

CANB7640

Practical Workshop

Class 01

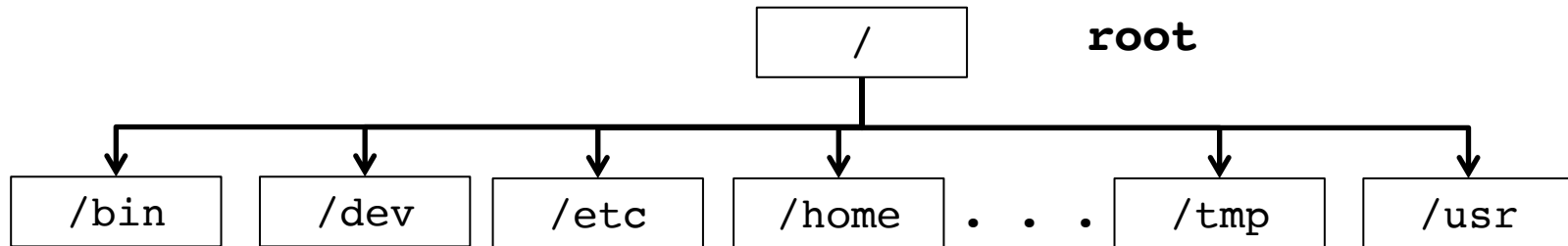
Aik Choon Tan, Ph.D.
Associate Professor of Bioinformatics
Division of Medical Oncology
Department of Medicine
aikchoon.tan@ucdenver.edu
8/29/2017

