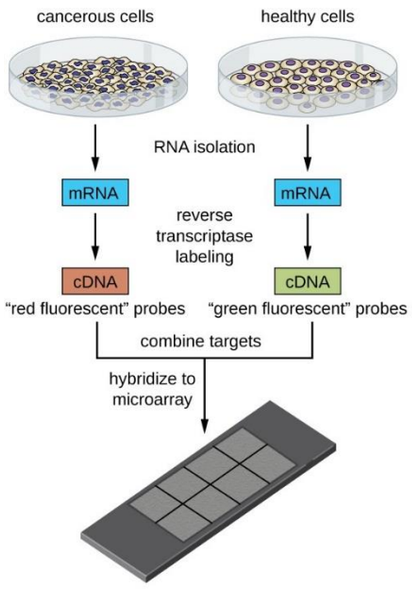
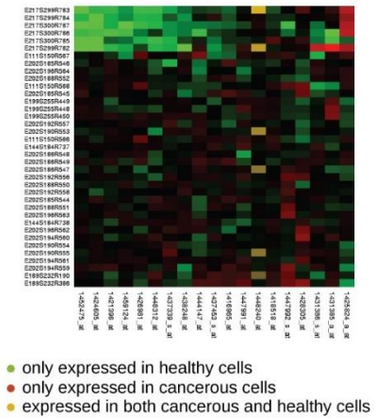
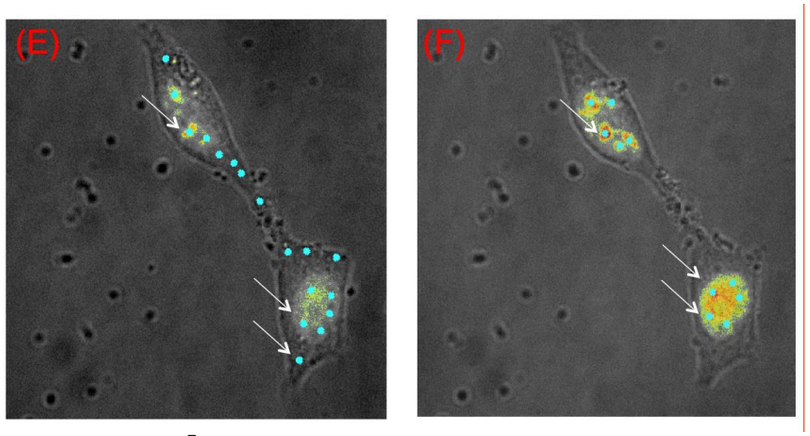
Introduction to Scientific Computation 113E

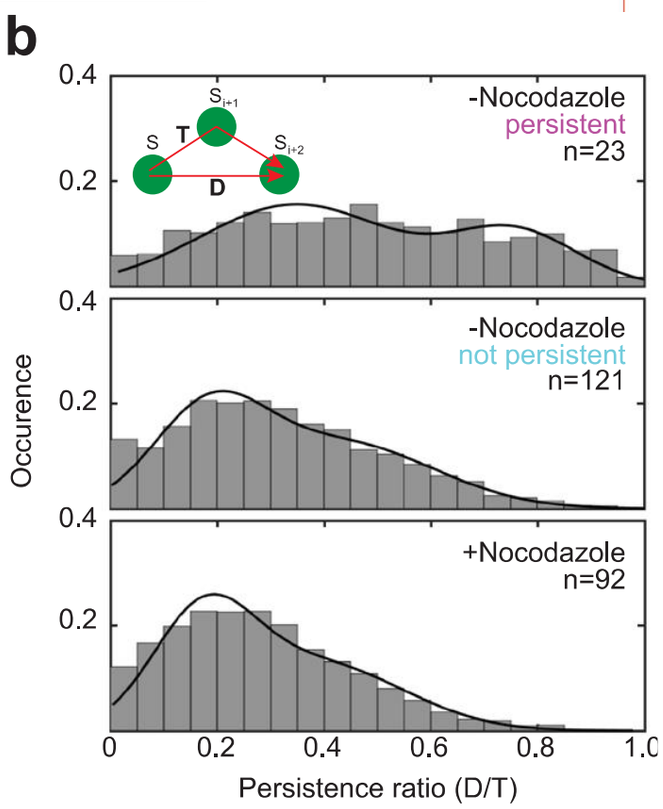


(a)



(b)





```
fretmoallimgsS = cell(1,1);


for findx1 = 1:1;
    fnamefr1 = flistfrmovie1{findx1};
datA  = zeros(ysize, xsize, nframes);
for k = 1:nframes;
    if mod(k, 10) == 0
        k;
    end
    datA(:,:,k) = imread(fnamefr1, 't
```

