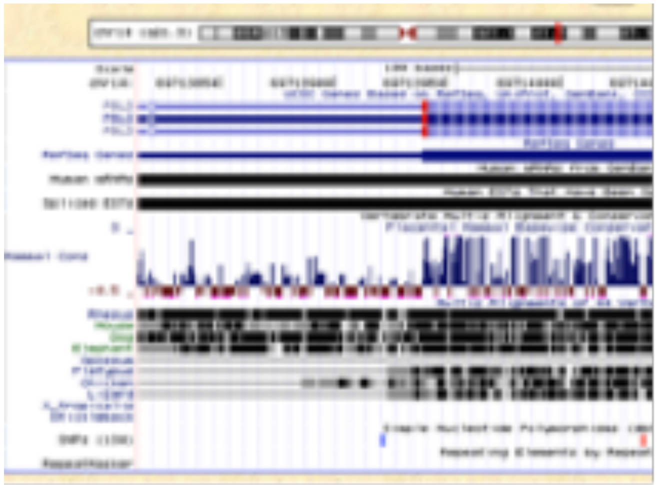
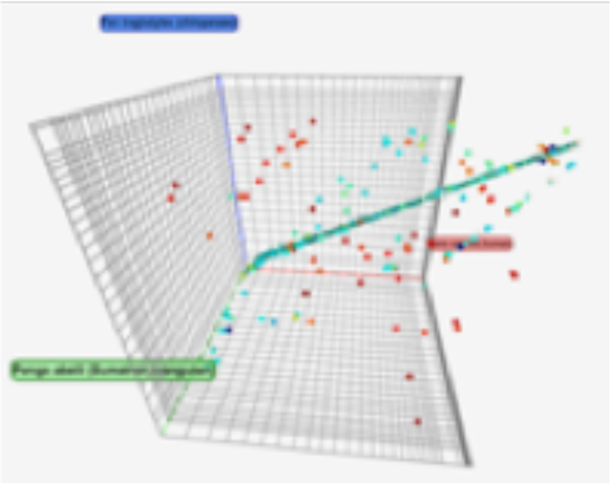
Computational Genomics

Introduction
to
Text Manipulation(s)





Introduction to Text Manipulation(s)

Add column to an existing dataset

What it does: You can enter any value and it will be added as a new column to your dataset

Example: If you original data looks like this:

```
chr1 10  100 geneA
chr2 200 300 geneB
chr2 400 500 geneC
```

Typing + in the text box will generate:

```
chr1 10  100 geneA +
chr2 200 300 geneB +
chr2 400 500 geneC +
```

You can also add line numbers by selecting Iterate: YES. In this case if you enter 1 in the text box you will get:

```
chr1 10  100 geneA 1
chr2 200 300 geneB 2
chr2 400 500 geneC 3
```

Introduction to Text Manipulation(s)

Cut columns from a table (cut)

What it does: This tool selects (cuts out) specified columns from the dataset.

Columns are specified as c1, c2, and so on.
Column count begins with 1
Columns can be specified in any order (e.g., c2,c1,c6)
If you specify more columns than actually present – empty spaces will be filled with dots

Input Example: Input dataset (six columns: c1, c2, c3, c4, c5, and c6):

```
chr1 10   1000  gene1 0 +
chr2 100  1500  gene2 0 +
```

cut on columns "c1,c4,c6" will return:

```
chr1 gene1 +
chr2 gene2 +
```

cut on columns "c6,c5,c4,c1" will return:

```
+ 0 gene1 chr1
+ 0 gene2 chr2
```

cut on columns "c1–c3" will return:

```
chr1 10   1000
chr2 100  1500
```

cut on columns "c8,c7,c4" will return:

```
. . gene1
. . gene2
```

Introduction to Text Manipulation(s)

Merge Columns together

What it does: This tool merges columns together. Any number of valid columns can be merged in any order

Example: Input dataset (five columns: c1, c2, c3, c4, and c5):

```
1 10   1000  gene1 chr
2 100  1500  gene2 chr
```

merging columns "c5,c1" will return:

```
1 10   1000  gene1 chr chr1
2 100  1500  gene2 chr chr2
```

