

# SESSION 14. PRACTICE

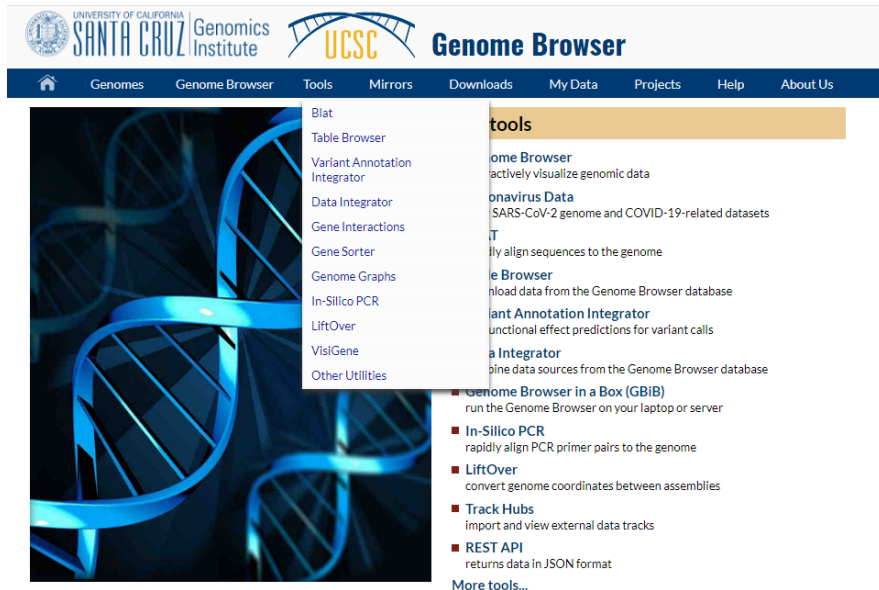
Personal genomes:  
The differences between you and me

# Counting SNPs

- Using the table browser at the UCSC genome database (<http://genome.ucsc.edu/cgi-bin/hgTables?org=human>)
- Comparing chr4 of eight different human individuals
  - (1) YanHuang (Han Chinese individual, anonymous)
  - (2) Seong-Jin Kim (Korean)
  - (3) James Watson
  - (4) Craig Venter
  - (5) YRI NA18507 (Yoruba, anonymous of the 1000 Genomes Project)
  - (6) NA12891 (Central European origin, anonymous of the 1000 Genomes Project)
  - (7) ABT (Desmond Tutu)
  - (8) KB1, Khoisan/Bushmen individual

# Counting SNPs

<http://genome.ucsc.edu/index.html>



The screenshot shows the UCSC Genome Browser website. The header includes the University of California Santa Cruz Genomics Institute logo and the UCSC Genome Browser title. A navigation bar contains links for Genomes, Genome Browser, Tools, Mirrors, Downloads, My Data, Projects, Help, and About Us. A dropdown menu for 'Tools' is open, listing various utilities. A secondary list of tools is visible on the right side of the page, including Genome Browser, Coronavirus Data, BLAT, Table Browser, Variant Annotation Integrator, Data Integrator, Gene Interactions, Gene Sorter, Genome Graphs, In-Silico PCR, LiftOver, VisiGene, and Other Utilities. A 'More tools...' link is also present.

UCSC Genome Browser

Genomes Genome Browser Tools Mirrors Downloads My Data Projects Help About Us

Tools

- Blat
- Table Browser
- Variant Annotation Integrator
- Data Integrator
- Gene Interactions
- Gene Sorter
- Genome Graphs
- In-Silico PCR
- LiftOver
- VisiGene
- Other Utilities

tools

- Genome Browser
  - actively visualize genomic data
- Coronavirus Data
  - SARS-CoV-2 genome and COVID-19-related datasets
- BLAT
  - rapidly align sequences to the genome
- Table Browser
  - upload data from the Genome Browser database
- Variant Annotation Integrator
  - functional effect predictions for variant calls
- Data Integrator
  - combine data sources from the Genome Browser database
- Genome Browser in a Box (GBiB)
  - run the Genome Browser on your laptop or server
- In-Silico PCR
  - rapidly align PCR primer pairs to the genome
- LiftOver
  - convert genome coordinates between assemblies
- Track Hubs
  - import and view external data tracks
- REST API
  - returns data in JSON format

More tools...