Variables - data1

corr_coef_y | allmeansdata | p_value_o | data1

27130x34 table

	1 Gene	2 ARNA	3 TRNA	4 ARNA1	5 TRNA1	6 ARNA2	7 TRNA2	8 ARNA3	9 TRNA3	10 TRNA4	11 TRNA5
1	"LOC1024…	0	1.0045	4.0185	0.9162	2.9799	3.3377	1.3212	2.1511	1.0805	
2	"ZBTB42"	27.8394	37.1676	55.2547	30.2348	42.7112	32.2643	54.1705	48.3991	35.6578	50.26
3	"FCAMR"	1.1136	0	1.0046	0	0.9933	0	0	0	0	
4	"ZNF503-…	41.2024	35.1586	40.1853	16.4917	35.7582	32.2643	60.7767	23.6618	47.5438	40.84
5	"NFU1"	111.3578	123.5573	118.5465	133.7663	91.3821	86.7800	76.6315	120.4600	100.4903	68.07
6	"ELSPBP1"	0	0	0	0	0	0	0	0	0	
7	"ZRANB3"	190.4218	183.8291	141.6531	169.4984	155.9455	92.3428	118.9109	121.5355	152.3563	102.63
8	"MECR"	259.4637	291.3139	188.8708	237.2977	289.0455	189.1358	257.6403	253.8264	347.9341	271.24
9	"LOC1057…	0	0	0	0	0	0	0	0	0	
10	"LINC003…	2.2272	2.0091	3.0139	4.5810	6.9530	0	0	0	1.0805	4.18
11	"AARSD1"	1.1136	0	0	0	0	2.2251	0	0	0	
12	"DEXI"	485.5200	435.9663	492.2695	308.7619	511.5410	493.9782	478.2860	357.0779	467.8742	569.72
13	"DCHS1"	1.1559e+03	1.0035e+03	1.2116e+03	1.3138e+03	1.3479e+03	1.4652e+03	1.8788e+03	1.4842e+03	1.7224e+03	1.2724e+
14	"PSMD2"	1.2550e+03	1.8232e+03	1.3914e+03	1.4861e+03	1.5545e+03	1.3206e+03	1.5630e+03	1.1282e+03	1.4382e+03	1.5730e+
15	"GABRR1"	3.3407	4.0181	2.0093	1.8324	5.9597	8.9005	6.6062	2.1511	2.1611	4.18
16	"PKNOX2"	780.6181	676.0491	640.9550	244.6274	522.4672	406.0857	486.2134	287.1680	504.6126	799.07
17	"TIPARP"	309.5747	294.3275	372.7184	721.0553	238.3881	354.9078	395.0484	379.6641	298.2293	273.34
18	"ADAM20"	113.5849	91.4123	74.3427	89.7883	100.3216	160.2091	104.3773	73.1364	63.7519	97.39
19	"LOC2847…	0	0	0	0	0	0	0	0	0	
20	"MIR4715"	0	0	0	0	0	1.1126	0	0	0	