Note that all original columns are preserved and the result of merge is added as the rightmost column

Introduction to Text Manipulation(s)

Change Case of selected columns

What it does: This tool selects specified columns from a dataset and converts the values of those columns to upper or lower case

Columns are specified as c1, c2, and so on.
Columns can be specified in any order (e.g., c2,c1,c6)

Example: Changing columns 1 and 3 (delimited by Comma) to upper case in:

```
apple,is,good
windows,is,bad
```

will result in:

```
APPLE is GOOD
WINDOWS is BAD
```

Introduction to Text Manipulation(s)

Unfold columns from a table

What it does: This tool will unfold one column of your input dataset.

Input Example:

```
a   b       1,2,3,4,5       c
```

Output Example:

```
    a   b       1       c
    a   b       2       c
    a   b       3       c
    a   b       4       c
    a   b       5       c
```

Introduction to Text Manipulation(s)

Concatenate datasets tail-to-head (cat)

What it does: Concatenates datasets

Example

Concatenating Dataset:

```
chrX  151087187  151087355  A  0  -
chrX  151572400  151572481  B  0  +
```

with Dataset1:

```
chr1  151242630  151242955  X  0  +
chr1  151271715  151271999  Y  0  +
chr1  151278832  151279227  Z  0  -
```

and with Dataset2:

```
chr2  100000030  200000955  P  0  +
chr2  100000015  200000999  Q  0  +
```

will result in the following:

```
chrX  151087187  151087355  A  0  -
chrX  151572400  151572481  B  0  +
chr1  151242630  151242955  X  0  +
chr1  151271715  151271999  Y  0  +
chr1  151278832  151279227  Z  0  -
chr2  100000030  200000955  P  0  +
chr2  100000015  200000999  Q  0  +
```

Introduction to Text Manipulation(s)

tac reverse a file (reverse cat)

```
What it does: tac is a Linux command that allows you to see a file line-by-line backwards
              It is named by analogy with cat

Mandatory arguments to long options are mandatory for short options too:

-b, -before                attach the separator before instead of after
-r, -regex                 interpret the separator as a regular expression
-s, -separator=STRING      use STRING as the separator instead of newline
```

Example:

Input file	default settings	with option -s 5:	with option -b and -s 5:
0	9		5
1	8	#	#
2	7	6	6
3	6	7	7
4	#	8	8
5	5	9	9
#	4	0	0
6	3	1	1
7	2	2	2
8	1	3	3
9	0	4	4

Introduction to Text Manipulation(s)

Join two files

What it does: This tool joins two tabular files based on a common key column

Example:

First File	
Fruit	Color
Apple	red
Banana	yellow
Orange	orange
Melon	green

Second File	
Fruit	Price
Orange	7
Avocado	8
Apple	4
Banana	3

Joining both files, using key column 1 and a header line, will return:

Joined File		
Fruit	Color	Price
Apple	red	4
Avocado	.	8
Banana	yellow	3
Melon	green	.
Orange	orange	7

Introduction to Text Manipulation(s)

Multi-Join (combine multiple files)

What it does: This tool joins multiple tabular files based on a common key column.

Example:

To join three files, based on the 4th column, and keeping the 7th,8th,9th columns:

First file (AAA):

```
chr4    888449   890171   FBtr0308778   0   +   266    1527    1722
chr4    972167   979017   FBtr0310651   0   -   3944   6428    6850
chr4    972186   979017   FBtr0089229   0   -   3944   6428    6831
chr4    972186   979017   FBtr0089231   0   -   3944   6428    6831
chr4    972186   979017   FBtr0089233   0   -   3944   6428    6831
chr4    995793   996435   FBtr0111046   0   +   7      166     642
chr4    995793   997931   FBtr0111044   0   +   28     683     2138
chr4    995793   997931   FBtr0111045   0   +   28     683     2138
chr4    1054029  1047719  FBtr0089223   0   -   5293   13394   13690
...
```