# Counting SNPs

Genomes Genome Browser Tools Mirrors Downloads My Data Projects Help About Us

## Table Browser

Use this program to retrieve the data associated with a track in text format, to calculate intersections between tracks, and to retrieve DNA sequence covered by a track. For help in using this application see [Using the Table Browser](#) for a description of the controls in this form, and the [User's Guide](#) for general information and sample queries. For more complex queries, you may want to use [Galaxy](#) or our [public MySQL server](#). To examine the biological function of your set through annotation enrichments, send the data to [GREAT](#). Refer to the [Credits](#) page for the list of contributors and usage restrictions associated with these data. All tables can be downloaded in their entirety from the [Sequence and Annotation Downloads](#) page.

**clade:**  **genome:**  **assembly:**

**group:**  **track:**

**table:**

**region:**  genome  position

**identifiers (names/accessions):**

**filter:**

**intersection:**

**correlation:**

**output format:**  Send output to  [Galaxy](#)  [GREAT](#)

**output file:**  (leave blank to keep output in browser)

**file type returned:**  plain text  gzip compressed

To reset **all** user cart settings (including custom tracks), [click here](#).

# Basic Shell Commands

```
$ cd [User_Folder]
$ mkdir Session14
$ cd Session14
```

# Counting SNPs

```
$ cp /home/biguser/tutor/Session14/snp.txt .  
$ less snp.txt
```

```
3263 A A A A A A A A T  
3351 T W W W T T W W A  
3544 T T T T T T T Y T  
3567 T T T T T T T Y T  
3774 K G T T T T T T T  
4131 G K G G G G G G T  
4190 A A A R A A A R A  
4306 T T T T T T T T C  
4371 C Y Y Y C C C C C  
4489 G R A A A A A A A  
6394 T T T T T T T T C  
6523 G R A R A A R A A
```

# Counting SNPs

38357	T	T	Y	T	T	T	Y	Y	T
38368	G	G	G	G	G	G	G	G	C
38392	T	T	T	T	T	T	T	T	C

- The first column is position
- Nucleotides from eight individuals
- The last column is the nucleotide of chimpanzee
- Positions where at least one genome has an unknown base have been removed
- Positions containing the same nucleotide in all nine genomes have been removed

# Counting SNPs

3263	A	A	A	A	A	A	A	A	T
3351	T	W	W	W	T	T	W	W	A
3544	T	T	T	T	T	T	T	Y	T
3567	T	T	T	T	T	T	T	Y	T
3774	K	G	T	T	T	T	T	T	T
4131	G	K	G	G	G	G	G	G	T
4190	A	A	A	R	A	A	A	R	A
4306	T	T	T	T	T	T	T	T	C

IUPAC nucleotide code	Base
A	Adenine
C	Cytosine
G	Guanine
T (or U)	Thymine (or Uracil)
R	A or G
Y	C or T
S	G or C
W	A or T
K	G or T
M	A or C
B	C or G or T
D	A or G or T
H	A or C or T
V	A or C or G
N	any base
. or -	gap



# Counting SNPs

Create a distance matrix from SNPs of 9 genomes

```
$ vi snp.py
```

```
#!/usr/bin/python

# obtain pairwise distances from snp data,
# counting sites where at least one allele is different

import re

humans = [
    # SNPs appear in the SNP data file in columns in this order
    'YH',      # Han chinese
    'SJK',     # Seong-Jin Kim
    'JW',     # James Watson
    'CV',     # Craig Venter
    'NA18507', # Yoruban of 1000 Genomes project
    'NA12891', # Of Central European origin
    'ABT',    # Archbishop Desmond Tutu
    'KB1',    # Bushmen individual
    'chimp'   # chimpanzee
]
```