Assc. Prof. Halil Bayraktar

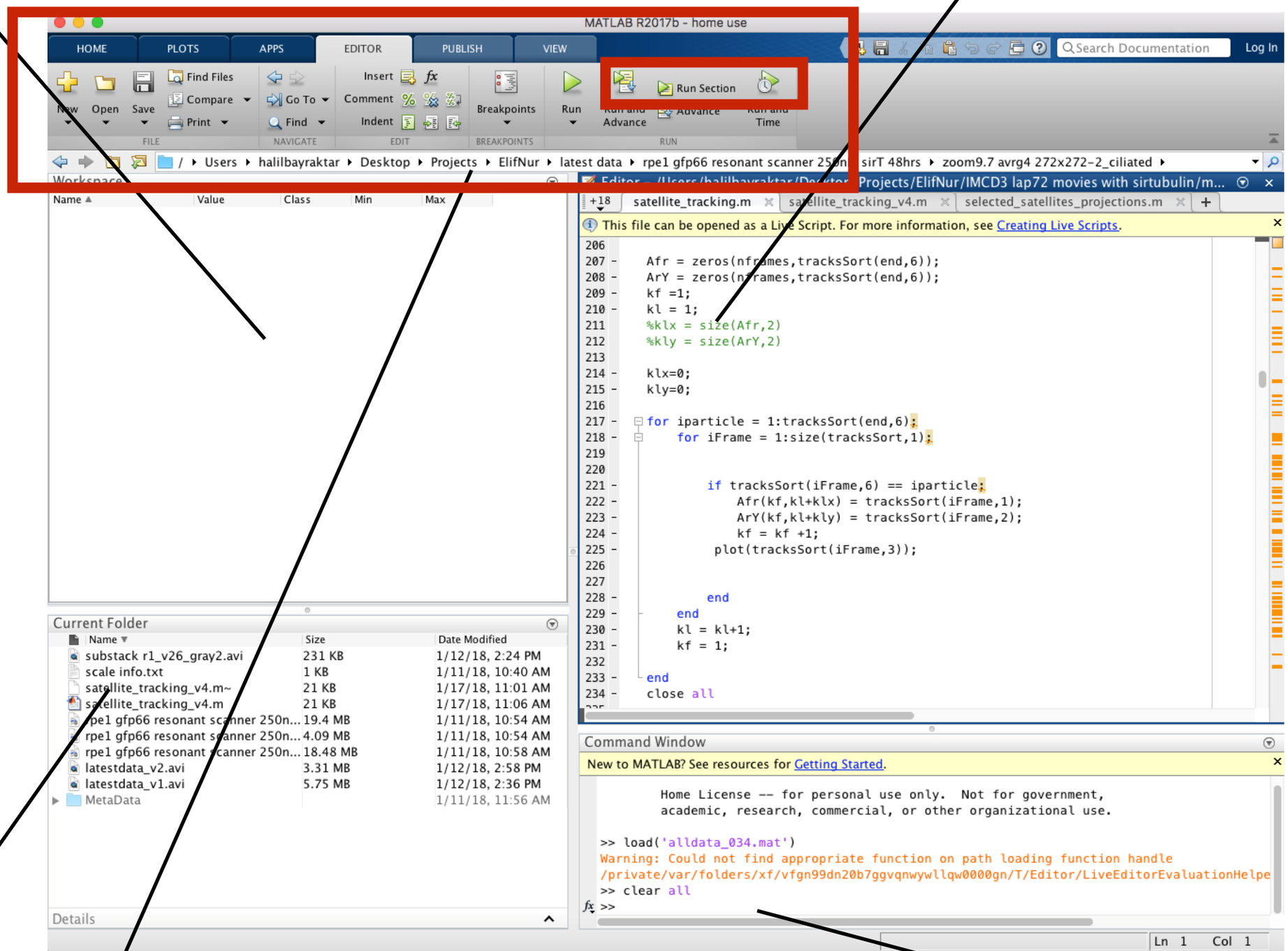
Lecture 2 –Intro to data analytics

MATLAB USER INTERFACE

Workspace
to show all variables

Menu and other options

Coding



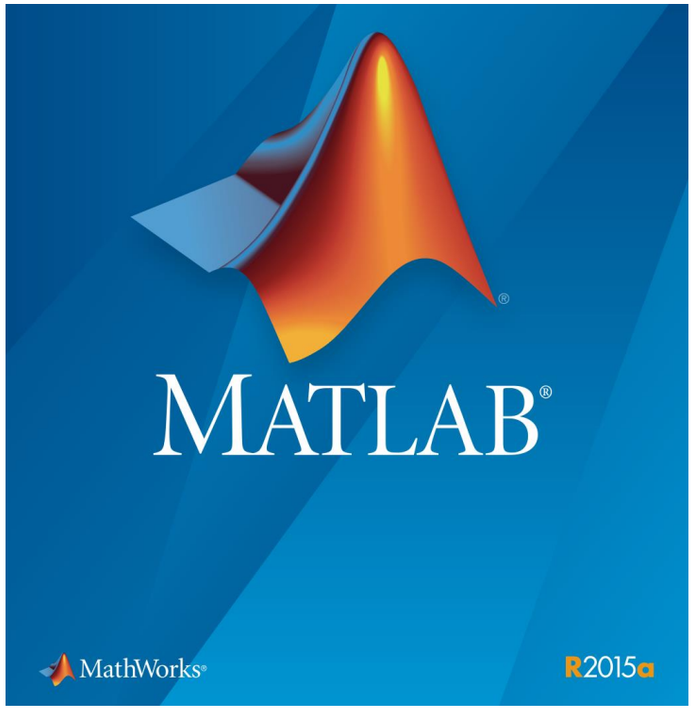
All files in a selected folder

Type commands here

.mat and .m files

.m : Files that contain the computer code. There are two kinds of .m files. Scripts and function. Script files do not need any input value to execute, however the function requires one or more arguments to return the output

.mat files: files that contains the data and variable names



Data analytics

- Biological data
- Finance data
- Computer engineering, image analysis
- Engineering, modelling
- Machine learning

Where do we use custom programming to determine the properties/dynamics of biological systems

1. Image/video processing
2. Genome analysis
3. Microarray analysis
4. Proteomics analysis
5. Advance graphics

Variables - data1

corr_coef_y | allmeansdata | p_value_o | data1

27130x34 table

	1 Gene	2 ARNA	3 TRNA	4 ARNA1	5 TRNA1	6 ARNA2	7 TRNA2	8 ARNA3	9 TRNA3	10 TRNA4	11 TRNA5
1	"LOC1024...	0	1.0045	4.0185	0.9162	2.9799	3.3377	1.3212	2.1511	1.0805	
2	"ZBTB42"	27.8394	37.1676	55.2547	30.2348	42.7112	32.2643	54.1705	48.3991	35.6578	50.26
3	"FCAMR"	1.1136	0	1.0046	0	0.9933	0	0	0	0	
4	"ZNF503-...	41.2024	35.1586	40.1853	16.4917	35.7582	32.2643	60.7767	23.6618	47.5438	40.84
5	"NFU1"	111.3578	123.5573	118.5465	133.7663	91.3821	86.7800	76.6315	120.4600	100.4903	68.07
6	"ELSPBP1"	0	0	0	0	0	0	0	0	0	
7	"ZRANB3"	190.4218	183.8291	141.6531	169.4984	155.9455	92.3428	118.9109	121.5355	152.3563	102.63
8	"MECR"	259.4637	291.3139	188.8708	237.2977	289.0455	189.1358	257.6403	253.8264	347.9341	271.24
9	"LOC1057...	0	0	0	0	0	0	0	0	0	
10	"LINC003...	2.2272	2.0091	3.0139	4.5810	6.9530	0	0	0	1.0805	4.18
11	"AARSD1"	1.1136	0	0	0	0	2.2251	0	0	0	
12	"DEXI"	485.5200	435.9663	492.2695	308.7619	511.5410	493.9782	478.2860	357.0779	467.8742	569.72
13	"DCHS1"	1.1559e+03	1.0035e+03	1.2116e+03	1.3138e+03	1.3479e+03	1.4652e+03	1.8788e+03	1.4842e+03	1.7224e+03	1.2724e+
14	"PSMD2"	1.2550e+03	1.8232e+03	1.3914e+03	1.4861e+03	1.5545e+03	1.3206e+03	1.5630e+03	1.1282e+03	1.4382e+03	1.5730e+
15	"GABRR1"	3.3407	4.0181	2.0093	1.8324	5.9597	8.9005	6.6062	2.1511	2.1611	4.18
16	"PKNOX2"	780.6181	676.0491	640.9550	244.6274	522.4672	406.0857	486.2134	287.1680	504.6126	799.07
17	"TIPARP"	309.5747	294.3275	372.7184	721.0553	238.3881	354.9078	395.0484	379.6641	298.2293	273.34
18	"ADAM20"	113.5849	91.4123	74.3427	89.7883	100.3216	160.2091	104.3773	73.1364	63.7519	97.39
19	"LOC2847...	0	0	0	0	0	0	0	0	0	
20	"MIR4715"	0	0	0	0	0	1.1126	0	0	0	

The dataset related with Alzheimer's disease was downloaded from Gene Expression Omnibus (http://www.ncbi.nlm.nih.gov/geo/): GSE104704.