Unix Directory Structure

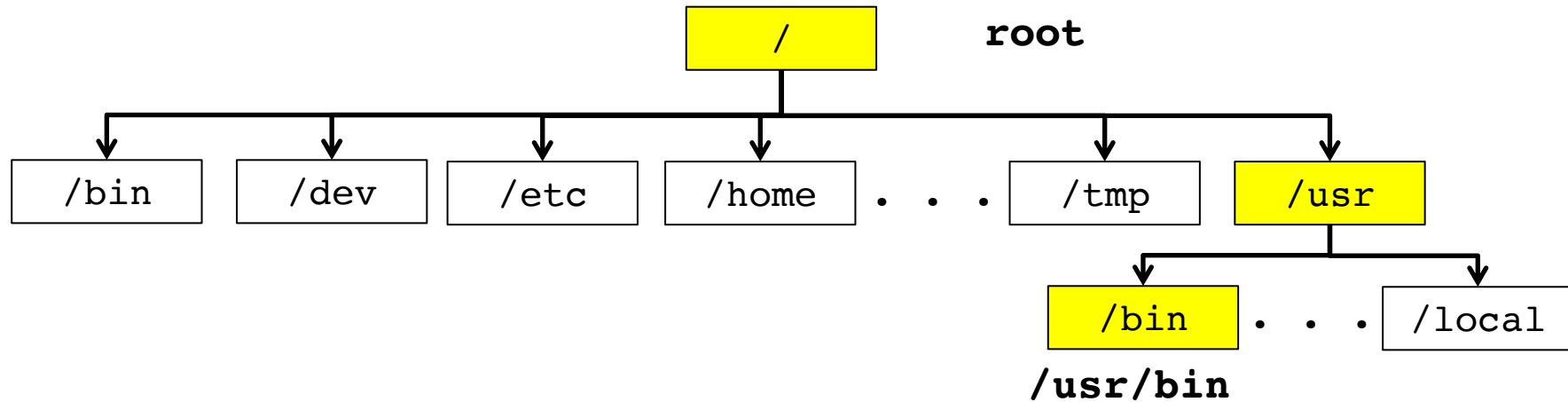
/

root

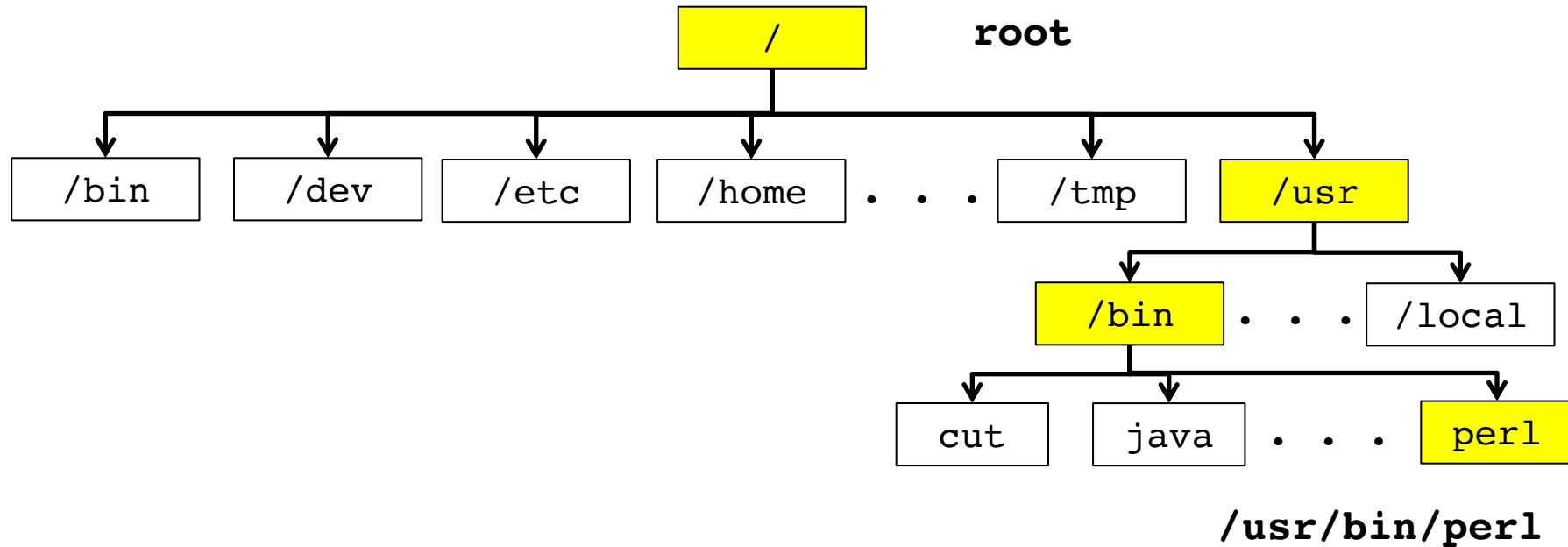
Unix Directory Structure



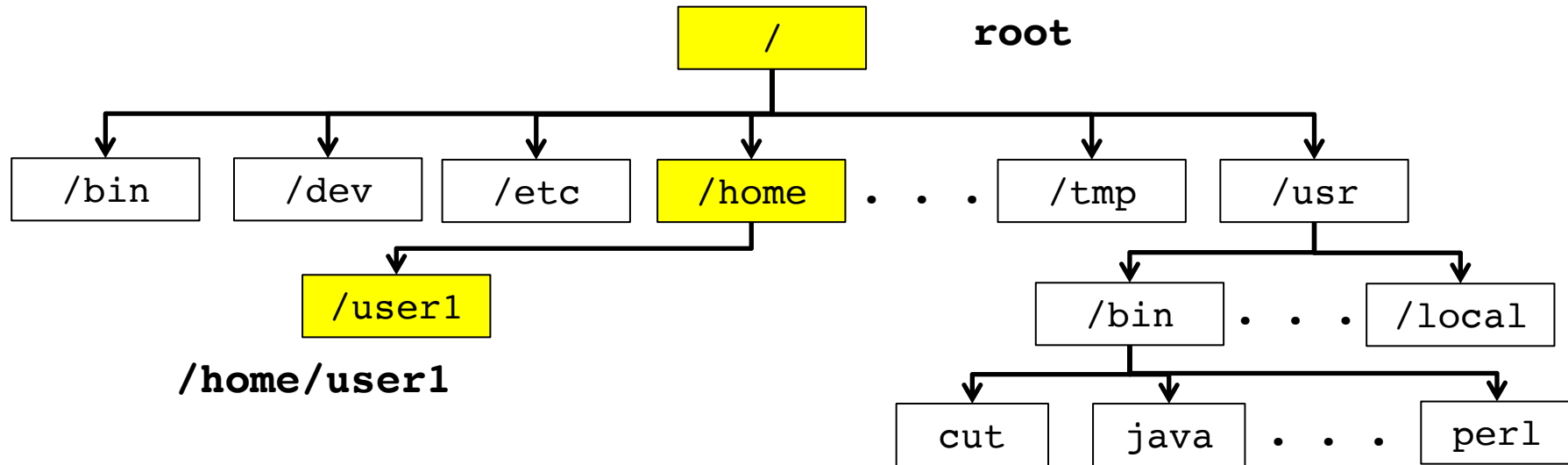
Unix Directory Structure



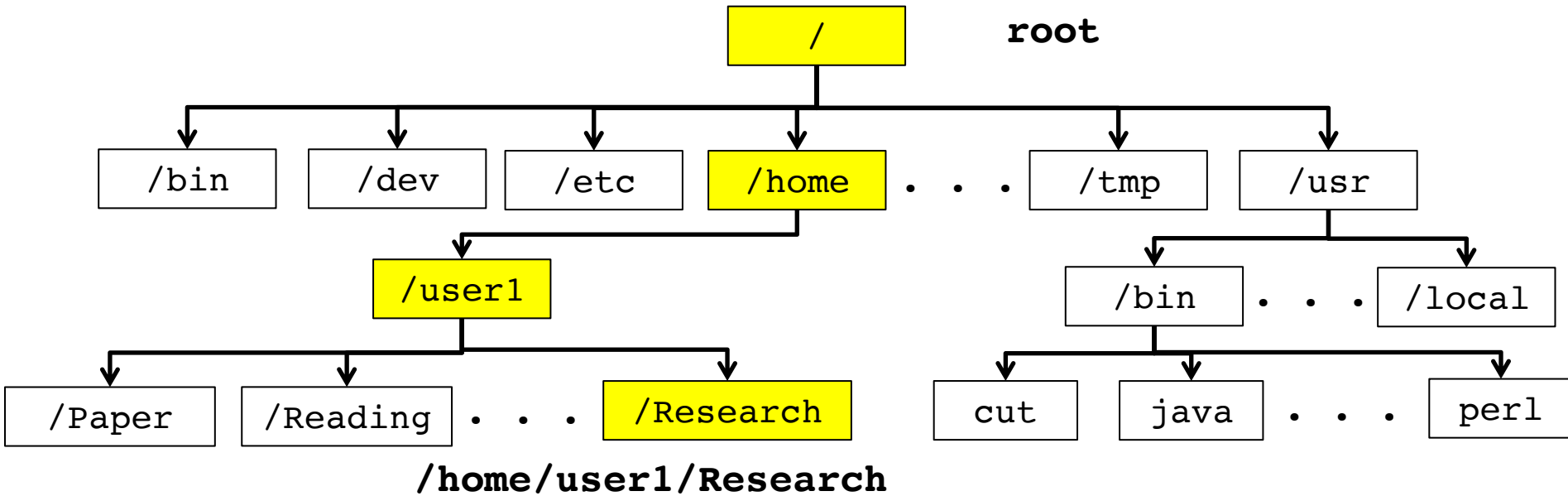
Unix Directory Structure



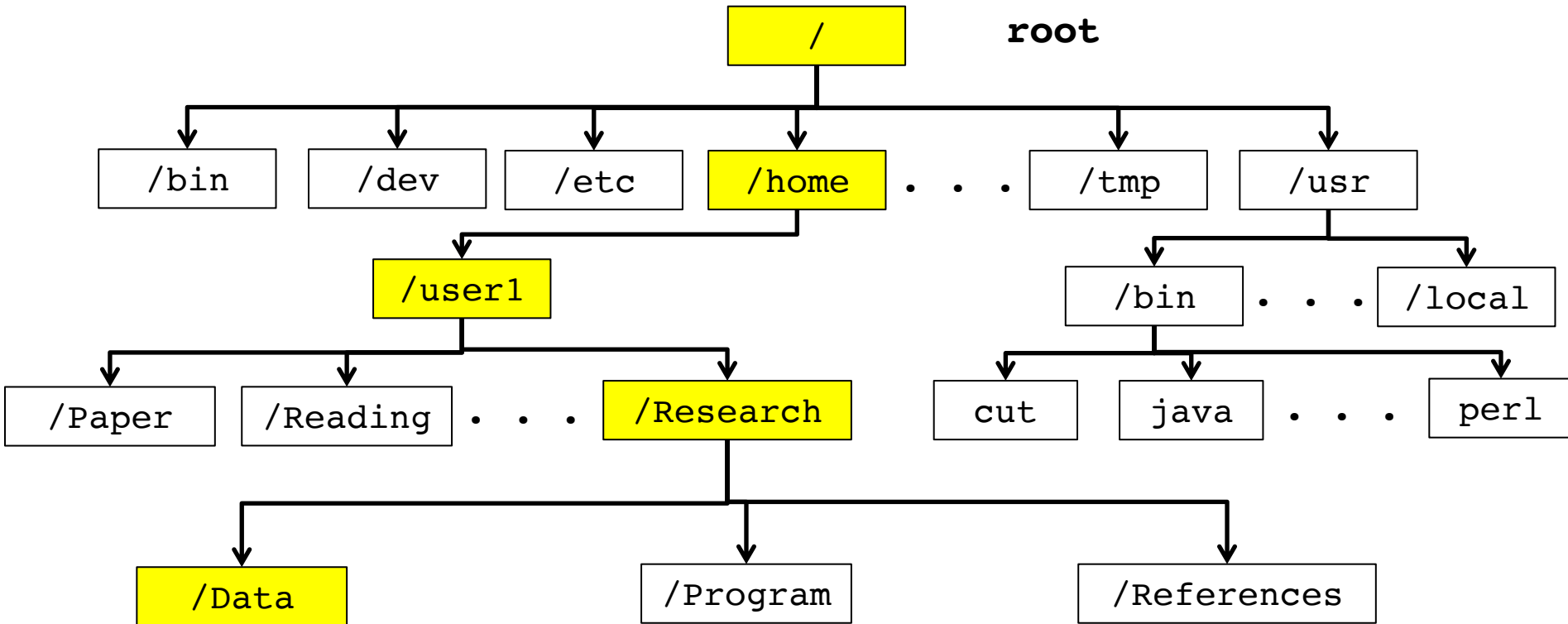
Unix Directory Structure



Unix Directory Structure

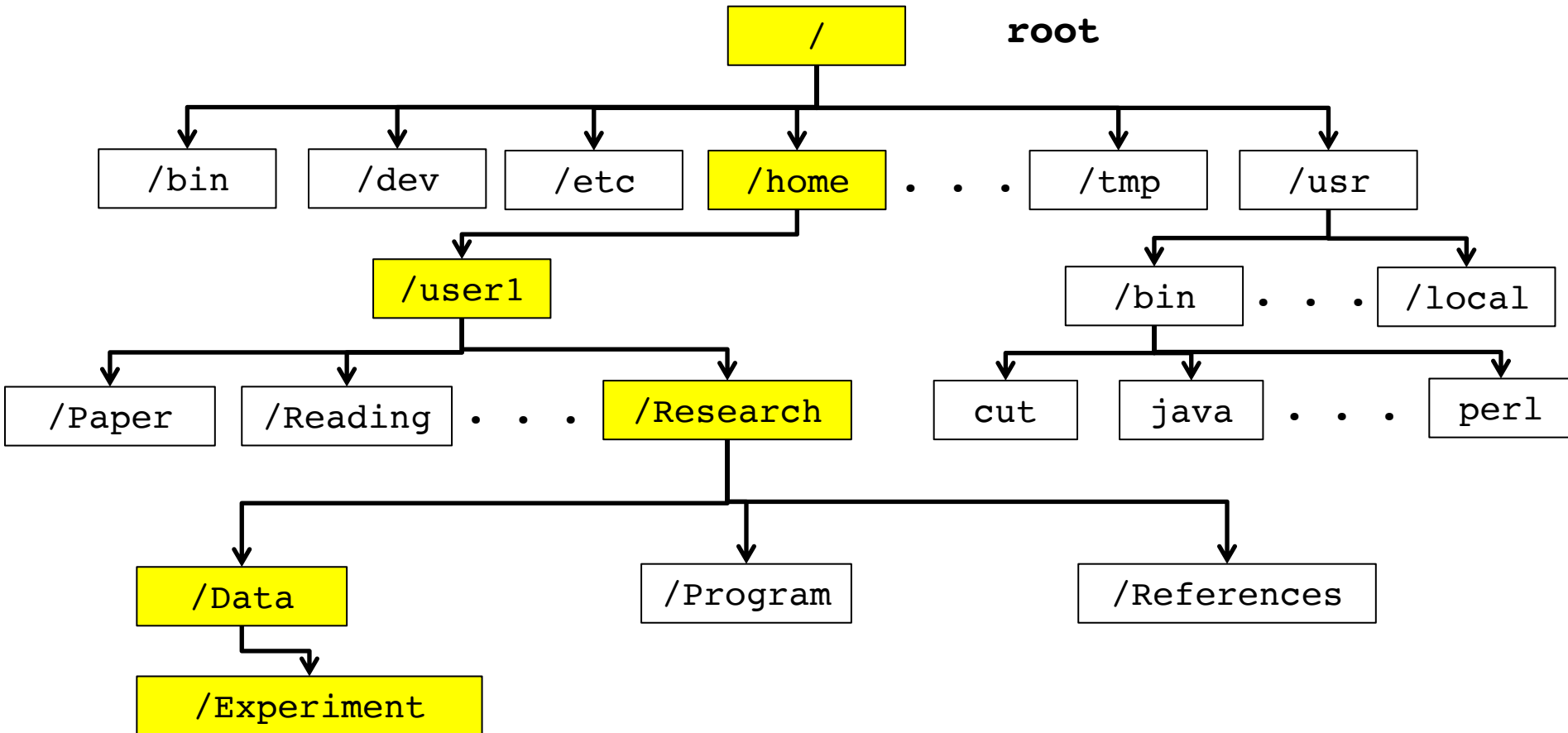


Unix Directory Structure



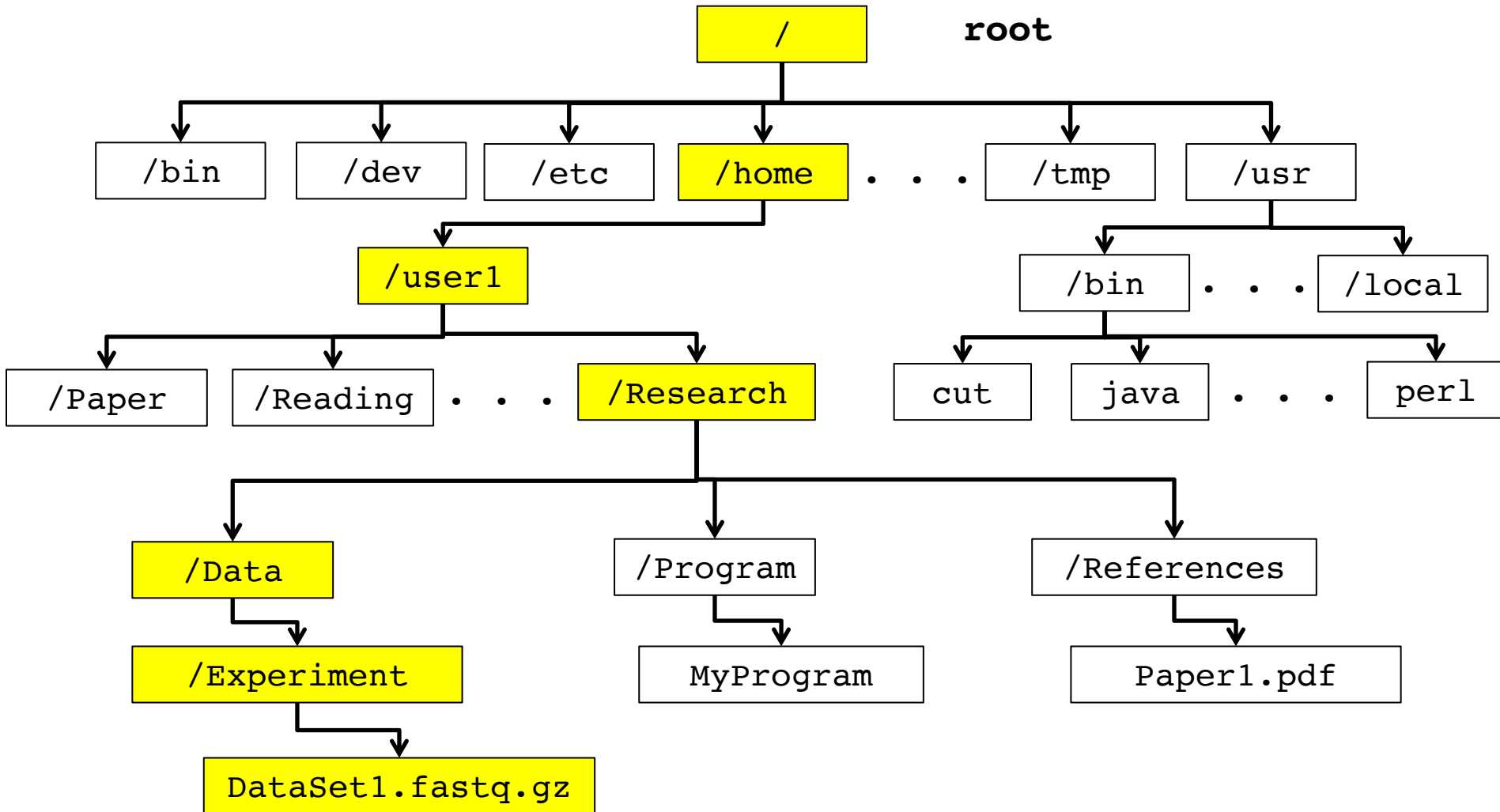
`/home/user1/Research/Data`

Unix Directory Structure



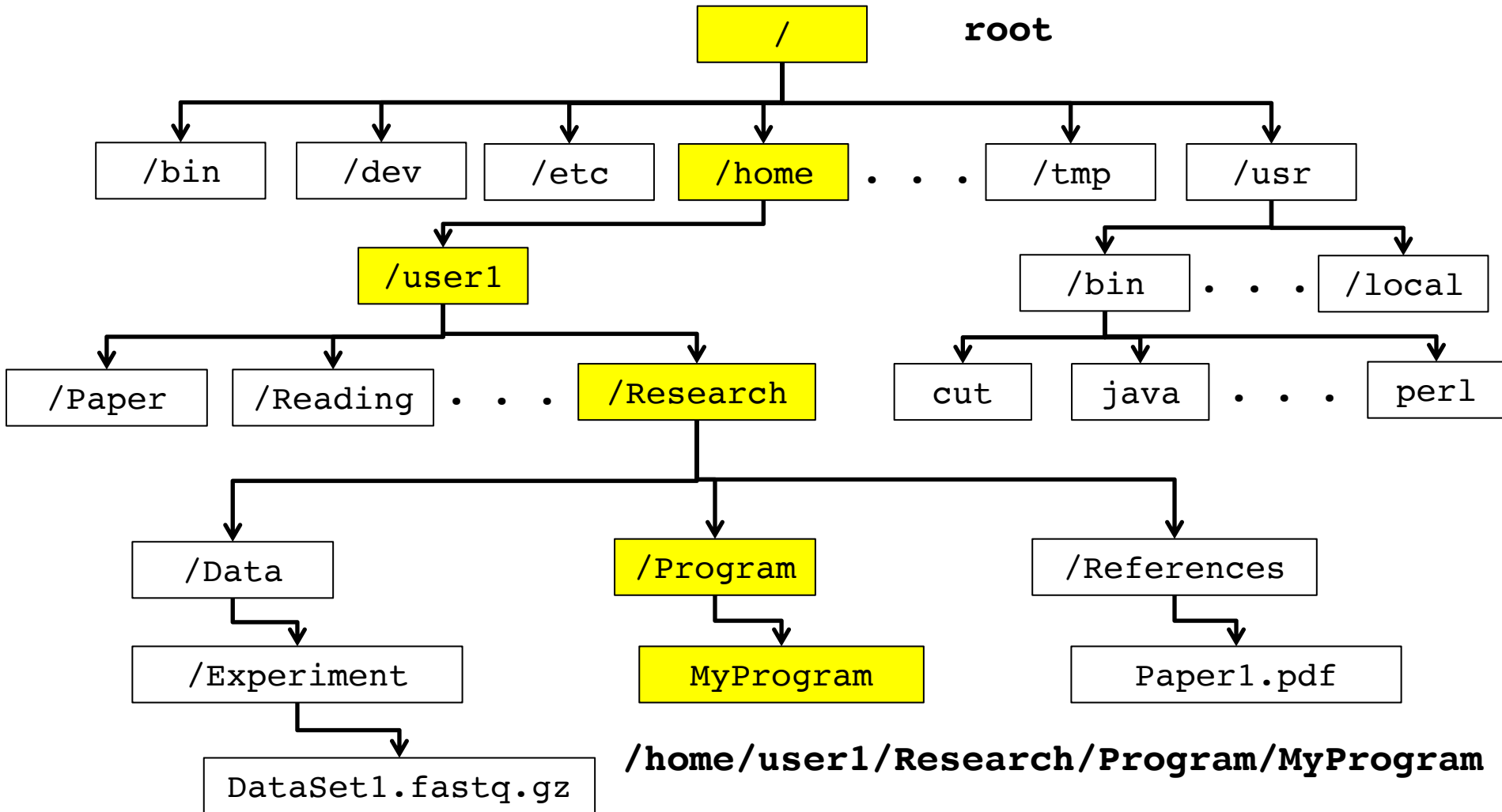
/home/user1/Research/Data/Experiment

Unix Directory Structure

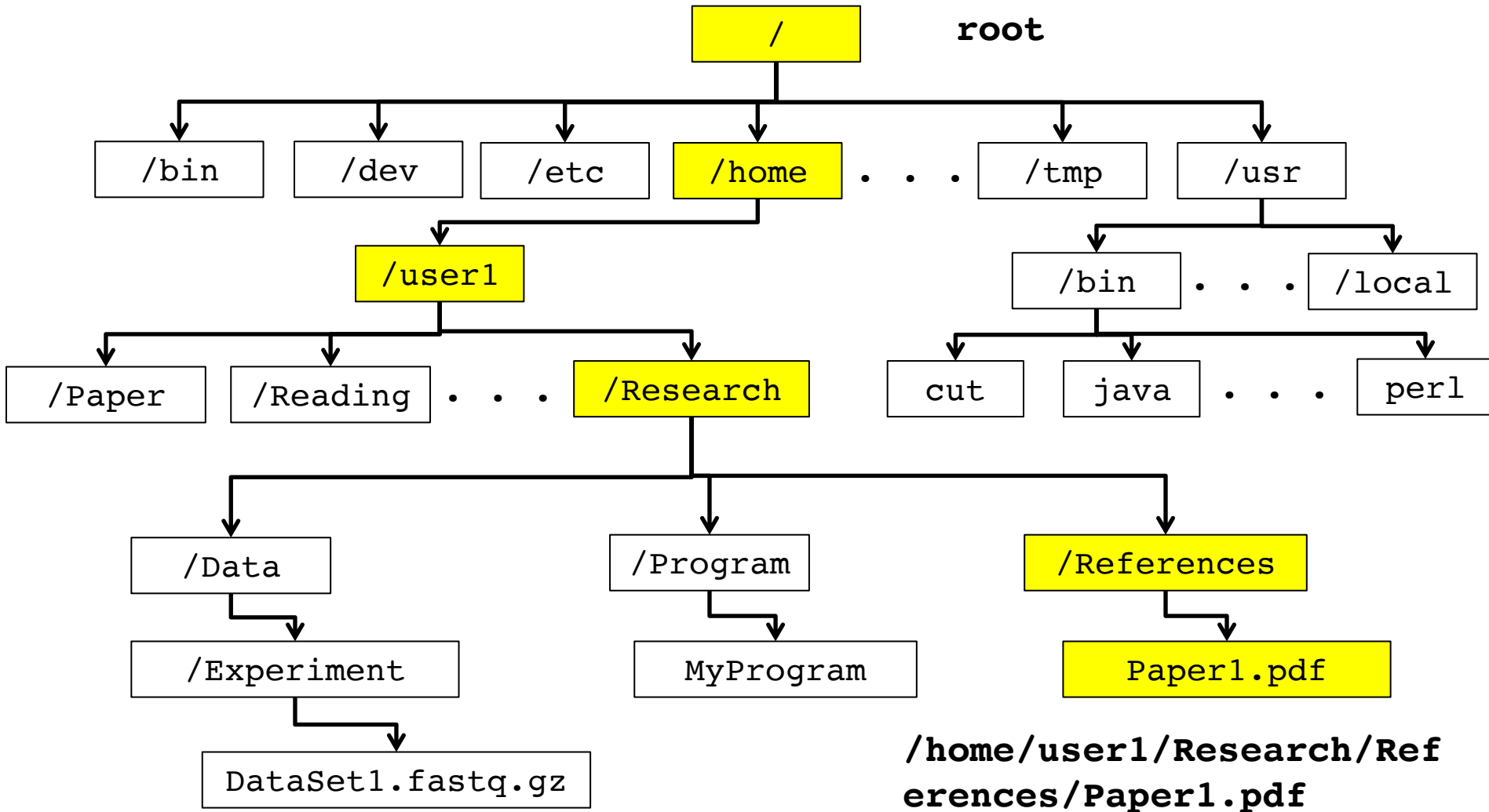


`/home/user1/Research/Data/Experiment/DataSet1.fastq.gz`

Unix Directory Structure



Unix Directory Structure



Practical

- In your terminal, type
`pwd`

What is your current working directory?

Practical

- change your directory to root
`cd /`
- List out the contents in root
`ls`
- Now, change your directory to `/bin`
- List out the contents in `/bin`

Practical

- change your directory back to your working directory