# Counting SNPs

```
# 1 #  
# initialize the distance matrix with zero values  
# for the diagonal cells  
diff = []  
for i in range(0, 10):  
    diff.append([])  
    for j in range(0, 10):  
        diff[i].append(0)  
print diff
```

```
$ python snp.py
```

```
[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
```



# Counting SNPs

```
# read the snp data from file
for line in open('snp.txt'):
    line = line.rstrip()
    columns = re.split(' ', line)
    # 2 #
    for i in range(1, 9):
        for j in range(i + 1, 10):
            # 3 #
            if columns[i] != columns[j]:
                diff[i][j] += 1
            # 4 #
            # to produce a symmetric matrix
            diff[j][i] += 1

print diff
```

```
$ python snp.py
```

```
[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 44597, 53594, 53913, 67914, 53710, 68837, 77272, 593367],
 [0, 44597, 0, 54192, 54537, 68826, 55281, 69404, 76929, 593496], [0, 53594, 54192, 0, 50859, 702
84, 51260, 70256, 77590, 592751], [0, 53913, 54537, 50859, 0, 70149, 51009, 69659, 77369, 592632]
, [0, 67914, 68826, 70284, 70149, 0, 69245, 70057, 79508, 599102], [0, 53710, 55281, 51260, 51009
, 69245, 0, 69941, 78130, 594831], [0, 68837, 69404, 70256, 69659, 70057, 69941, 0, 77707, 599292
], [0, 77272, 76929, 77590, 77369, 79508, 78130, 77707, 0, 600776], [0, 593367, 593496, 592751, 5
92632, 599102, 594831, 599292, 600776, 0]]
```

# Counting SNPs

```
# 5 #
# print a header for PHYLIP format
# with the number of species
print ' ', '9'

# print the matrix data
for i in range(1, 10):
    # 6 #
    txt = humans[i - 1]
    txt = txt[0:7]
    print txt,
    length = 10 - len(txt)
    short = ' ' * (length - 2)
    print short,
    for j in range(1, 10):
        print diff[i][j],
    print ''
```

이전 코드 부분에 있었던 print 명령어  
앞에 #을 붙여서 비활성화를 꼭 시켜주세요!

# Counting SNPs

```
$ python snp.py
```

```
YH SJK JW CV NA18507 NA12891 ABT KB1 chimp
```

YH	0	44597	53594	53913	67914	53710	68837	77272	593367
SJK	44597	0	54192	54537	68826	55281	69404	76929	593496
JW	53594	54192	0	50859	70284	51260	70256	77590	592751
CV	53913	54537	50859	0	70149	51009	69659	77369	592632
NA18507	67914	68826	70284	70149	0	69245	70057	79508	599102
NA12891	53710	55281	51260	51009	69245	0	69941	78130	594831
ABT	68837	69404	70256	69659	70057	69941	0	77707	599292
KB1	77272	76929	77590	77369	79508	78130	77707	0	600776
chimp	593367	593496	592751	592632	599102	594831	599292	600776	0

```
$ python snp.py > snp.out
```

# Phylip package - neighbor

- Phylip package

(<http://evolution.genetics.washington.edu/phylip.html>)



## PHYLP

A new release of PHYLP, version 3.696, is now available as source code. This release differs only in its license -- it has an open source license, so that PHYLP can be distributed with other software that has commercial licenses or has a restrictive open-source source license. Executables are currently at version 3.695, with the old license, but I will update them soon.

PHYLP is a *free* package of programs for inferring phylogenies. It is distributed as source code, documentation files, and a number of different types of executables. These Web pages, by [Joe Felsenstein](#) of the [Department of Genome Sciences](#) and the [Department of Biology](#) at the [University of Washington](#), contain information on PHYLP and ways to transfer the executables, source code and documentation to your computer.