2d data
Microarray data cell type vs gene expression

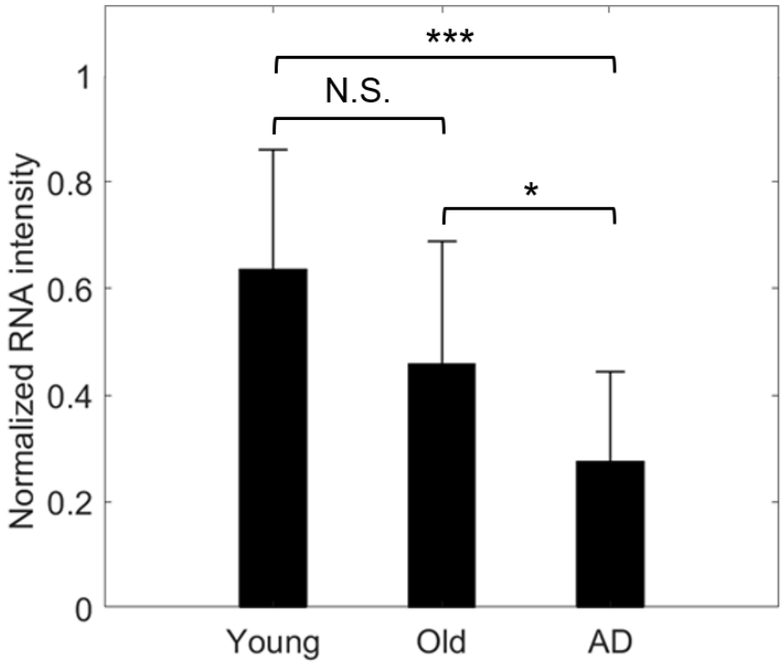
	1 GeneIDs	2 subdatanormalized1	3 subdatanormalized2	4 subdatanormalized3	5 subdatanormalized4	6 subdatanormalized5	7 subdatanormalized6	8 subdatanormalized7	9 subdatanormalized8	subdat
1	"LAPTM4B"	0.5499	0.5123	0.4750	0.6417	0.3751	0.2909	0.5757	0.4148	
2	"CXorf56"	0.4806	0.5591	0.3765	0.5342	0.4437	0.4128	0.4202	0.7452	
3	"RTCA"	0.4826	0.6229	0.4979	0.8292	0.5208	0.4182	0.4049	0.4738	
4	"TSPAN5"	0.5415	0.6838	0.4375	0.5692	0.6326	0.6206	0.5114	0.4138	
5	"UTP11"	0.4749	0.6544	0.5462	0.5711	0.4935	0.3337	0.3777	0.4133	
6	"ATXN10"	0.5995	0.7425	0.5714	0.5560	0.6368	0.4644	0.4894	0.6304	
7	"PRKACB"	0.5187	0.6406	0.3364	0.5173	0.4846	0.2804	0.2811	0.4022	
8	"C11orf58"	0.4531	0.5953	0.4352	0.7056	0.4311	0.4252	0.3790	0.4615	
9	"MICALL2"	0.3795	0.4109	0.7202	0.6093	0.7226	0.6862	1	0.8565	
10	"LAMTOR5"	0.7566	0.7240	0.5773	0.6603	0.5077	0.5051	0.7173	0.5259	
11	"C5orf30"	0.7122	0.7768	0.7121	0.6834	0.5793	0.3692	0.6684	0.4401	
12	"UROD"	0.5457	0.6806	0.5123	0.5921	0.5728	0.4124	0.4954	0.3697	
13	"NUDT21"	0.4438	0.6518	0.4035	0.5406	0.4282	0.3858	0.3713	0.3627	
14	"EIF2S1"	0.4938	0.6331	0.4381	0.4600	0.5108	0.3533	0.3776	0.4585	
15	"ATP6AP2"	0.4356	0.6230	0.3129	0.4570	0.3792	0.2408	0.2591	0.3001	
16	"AREL1"	0.6366	0.8741	0.6360	0.6675	0.7324	0.6043	0.5871	0.6385	
17	"NDUFB6"	0.5893	0.6998	0.4105	0.6198	0.5323	0.3912	0.3628	0.4431	
18	"EFL1"	0.5592	0.6251	0.5886	0.5635	0.6000	0.5142	0.6202	0.7045	
19	"AASDHP...	0.3772	0.6394	0.3226	0.4832	0.4696	0.2831	0.2917	0.3238	
20	"KANK2"	0.4338	0.4338	0.9057	0.5561	0.7931	0.7422	1	0.6848	
21	"PLIN4"	0.4174	0.3096	0.5866	0.3409	0.5844	0.8701	1	0.3954	
22	"KLK14"	0.8725	0.2705	0.4919	0.2243	0.3161	0.5175	0.5823	0.5793	
23	"VPS35"	0.5232	0.6101	0.3973	0.5165	0.4300	0.3487	0.3195	0.3565	
24	"LINC01372"	0.7295	0.9006	0.7967	0.6634	0.8905	0.8823	0.8656	0.7046	
25	"TUBA3D"	1	0.3061	0.5961	0.3232	0.5416	0.8920	0.5721	0.2415	
26	"COL28A1"	0.7196	0.7934	0.7935	0.7456	0.8083	0.8255	0.4954	0.9010	
27	"HDAC10"	0.5768	0.4766	0.7471	0.3489	0.6483	1	0.7115	0.4978	

27140 genes and 30 people, how to analyze data?

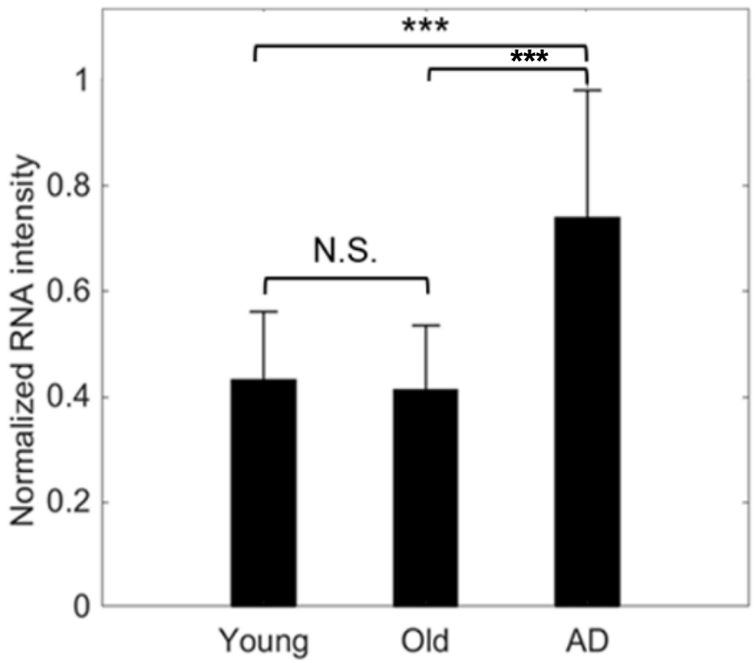
The dataset related with Alzheimer's disease was downloaded from Gene Expression Omnibus (http://www.ncbi.nlm.nih.gov/geo/): GSE104704.