`cd`
- List out the contents in your directory
`ls`
- Now, make a new directory name **CANB7640** using `mkdir`
- Make a new directory **CLASS01** in **CANB7640** folder using `mkdir`

Practical

- change your directory back to your working directory

`cd`

- List out the contents in your directory

`ls`

- Now, make a new directory name **CANB7640** using

`mkdir`

- Make a new directory **CLASS01** in **CANB7640** folder

using `mkdir`

Practical

- Copy **HG-U133_Plus_2.na32.chip** to **CLASS01** folder

Use `cp`

- Count how many lines in **HG-U133_Plus_2.na32.chip** file

Use `wc`

- Check the **HG-U133_Plus_2.na32.chip** file

Use `more`

- how many columns in **HG-U133_Plus_2.na32.chip** ?

Practical

- Print out the top 5 lines of **HG-U133_Plus_2.na32.chip** file

Use `head`

- Print out the last 15 lines of **HG-U133_Plus_2.na32.chip** file

Use `tail`

Practical

- Extract the GeneSymbol column of **HG-U133_Plus_2.na32.chip** file and output to **Gene.txt**

Use `cut`

- Sort the file in **Gene.txt** (ascending, A - Z)

Use `sort`

- Then sort the file in **Gene.txt** (descending, Z - A)