Second File (BBB):

```
chr4    90286    134453   FBtr0309803   0   +   657   29084   44167
chr4    251355   266499   FBtr0089116   0   +   56    1296    15144
chr4    252050   266506   FBtr0308086   0   +   56    1296    14456
chr4    252050   266506   FBtr0308087   0   +   56    1296    14456
chr4    252053   266528   FBtr0300796   0   +   56    1296    14475
chr4    252053   266528   FBtr0300800   0   +   56    1296    14475
chr4    252055   266528   FBtr0300798   0   +   56    1296    14473
chr4    252055   266528   FBtr0300799   0   +   56    1296    14473
chr4    252541   266528   FBtr0300797   0   +   56    1296    13987
...
```

Third file (CCC):

```
chr4    972167   979017   FBtr0310651   0   -   9927    6738    6850
chr4    972186   979017   FBtr0089229   0   -   9927    6738    6831
chr4    972186   979017   FBtr0089231   0   -   9927    6738    6831
chr4    972186   979017   FBtr0089233   0   -   9927    6738    6831
chr4    995793   996435   FBtr0111046   0   +   5       304     642
chr4    995793   997931   FBtr0111044   0   +   17      714     2138
chr4    995793   997931   FBtr0111045   0   +   17      714     2138
chr4    1054029  1047719  FBtr0089223   0   -   17646   13536   13690
...
```

Joining the files, using key column 4, value columns 7,8,9 and a header line, will return:

Input files need not be sorted.

Third file (CCC):

```
key            AAA__V7  AAA__V8  AAA__V9  BBB__V7  BBB__V8  BBB__V9  CCC__V7  CCC__V8  CCC__V9
FBtr0089116          0        0        0       56     1296    15144        0        0        0
FBtr0089223       5293    13394    13690        0        0        0    17646    13536    13690
FBtr0089229       3944     6428     6831        0        0        0     9927     6738     6831
FBtr0089231       3944     6428     6831        0        0        0     9927     6738     6831
FBtr0089233       3944     6428     6831        0        0        0     9927     6738     6831
FBtr0111044         28      683     2138        0        0        0       17      714     2138
FBtr0111045         28      683     2138        0        0        0       17      714     2138
FBtr0111046          7      166      642        0        0        0        5      304      642
FBtr0300796          0        0        0       56     1296    14475        0        0        0
...
```

Introduction to Text Manipulation(s)

Paste two files side by side

What it does: This tool merges two datasets side by side

If the first (left) dataset contains column assignments such as chromosome, start, end and strand, these will be preserved

However, if you would like to change column assignments, click the pencil icon in the history item

Example:

First dataset:

```
a 1
a 2
a 3
```

Second dataset:

```
20
30
40
```

Pasting them together will produce:

Final dataset:

```
a 1 20
a 2 30
a 3 40
```

Introduction to Text Manipulation(s)

Select first lines from a dataset (head)

What it does: This tool outputs specified number of lines from the beginning of a dataset

Example: Selecting 2 lines from this:

```
chr7  56632  56652  D17003_CTCF_R6  310  +
chr7  56736  56756  D17003_CTCF_R7  354  +
chr7  56761  56781  D17003_CTCF_R4  220  +
chr7  56772  56792  D17003_CTCF_R7  372  +
chr7  56775  56795  D17003_CTCF_R4  207  +
```

will produce:

```
chr7  56632  56652  D17003_CTCF_R6  310  +
chr7  56736  56756  D17003_CTCF_R7  354  +
```

Introduction to Text Manipulation(s)

Select last lines from a dataset (tail)

What it does: This tool outputs specified number of lines from the end of a dataset

Example: Selecting 2 lines from this:

```
chr7    57134   57154   D17003_CTCF_R7  356     -
chr7    57247   57267   D17003_CTCF_R4  207     +
chr7    57314   57334   D17003_CTCF_R5  269     +
chr7    57341   57361   D17003_CTCF_R7  375     +
chr7    57457   57477   D17003_CTCF_R3  188     +
```

will produce:

```
chr7    57341   57361   D17003_CTCF_R7  375     +
chr7    57457   57477   D17003_CTCF_R3  188     +
```

Introduction to Text Manipulation(s)