Data inference and statistical methods: From big data to results

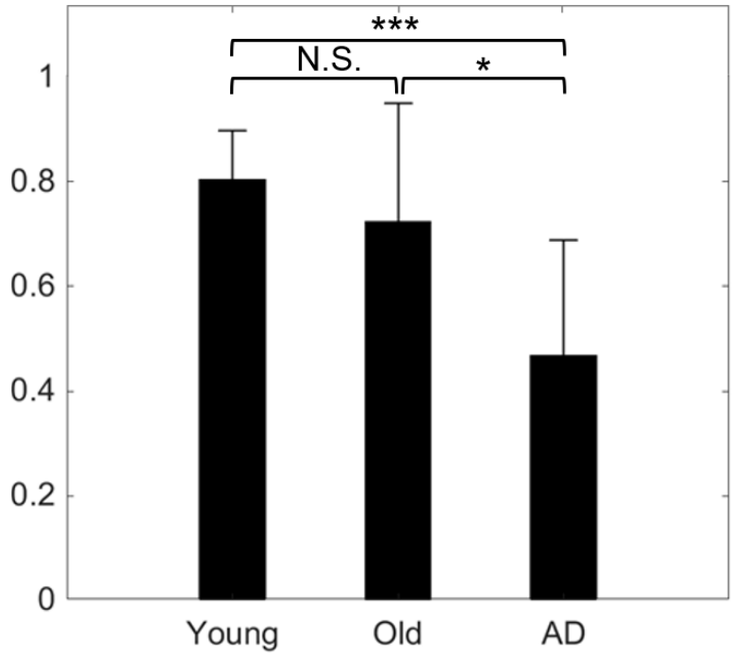
- Three genes were detected to have distinct RNA levels in AD patients.