Use `sort`

Practical

- Extract the *Gene Symbol* column of **HG-U133_Plus_2.na32.chip** file and output to **Gene.txt**

Use `cut`

- Sort the file in **Gene.txt** (ascending, A - Z)

Use `sort`

- Then sort the file in **Gene.txt** (descending, Z - A)

Use `sort`

Practical

- Count the unique *Gene Symbol* in **Gene.txt** and output as **UniqueGene.txt**

Use `sort` and `uniq`

- How many unique Gene Symbols?

Practical

- Count the unique *Gene Symbol* in **Gene.txt** and output as **UniqueGene.txt**

Use `sort` and `uniq`

- How many unique Gene Symbols?

Practical

- Print out WNT-family genes from **UniqueGene.txt** and output as **GREP_WNT.txt**

Use `grep`

- How many WNT-family genes ?

Practical

- Now write a perl script to print out WNT-family genes from **UniqueGene.txt** and save as **PERL_WNT.txt**

PrintGene.pl

```
#!/usr/bin/perl -w

while (<>)
{
    @temp = split(/\t/);
    chomp @temp;

    if ($temp[0] =~ "WNT*")
    {
        print "$temp[0]\n";
    }
}
```


Practical

- Compare the output from **GREP_WNT.txt** and **PERL_WNT.txt**

Use `comm`

- Do both files contain the same WNT-family genes?

Practical

- Now write a perl script to randomly print out 500 genes from **UniqueGene.txt** and save as **A_LIST.txt**

PrintRandomGene.pl

```
#!/usr/bin/perl -w

@mygenes = (<>);
chomp @mygenes;

$range = 21000;

for ($i = 1; $i <= 500; $i++)
{
    $random_number = rand($range);

    $int_random_number = int($random_number);

    print "$int_random_number\t$mygenes[$int_random_number]\n";
}
```

Practical

- Use the same perl script to randomly print out another 500 genes from **UniqueGene.txt** and save as **B_LIST.txt**
- Compare the two gene lists, **A_LIST.txt** and **B_LIST.txt** and print out the common genes between the two lists (overlapping genes).

Use `comm`

***HINT:** need to sort the two lists before you can use `comm` for finding the common genes between the two lists.

Practical

- Try another approach using `cat` and `uniq` commands to find out the common genes between the two lists

Use `cat` and `uniq` (with option `-d`)

***HINT:** need to sort the combined list before you can use `uniq -d` for finding the common genes between the two lists.