Remove beginning of a file

What it does: This tool removes a specified number of lines from the beginning of a dataset

Example:

Input File:

```
chr7  56632  56652   D17003_CTCF_R6  310  +
chr7  56736  56756   D17003_CTCF_R7  354  +
chr7  56761  56781   D17003_CTCF_R4  220  +
chr7  56772  56792   D17003_CTCF_R7  372  +
chr7  56775  56795   D17003_CTCF_R4  207  +
```

After removing the first 3 lines the dataset will look like this:

```
chr7  56772  56792   D17003_CTCF_R7  372  +
chr7  56775  56795   D17003_CTCF_R4  207  +
```

Introduction to Text Manipulation(s)

Sort data in ascending or descending order

This tool sorts an input file.

Sorting Styles:

- Fast Numeric: sort by numeric values. Handles integer values (e.g. 43, 134) and decimal-point values (e.g. 3.14). Does not handle scientific notation (e.g. -2.32e2)

- General Numeric: sort by numeric values. Handles all numeric notations (including scientific notation). Slower than fast numeric, so use only when necessary.

- Natural Sort: Sort in 'natural' order (natural to humans, not to computers).

- Alphabetical sort: Sort in strict alphabetical order.

- Human-readable numbers: Sort human readable numbers (e.g. 1G > 2M > 3K > 400)

- Random order: return lines in random order

Introduction to Text Manipulation(s)

Sort data in ascending or descending order

Example – Header line

Input file (note first line is a header line, should not be sorted):

```
Fruit.  Color   Price
Banana  Yellow  4.1
Avocado Green   8.0
Apple   Red     3.0
Melon   Green   6.1
```

Sorting by numeric order on column 3, with header, will return:

```
Fruit   Color   Price
Apple   Red     3.0
Banana  Yellow  4.1
Melon   Green   6.1
Avocado Green   8.0
```

Introduction to Text Manipulation(s)

Sort data in ascending or descending order

Example – Natural vs. Alphabetical sorting

Given the following list:

```
chr4
chr13
chr1
chr10
chr20
chr2
```

Alphabetical sort would produce the following sorted list:

```
chr1
chr10
chr13
chr2
chr20
chr4
```

Natural Sort would produce the following sorted list:

```
chr1
chr2
chr4
chr10
chr13
chr20
```

Introduction to Text Manipulation(s)

Select random lines from a file

What it does

This tool selects N random lines from a file, with no repeats, and preserving ordering

Example

Input File:

```
chr7  56632  56652   D17003_CTCF_R6  310  +
chr7  56736  56756   D17003_CTCF_R7  354  +
chr7  56761  56781   D17003_CTCF_R4  220  +
chr7  56772  56792   D17003_CTCF_R7  372  +
chr7  56775  56795   D17003_CTCF_R4  207  +
```

Selecting 2 random lines might return this:

```
chr7  56736  56756   D17003_CTCF_R7  354  +
chr7  56772  56792   D17003_CTCF_R7  372  +
```

Introduction to Text Manipulation(s)

Unique occurrences of each record

What it does:

This tool returns all unique lines using the 'sort -u' command.

It can be used with unsorted files

The input file needs to be tab separated. Please convert your file if necessary

Input File:

```
chr1   10  100  gene1
chr1  105  200  gene2
chr1   10  100  gene1
chr2   10  100  gene4
chr2 1000 1900  gene5
chr3   15 1656  gene6
chr2   10  100  gene4
```