LOC283440



PRKACB

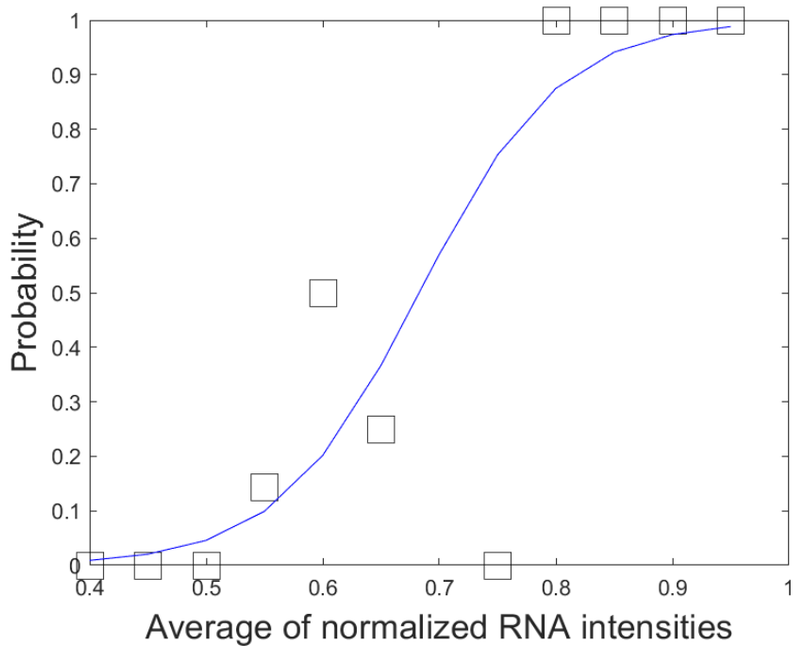


LINC01372

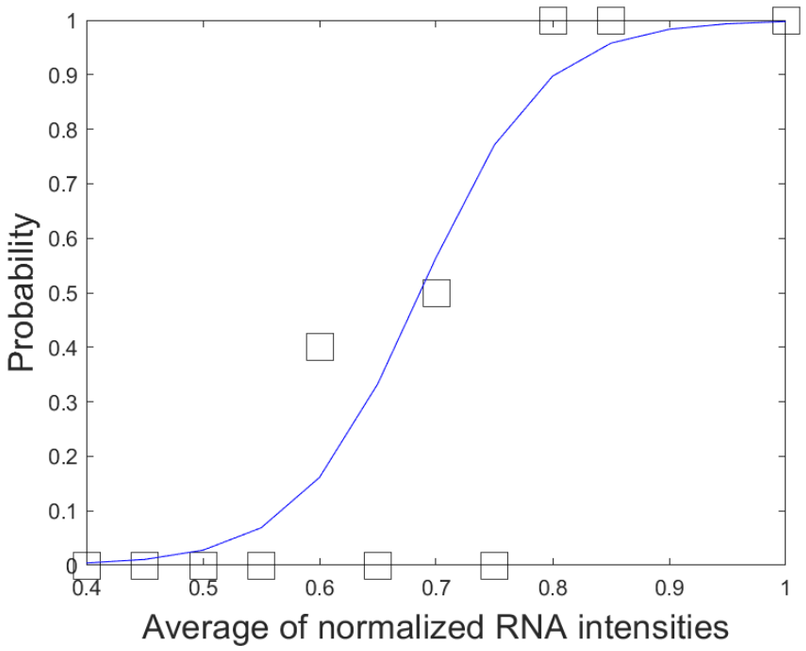
Ad= Alzheimer disease

Finding genes that are associated with Alzheimer

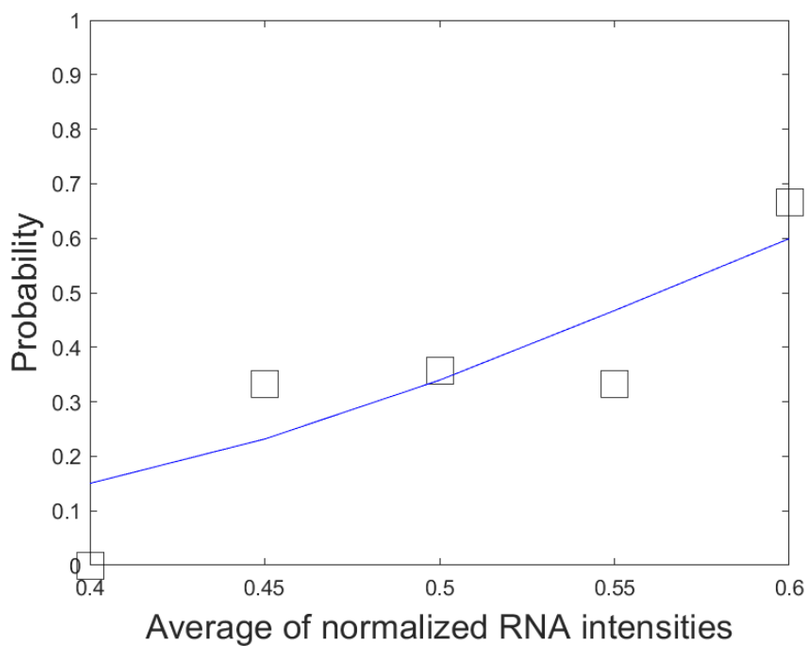
- logistic regression models were used.
- Statistical P-values of logit coefficients were computed



By using the first seven genes
(p < 5e-05)



By using the first three genes
(p < 3e-05)



By using the last nine genes
(0.004 < p < 0.005)

Arrays operations
Example 2: Image analysis

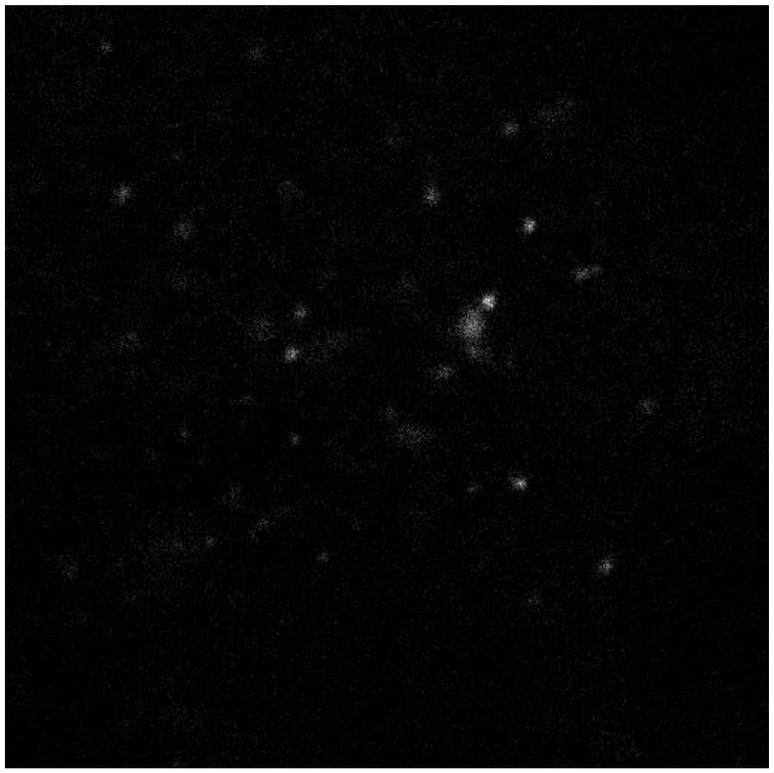
85	78	75	79	82	81	80	81	81	89
74	65	57	56	55	52	49	50	71	74
73	63	56	56	58	57	58	61	66	62
64	56	52	55	59	61	63	66	61	54
68	60	54	56	57	55	53	54	53	47
73	64	57	56	57	54	53	54	51	49
66	55	45	43	45	47	50	54	51	54
80	66	52	47	47	49	54	60	47	55
83	73	59	49	48	53	58	60	55	52
74	67	61	58	53	48	49	54	54	52

Challenges in Fluorescence Microscopy

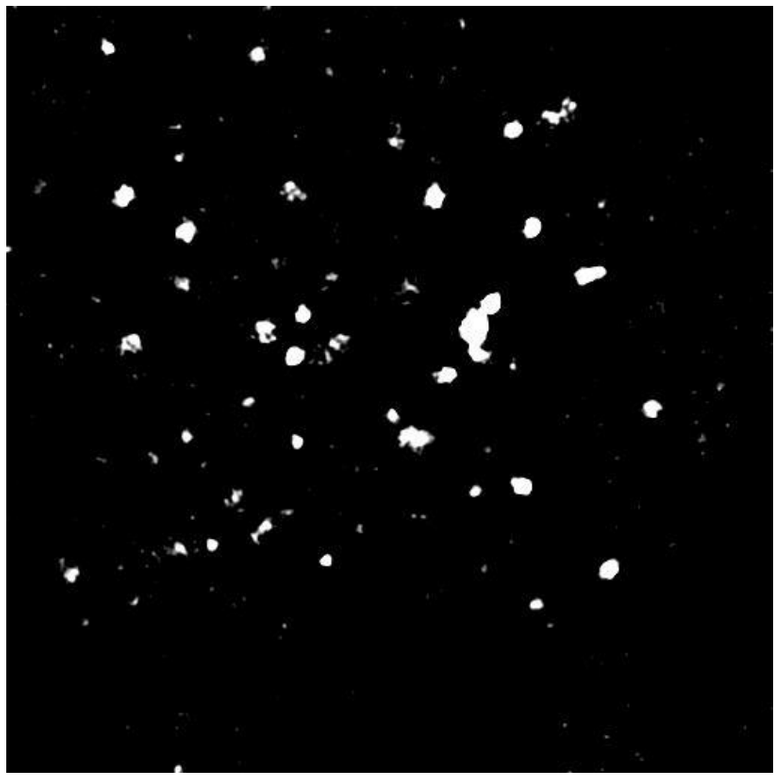
Although we use high-end and super expensive microscopes, they are not perfect.