- [A general description](#) of PHYLP.
- [Programs](#) in the PHYLP package
- About the [Executables](#)
- About the [Source code](#) ... compiling it yourself
- The documentation web pages for PHYLP can be read [here](#)
- [Get me PHYLP](#) (version 3.695)
- [How to install PHYLP](#)
- [Frequently asked questions](#)
- PHYLP's [Facebook page](#) for discussing problems.
- An excellent guide to using PHYLP with molecular data is available [here](#).
- [PHYLP on the web](#) (HTML documentation, server services)
- [Current and future versions of PHYLP \(including new features\)](#)
- [Older versions of PHYLP, including version 3.5](#)
- [Bugs in the package, known or recently fixed](#)
- [Phylogeny programs](#) available elsewhere
- [Credits \(people, grants etc.\)](#)

# Phylip package - neighbor

```
neighbor: can't find input file "infile"  
Please enter a new file name> snp.out
```

```
Neighbor-Joining/UPGMA method version 3.695
```

```
Settings for this run:
```

```
N      Neighbor-joining or UPGMA tree?  Neighbor-joining  
0      Outgroup root?  No, use as outgroup species  1  
L      Lower-triangular data matrix?  No  
R      Upper-triangular data matrix?  No  
S      Subreplicates?  No  
J      Randomize input order of species?  No. Use input order  
M      Analyze multiple data sets?  No  
0      Terminal type (IBM PC, ANSI, none)?  ANSI  
1      Print out the data at start of run  No  
2      Print indications of progress of run  Yes  
3      Print out tree  Yes  
4      Write out trees onto tree file?  Yes
```

```
Y to accept these or type the letter for one to change
```

```
N
```

# Phylip package - neighbor

Settings for this run:

```
N Neighbor-joining or UPGMA tree? UPGMA
L Lower-triangular data matrix? No
R Upper-triangular data matrix? No
S Subreplicates? No
J Randomize input order of species? No. Use input order
M Analyze multiple data sets? No
0 Terminal type (IBM PC, ANSI, none)? ANSI
1 Print out the data at start of run No
2 Print indications of progress of run Yes
3 Print out tree Yes
4 Write out trees onto tree file? Yes
```

Y to accept these or type the letter for one to change

Y

```
Cycle 8: species 1 (22298.50000) joins species 2 (22298.50000)
Cycle 7: species 3 (25429.50000) joins species 4 (25429.50000)
Cycle 6: node 3 ( 137.75000) joins species 6 (25567.25000)
Cycle 5: node 1 (4803.75000) joins node 3 (1535.00000)
Cycle 4: node 1 (7539.55000) joins species 5 (34641.80000)
Cycle 3: node 1 ( 204.36667) joins species 7 (34846.16667)
Cycle 2: node 1 (4047.04762) joins species 8 (38893.21429)
Cycle 1: node 1 (258997.22321) joins species 9 (297890.43750)
```

Output written on file "outfile"

Tree written on file "outtree"

Done.

outtree

```
(((((YH:22298.50000,SJK:
22298.50000):4803.75000,
((JW:25429.50000,
CV:25429.50000):
137.75000,NA12891:25567.2
5000):1535.00000):
7539.55000,
NA18507:34641.80000):
204.36667,ABT:
34846.16667):4047.04762,
KB1:38893.21429):
258997.22321,chimp:
297890.43750);
```



# NJplot

- NJplot

(<http://doua.prabi.fr/software/njplot>)

## NJplot

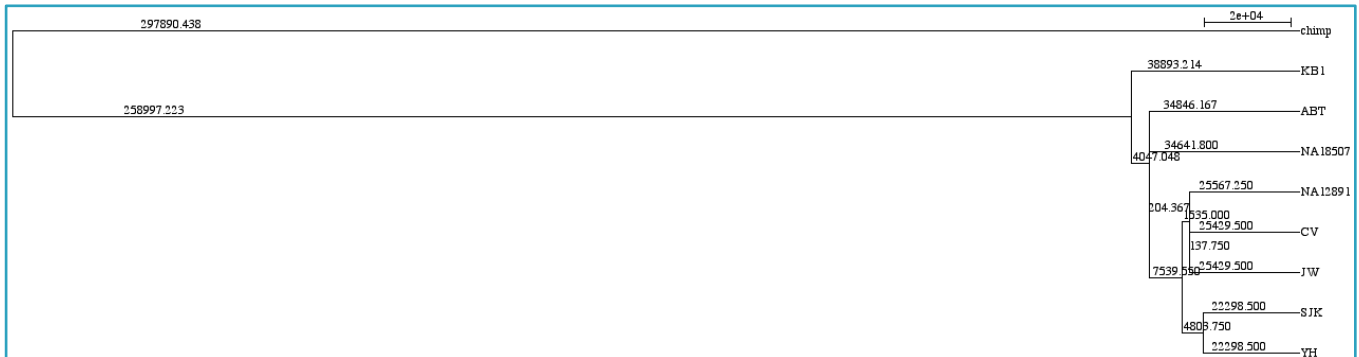
NEW: NJplot plots trees in PDF and PostScript formats (not for MacOS).