Practical

- What about comparing more than two gene lists?
- You need a program
- Write out the pseudocode for finding the common gene lists from two or more files
- **HINT:** need to think about the inputs and outputs, and the data structures (how to store the inputs)

Practical

- Assume these are your three gene lists (A, B, C)

A

GENE1
GENE2
GENE3
GENE4
GENE5

B

GENE1
GENE2
GENE5
GENE6
GENE7
GENE8
GENE9

C

GENE1
GENE10
GENE11
GENE2
GENE4
GENE7

Practical

Data Preprocessing

1. label the source
 - Write a program to do this?
2. combine the lists
 - Write a program to do this?
 - Use command ?
3. Sort
 - Write a program to do this?
 - Use command?

Practical

Label the Source

PrintLabel.pl

```
#!/usr/bin/perl -w

$label = "A";

while (<>)
{
    @tmp = split (/\t/);
    chomp @tmp;

    print "$tmp[0]\t$label\n";
}
```

```
perl PrintLabel.pl < Gene_A_List.txt > Gene_A_List_Label.txt
```


Practical

Label the Source

PrintLabel2.pl

```
#!/usr/bin/perl -w

$label = "A";

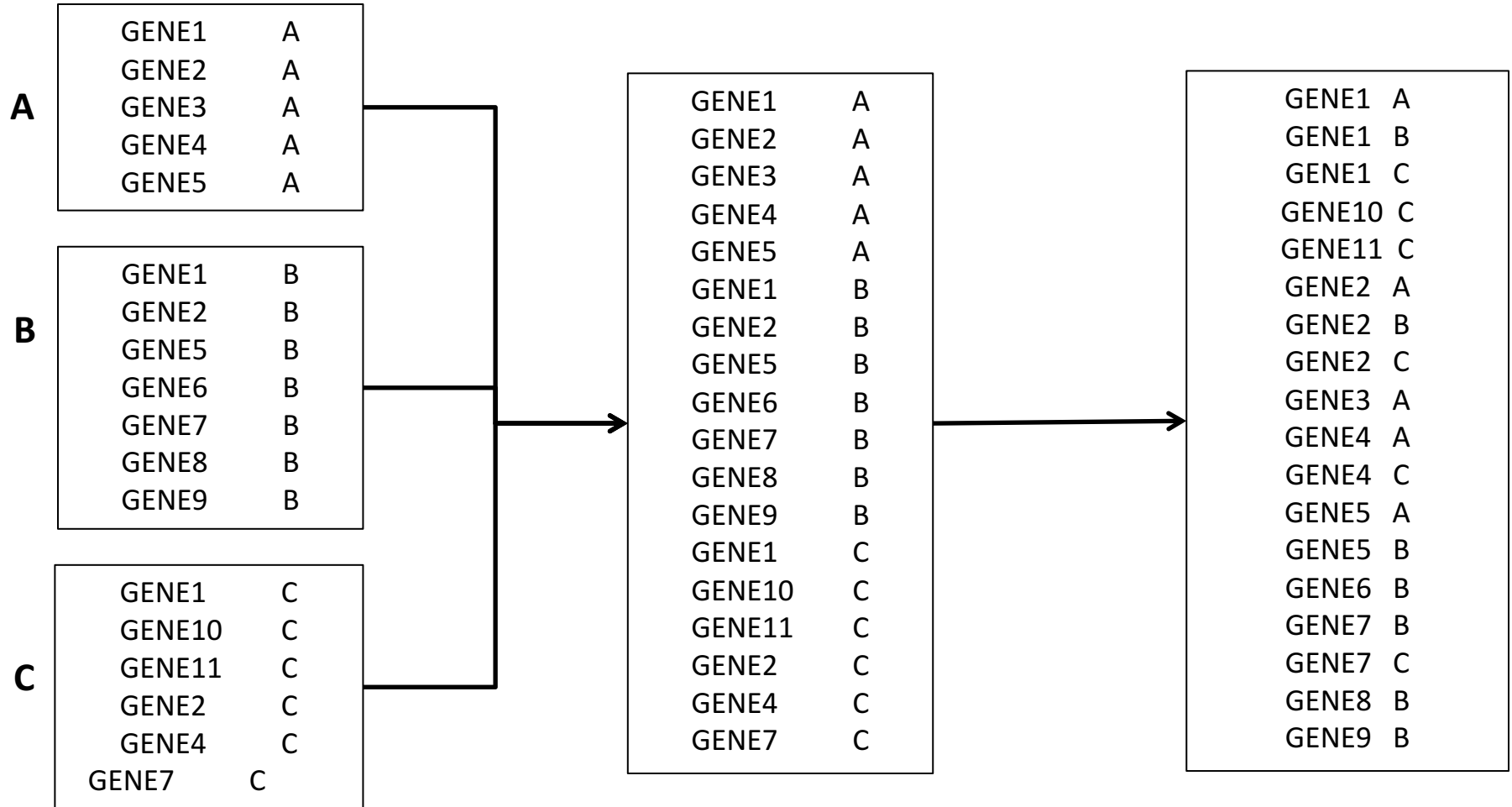
@list = (<>);

foreach $list(@list)
{
    chomp $list;
    print "$list\t$label\n";
}
```

```
perl PrintLabel2.pl < Gene_A_List.txt > Gene_A_List_Label.txt
```