- Low signal to noise ratio
- Some issues: Blur images, pixel noise, focus loss, diffraction issues etc.
- Solution: Post-processing of image/videos

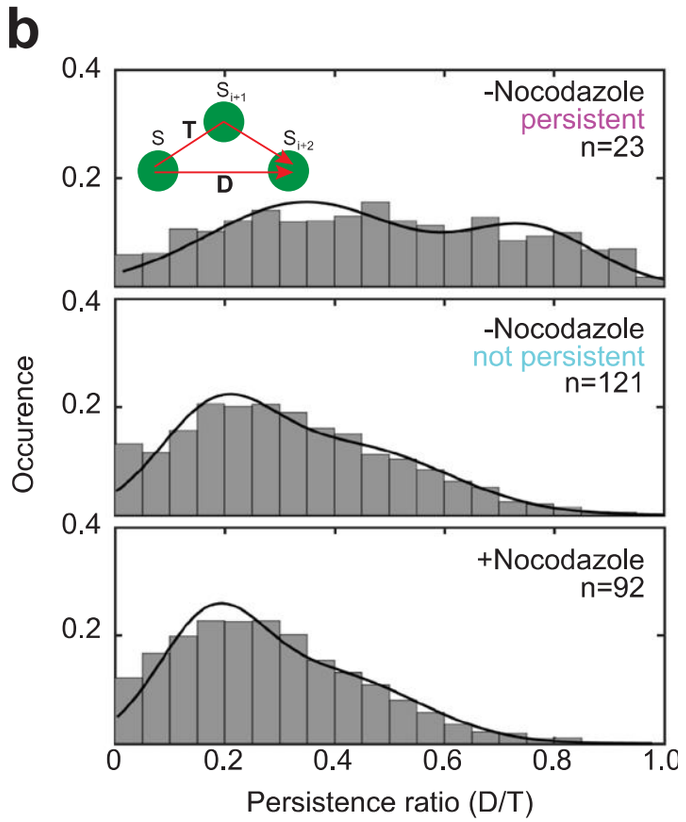
Technical term: Bandpass (low-pass) filter was used to remove noise.



Raw data:Pre-processing

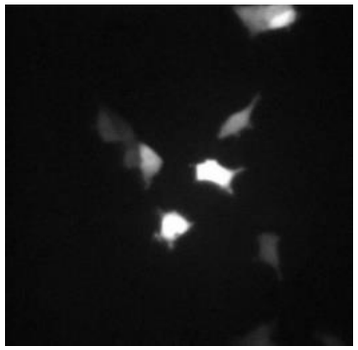


Post-processing

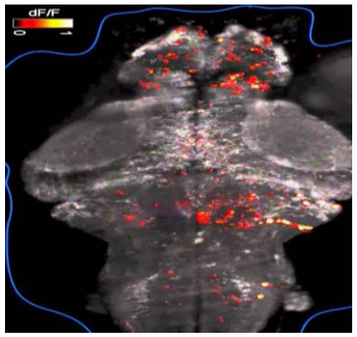


Conkar et al. scientific reports 2019

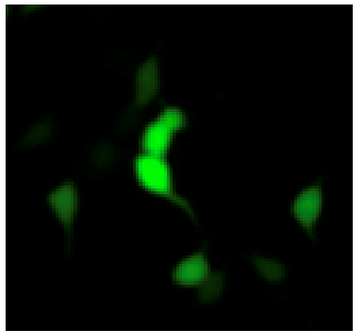
Why is the image processing intrinsically difficult?