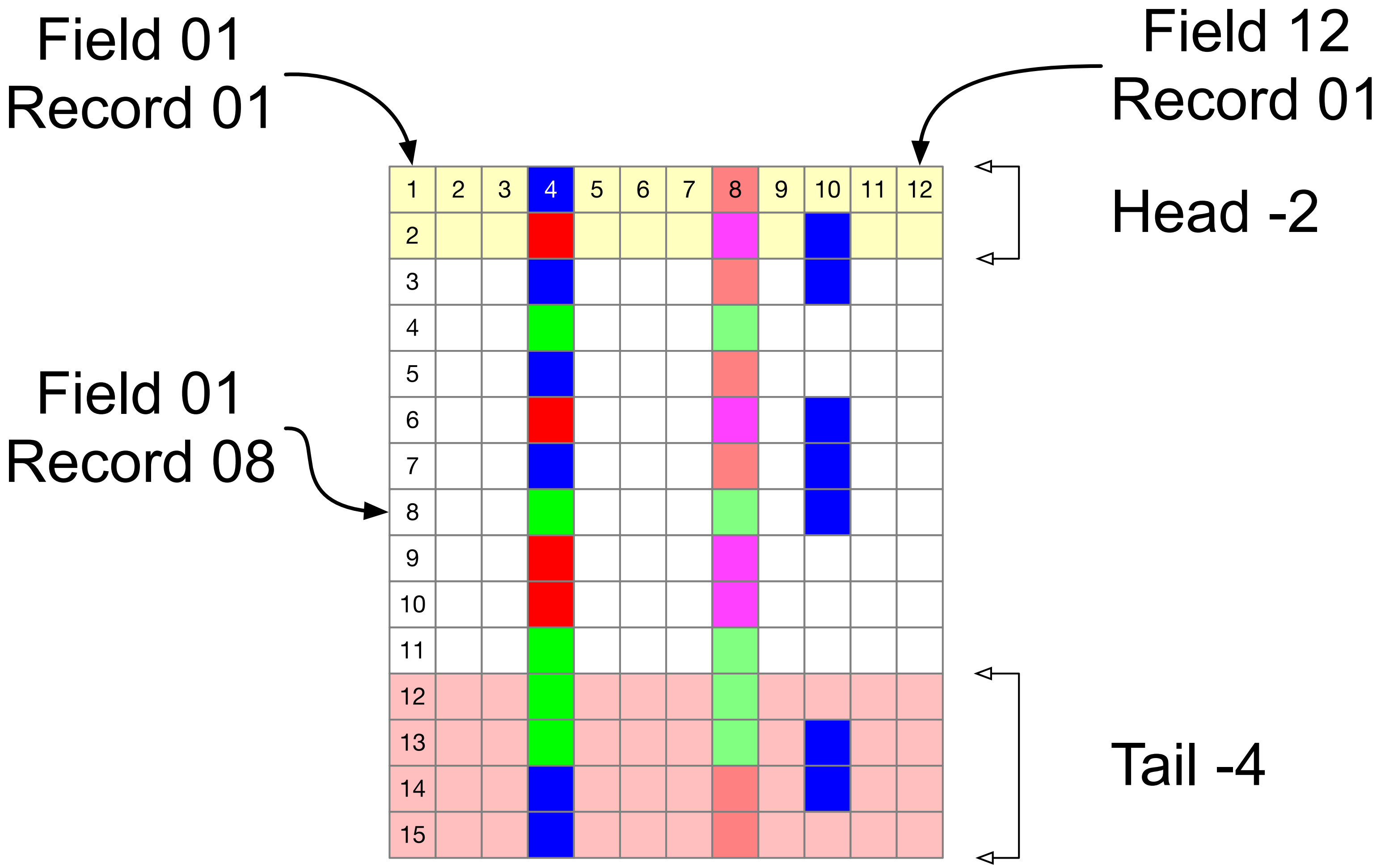
Unique lines will result in:

```
chr1   10  100  gene1
chr1  105  200  gene2
chr2   10  100  gene4
chr2 1000 1900  gene5
chr3   15 1656  gene6
```

Introduction to Text Manipulation(s)

Defining a Table

Table containing
12 fields (columns) and 15 records (rows or lines)



