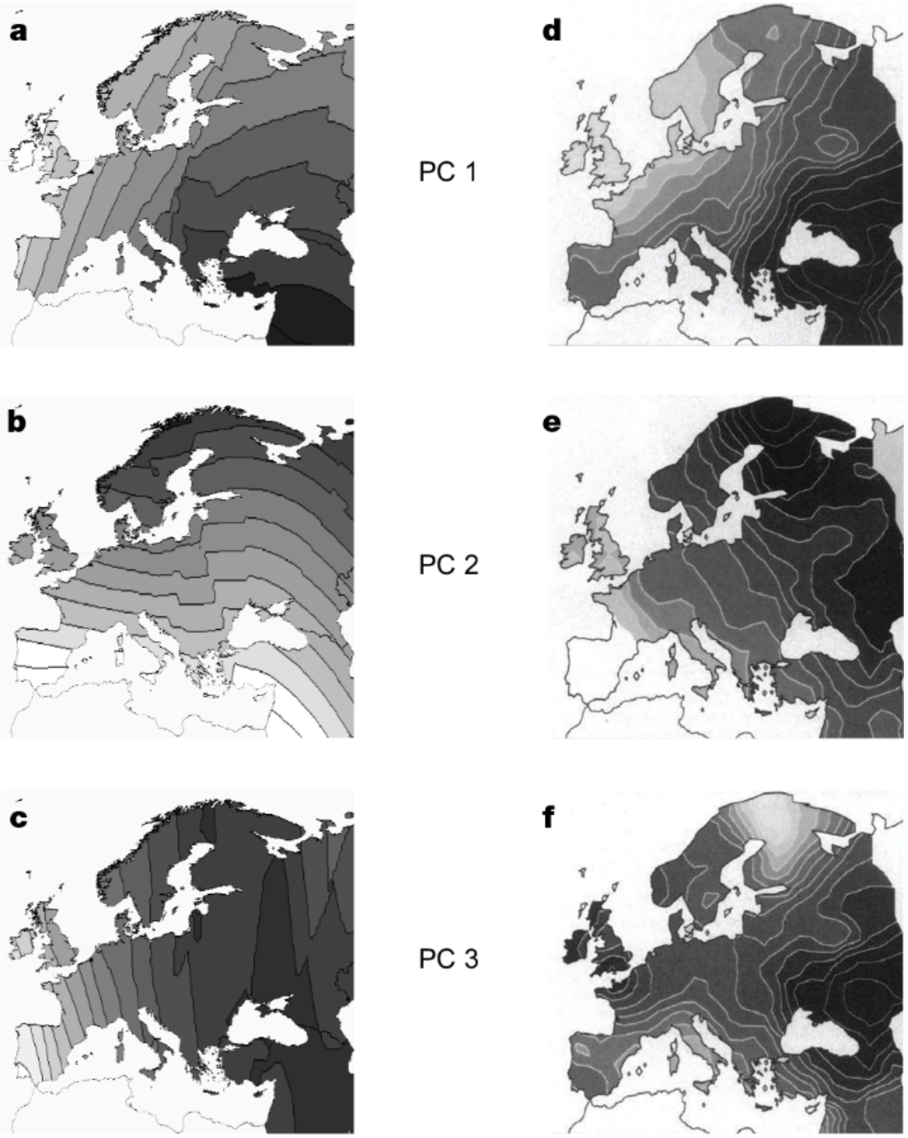


Clinal gradients in principal components 1–3 in allozyme allele frequencies in Europeans

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

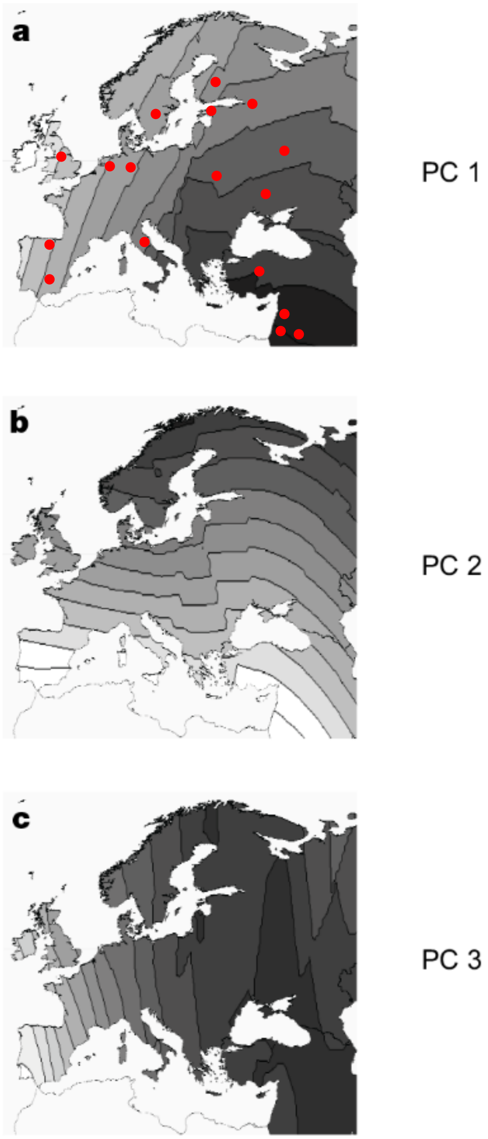


Clinal gradients in principal components 1–3 in allozyme allele frequencies in Europeans

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

To be continued …

Questions?