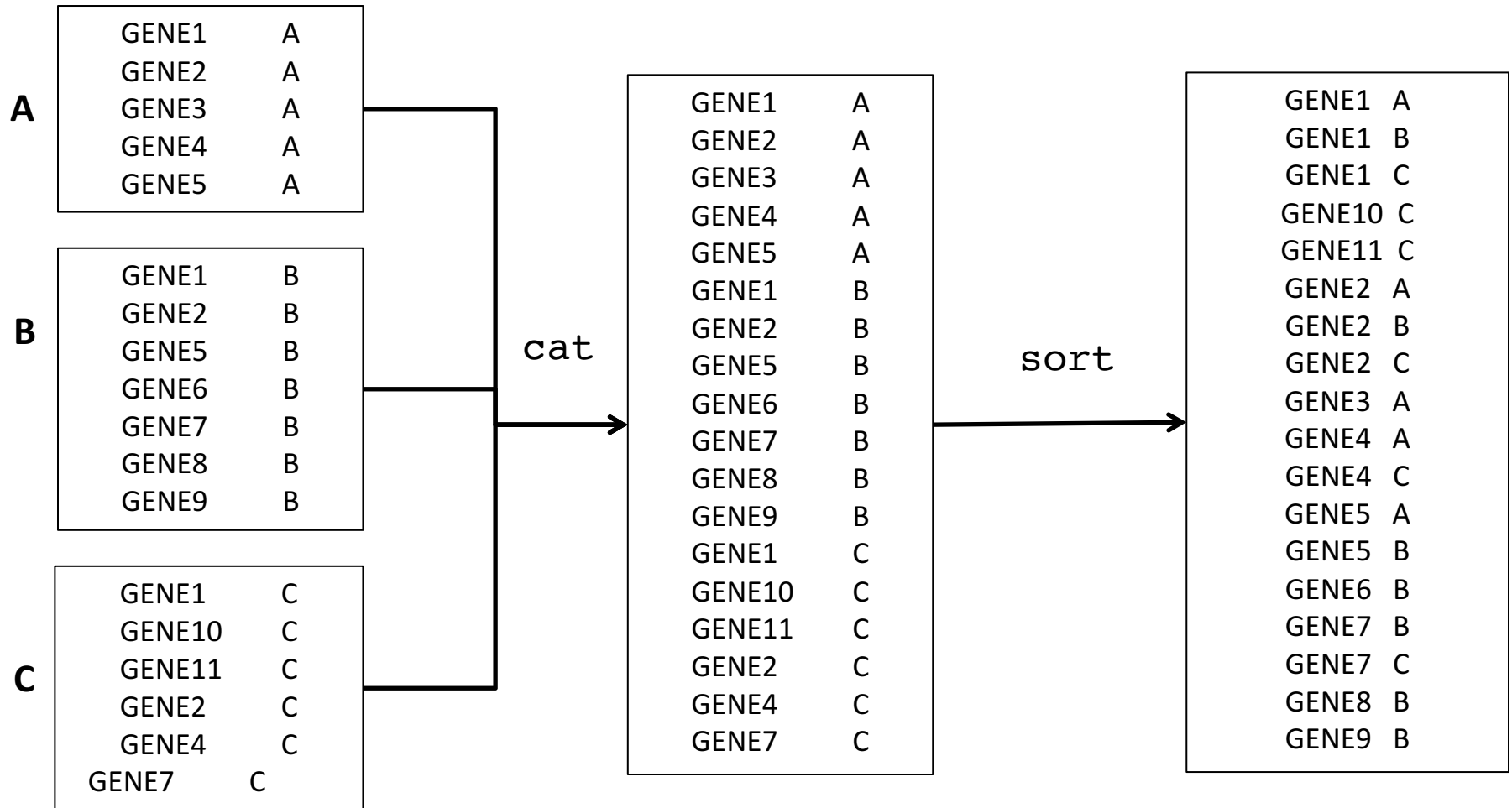
Practical

- label the source, combine the lists and sort



Practical

- labeled the source, combined the lists and sort



Practical

FindCommonGenes.pl

```
#!/usr/bin/perl -w

$count = 0;
while (<>)
{
    @data = (split /\t/);
    chomp @data;

    $gene_new = $data[0];
    $class_new = $data[1];
    if (exists $gene{$gene_new})
    {
        push (@class, $class_new);
        $count++;
    }
    else
    {
        if ($count == 0)
        {
            $gene{$gene_new} = $gene_new;
            $gene_old = $gene_new;
            push (@gene, $gene_new);
            push (@class, $class_new);
            $count++;
        }
        else
        {
            print "$count\t$gene_old\t@class\n";
            @class = ();
            $gene{$gene_new} = $gene_new;
            $gene_old = $gene_new;
            push (@gene, $gene_new);
            push (@class, $class_new);
            $count = 1;
        }
    }
}
}
```

Practical

input

```
GENE1 A
GENE1 B
GENE1 C
GENE10 C
GENE11 C
GENE2 A
GENE2 B
GENE2 C
GENE3 A
GENE4 A
GENE4 C
GENE5 A
GENE5 B
GENE6 B
GENE7 B
GENE7 C
GENE8 B
GENE9 B
```

```
perl FindCommonGenes.pl < genes_A_B_C | sort -r
```

FindCommonGenes.pl

output

```
3 GENE2 A B C
3 GENE1 A B C
2 GENE7 B C
2 GENE5 A B
2 GENE4 A C
1 GENE9 B
1 GENE8 B
1 GENE6 B
1 GENE3 A
1 GENE11 C
1 GENE10 C
```

Practical

```
GENE1 A
GENE1 B
GENE1 C
GENE10 C
GENE11 C
GENE2 A
GENE2 B
GENE2 C
GENE3 A
GENE4 A
GENE4 C
GENE5 A
GENE5 B
GENE6 B
GENE7 B
GENE7 C
GENE8 B
GENE9 B
```

```
perl FindCommonGenes.pl < genes_A_B_C | sort -r
```

FindCommonGenes.pl

```
3 GENE2 A B C
3 GENE1 A B C
2 GENE7 B C
2 GENE5 A B
2 GENE4 A C
1 GENE9 B
1 GENE8 B
1 GENE6 B
1 GENE3 A
1 GENE11 C
1 GENE10 C
```

Practical

Table (List)

3	GENE2	A	B	C
3	GENE1	A	B	C
2	GENE7	B	C	
2	GENE5	A	B	
2	GENE4	A	C	
1	GENE9	B		
1	GENE8	B		
1	GENE6	B		
1	GENE3	A		
1	GENE11	C		
1	GENE10	C		

Practical

Table (List)

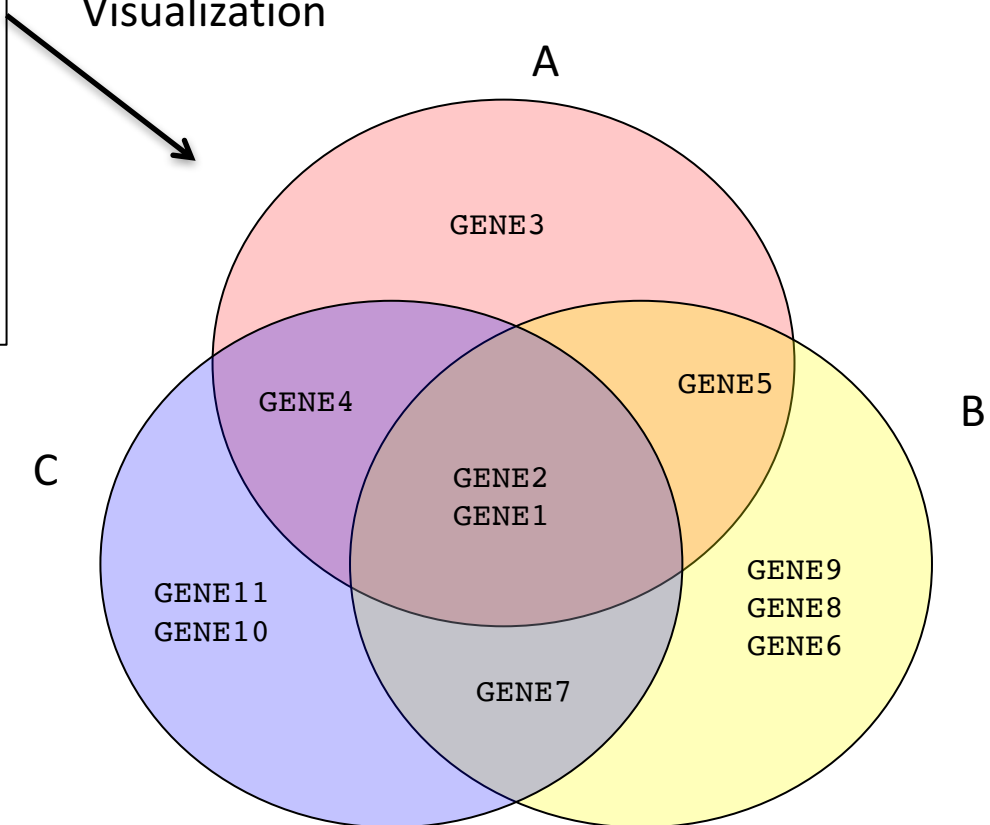
3	GENE2	A	B	C
3	GENE1	A	B	C
2	GENE7	B	C	
2	GENE5	A	B	
2	GENE4	A	C	
1	GENE9	B		
1	GENE8	B		
1	GENE6	B		
1	GENE3	A		
1	GENE11	C		
1	GENE10	C		