Field 01
Record 01

Field 12
Record 01

Field 01
Record 08

Head -2

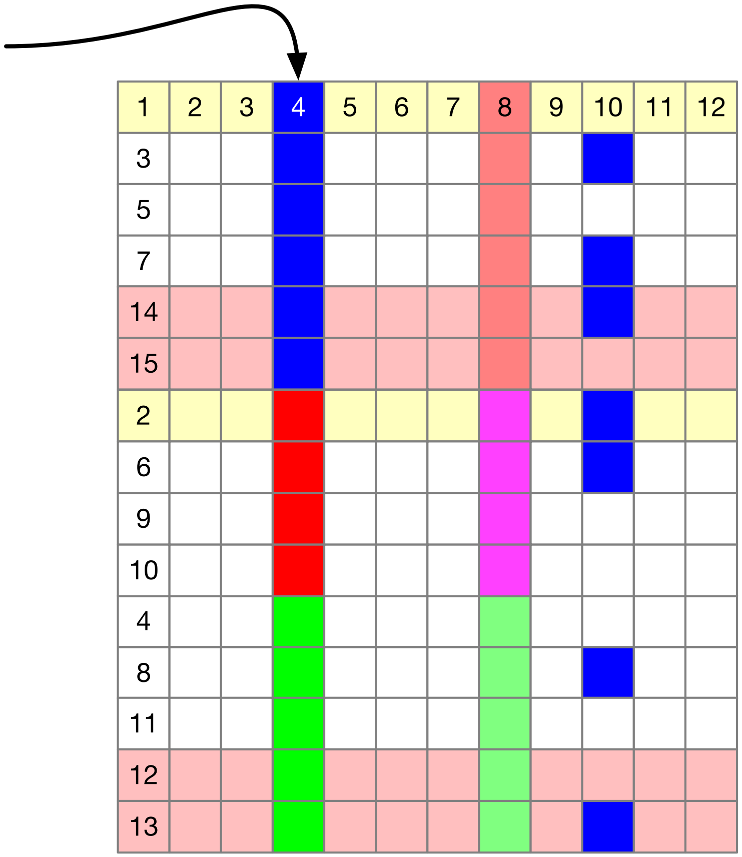
Tail -4

Introduction to Text Manipulation(s)

Defining a Table

Table containing
12 fields (columns) and 15 records (rows or lines)

Sorting by Colors
On Field 04



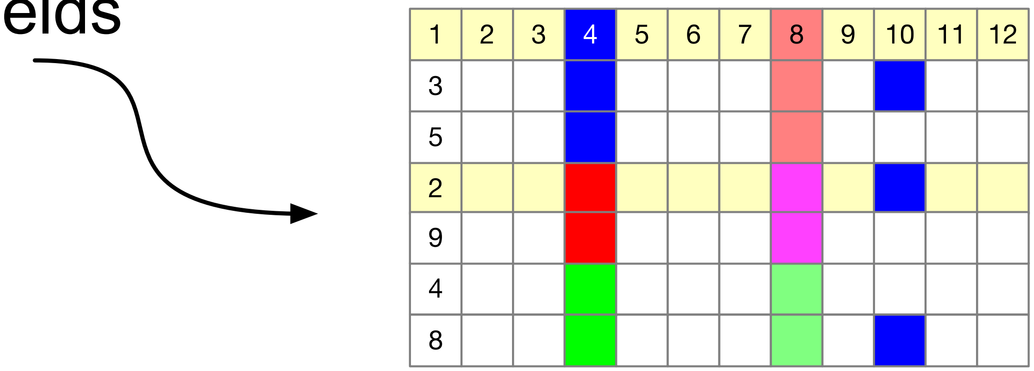
1	2	3	4	5	6	7	8	9	10	11	12
3			🟦				🟥		🟦		
5			🟦				🟥				
7			🟦				🟥		🟦		
14			🟦				🟥		🟦		
15			🟦				🟥				
2			🟥				🟪		🟦		
6			🟥				🟪		🟦		
9			🟥				🟪				
10			🟥				🟪				
4			🟩				🟩				
8			🟩				🟩		🟦		
11			🟩				🟩				
12			🟩				🟩				
13			🟩				🟩		🟦		

Introduction to Text Manipulation(s)

Defining a Table

Table containing
12 fields (columns) and 15 records (rows or lines)

Removing Duplicate
Fields



1	2	3	4	5	6	7	8	9	10	11	12
3			🟦				🟧		🟦		
5			🟦				🟧				
2			🟥				🟪		🟦		
9			🟥				🟪				
4			🟩				🟩				
8			🟩				🟩		🟦		