NEW: NJplot allows to open several tree windows.

NEW: NJplot can draw multibranching trees with or without branch lengths.

NJplot is a tree drawing program able to draw any phylogenetic tree expressed in the [Newick](#) phylogenetic tree format (e.g., the format used by the PHYLIP package). NJplot is especially convenient for rooting the unrooted trees obtained from parsimony, distance or maximum likelihood tree-building methods.

A screen shot of the main window of njplot is available [here](#).



# Exercise

- The file `'chr4snp.txt'` is a list of SNPs in the human chromosome 4, according to dbSNP build 130. Write a Python script that will list the SNPs (positions) that are present in this file but that are not found in the file `'snp.txt'`. The file `'chr4snp.txt'` uses 'zero-based' numbering. From a practical point of view, this means that the third column positions in that file are comparable to the position numbers in `'snp.txt'`.

```
cp /home/biguser/tutor/Session14/chr4snp.txt .
```

# Exercise

3263	A	A	A	A	A	A	A	T	
3351	T	W	W	W	T	T	W	W	A
3544	T	T	T	T	T	T	T	Y	T
3567	T	T	T	T	T	T	T	Y	T
3774	K	G	T	T	T	T	T	T	T
4131	G	K	G	G	G	G	G	G	T
4190	A	A	A	R	A	A	A	R	A
4306	T	T	T	T	T	T	T	T	C
4371	C	Y	Y	Y	C	C	C	C	C
4489	G	R	A	A	A	A	A	A	A
6394	T	T	T	T	T	T	T	T	C
6523	G	R	A	R	A	A	R	A	A

Snp.txt

#chrom	chromStart	chromEnd	name
chr4	190	191	rs61793641
chr4	283	284	rs73217955
chr4	303	304	rs73791797
chr4	312	313	rs61793642
chr4	319	320	rs73217956
chr4	353	354	rs61793643
chr4	405	406	rs73217959
chr4	430	431	rs61793644
chr4	461	462	rs73217960
chr4	567	568	rs61793645
chr4	615	616	rs71614925
chr4	1298	1299	rs71614926
chr4	1359	1360	rs11944932
chr4	1450	1451	rs6842902
chr4	1525	1526	rs11735203
chr4	1596	1597	rs71614927
chr4	1636	1637	rs71602446
chr4	1688	1689	rs11248007
chr4	1717	1718	rs6827402
chr4	1796	1797	rs6827457
chr4	1881	1882	rs11735303
chr4	1946	1947	rs6819915
chr4	1960	1961	rs6847489
chr4	1983	1984	rs6819945
chr4	1985	1986	rs7686224

실제 position

chr4snp.txt

Chr4snp.txt 에 있는 snp 번호 snp.txt 에 있는 position을 불러 할 것!

# Exercise

```
import sys

infile1 = open(sys.argv[1], 'r') #'snp.txt'
infile2 = open(sys.argv[2], 'r') #'chr4snp.txt'

snppos = dict()
for line in infile1.readlines():
    columns = line.split(' ')
    pos = columns[0]
    snppos[pos] = ''
infile1.close()

for line in infile2.readlines():
    line = line.strip()
    if not line.startswith('#'):
        columns = line.split('\t')
        pos = columns[2]
        if not snppos.has_key(pos):
            print pos

infile2.close()
```

```
191262559
191262569
191262593
191262622
191262626
191262659
191262699
191262724
191262788
191262790
191262826
191262846
191262860
```