BORING

Practical

3	GENE2	A	B	C
3	GENE1	A	B	C
2	GENE7	B	C	
2	GENE5	A	B	
2	GENE4	A	C	
1	GENE9	B		
1	GENE8	B		
1	GENE6	B		
1	GENE3	A		
1	GENE11	C		
1	GENE10	C		

Visualization

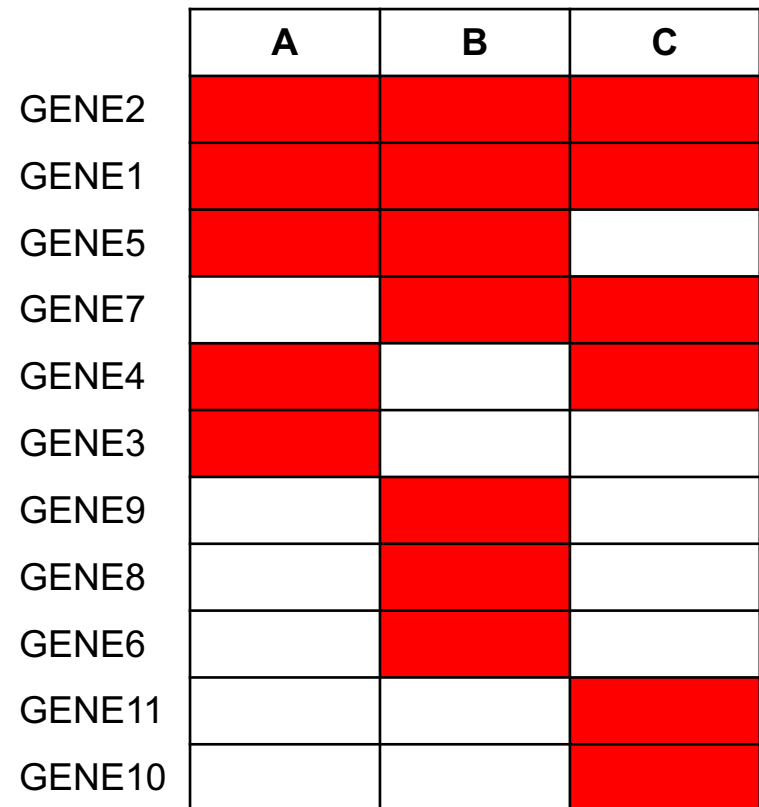


Practical

3	GENE2	A	B	C
3	GENE1	A	B	C
2	GENE7	B	C	
2	GENE5	A	B	
2	GENE4	A	C	
1	GENE9	B		
1	GENE8	B		
1	GENE6	B		
1	GENE3	A		
1	GENE11	C		
1	GENE10	C		

Visualization

Heatmap



Practical

- Try to get the probe sets covering the following genes from the **U133_Plus_2.na32.chip** and output into **MY_PROBESETS.txt**

KRAS

BRAF

MAP2K1

MAP2K2

PIK3CA

- **HINT:** use `grep`

Practical

- Try to get the probe sets covering the following genes from the **HG-U133_Plus_2.na32.chip** and output into **MY_PROBESETS.txt**

KRAS

BRAF

MAP2K1

MAP2K2

PIK3CA

- **HINT:** use `grep`

Practical

RUN_GREP_GENES

```
grep "\bKRAS\b" HG-U133_Plus_2.na32.chip
grep "\bBRAF\b" HG-U133_Plus_2.na32.chip
grep "\bMAP2K1\b" HG-U133_Plus_2.na32.chip
grep "\bMAP2K2\b" HG-U133_Plus_2.na32.chip
grep "\bPIK3CA\b" HG-U133_Plus_2.na32.chip
```

Practical

```
-rw-r--r-- 1 aikchoontan staff 211 Sep 4 10:20 RUN_GREP_GENES
```

```
chmod 777 RUN_GREP_GENES
```

```
-rwxrwxrwx 1 aikchoontan staff 211 Sep 4 10:20 RUN_GREP_GENES
```

```
./RUN_GREP_GENES
```

Practical

1559203_s_at	v-Ki-ras2 Kirsten rat sarcoma viral oncogene homolog	KRAS
1559204_x_at	v-Ki-ras2 Kirsten rat sarcoma viral oncogene homolog	KRAS
204009_s_at	v-Ki-ras2 Kirsten rat sarcoma viral oncogene homolog	KRAS
204010_s_at	v-Ki-ras2 Kirsten rat sarcoma viral oncogene homolog	KRAS
214352_s_at	v-Ki-ras2 Kirsten rat sarcoma viral oncogene homolog	KRAS
206044_s_at	v-raf murine sarcoma viral oncogene homolog B1	BRAF
236402_at	v-raf murine sarcoma viral oncogene homolog B1	BRAF
243829_at	v-raf murine sarcoma viral oncogene homolog B1	BRAF
202670_at	mitogen-activated protein kinase kinase 1	MAP2K1
202424_at	mitogen-activated protein kinase kinase 2	MAP2K2
213487_at	Mitogen-activated protein kinase kinase 2	MAP2K2
213490_s_at	mitogen-activated protein kinase kinase 2	MAP2K2
204369_at	phosphoinositide-3-kinase, catalytic, alpha polypeptide	PIK3CA
231854_at	phosphoinositide-3-kinase, catalytic, alpha polypeptide	PIK3CA
235980_at	phosphoinositide-3-kinase, catalytic, alpha polypeptide	PIK3CA

Assignment #1

- Write a **perl** script to do the same
 - Use these in the program:
 - `@array`
 - `foreach loop`
 - `if loop`