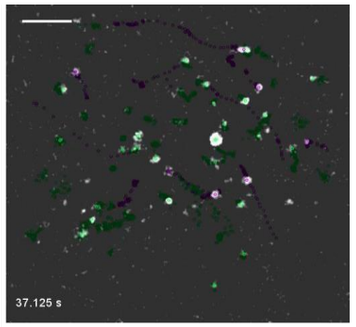


Intensity levels

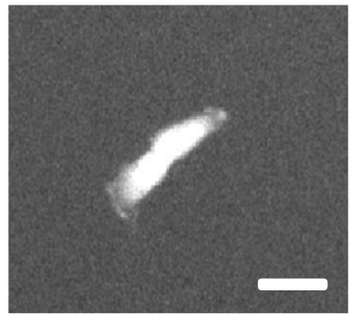


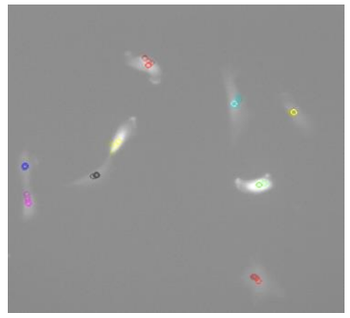
Ahrens et al 2013





Conkar et al 2019



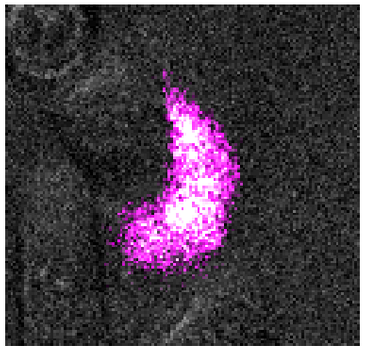


Cell splitting and apoptosis

Micro

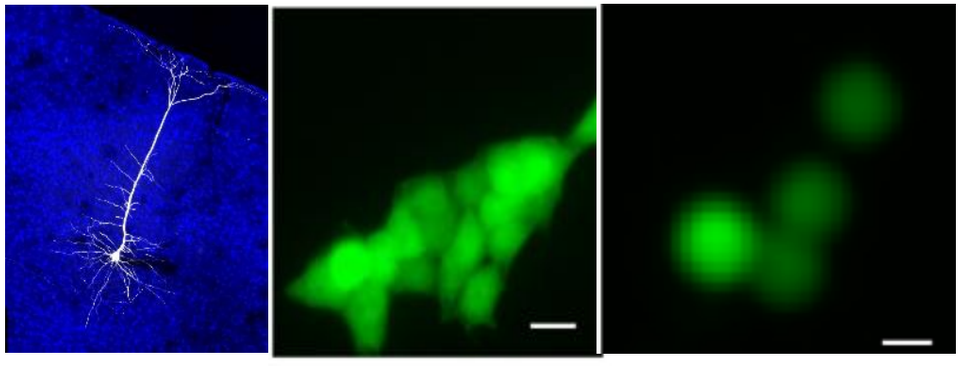
Object size

Nano



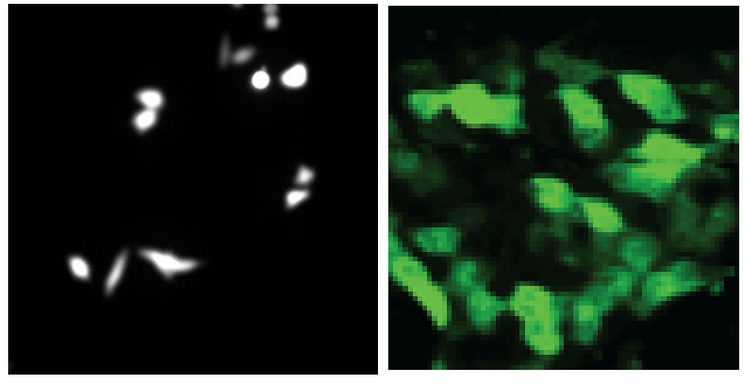
Photodamage

- Image analysis



Shape

Uneven labeling



Confluency

Printing Operators: disp, fprintf, sprintf

```
%%
x=2
disp(x)
disp('matlab')
fprintf('matlab course %i', x)
fprintf('matlab course %f', x)
fprintf('matlab course %0.10f', x)
y=sprintf('matlab course %0.10f', x)
```

Operators

Sum
Division
Multiplication

```
a=[1 2;3 4; 8 0]
b=[5 6;7 8;9 10]
c=a+b
d=a-b

e=a+5
f=a.*5
g=a./5
h=a.*b
h=a/b
```

Array addressing: select a region from matrix (2D array)

Column1 Column2

row 1
row 2

	1	2	3	4	5
1	17	24	1	8	15
2	23	5	7	14	16
3	4	6	13	20	22
4	10	12	19	21	3
5	11	18	25	2	9
6					
7					

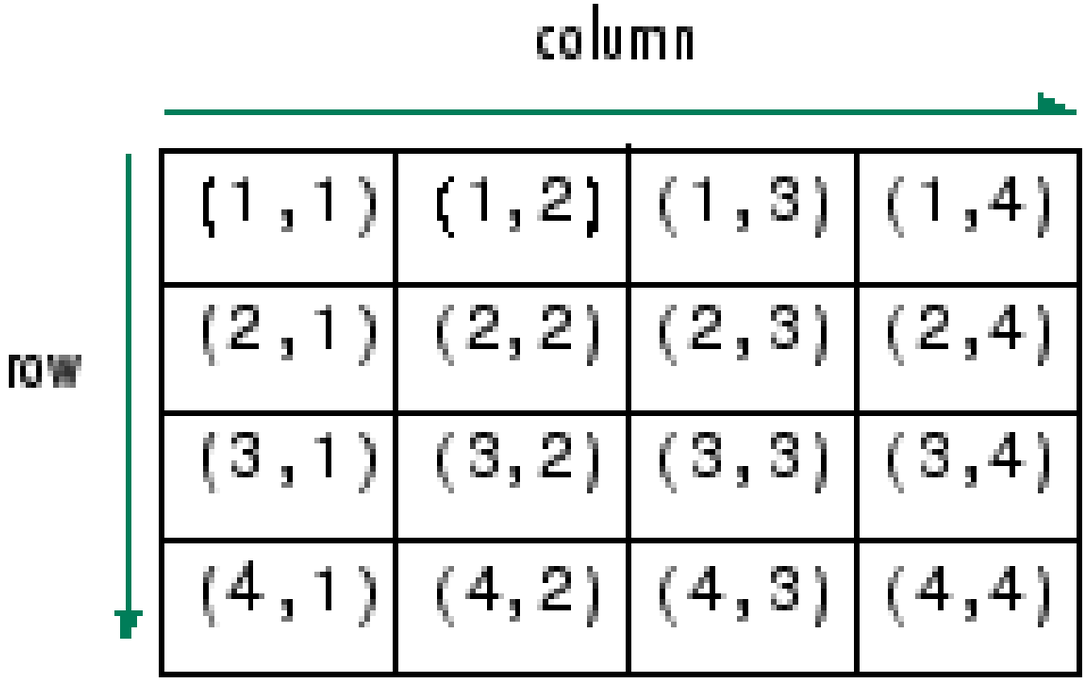
Row Column

b=a([3,4],[1,2,3])

b =

4 6 13
10 12 19

An array having more than two dimensions is called a multidimensional array




```
x=[1 2;3 4]

x =

 1     2
 3     4
```

Exam scores

MD1 MD2

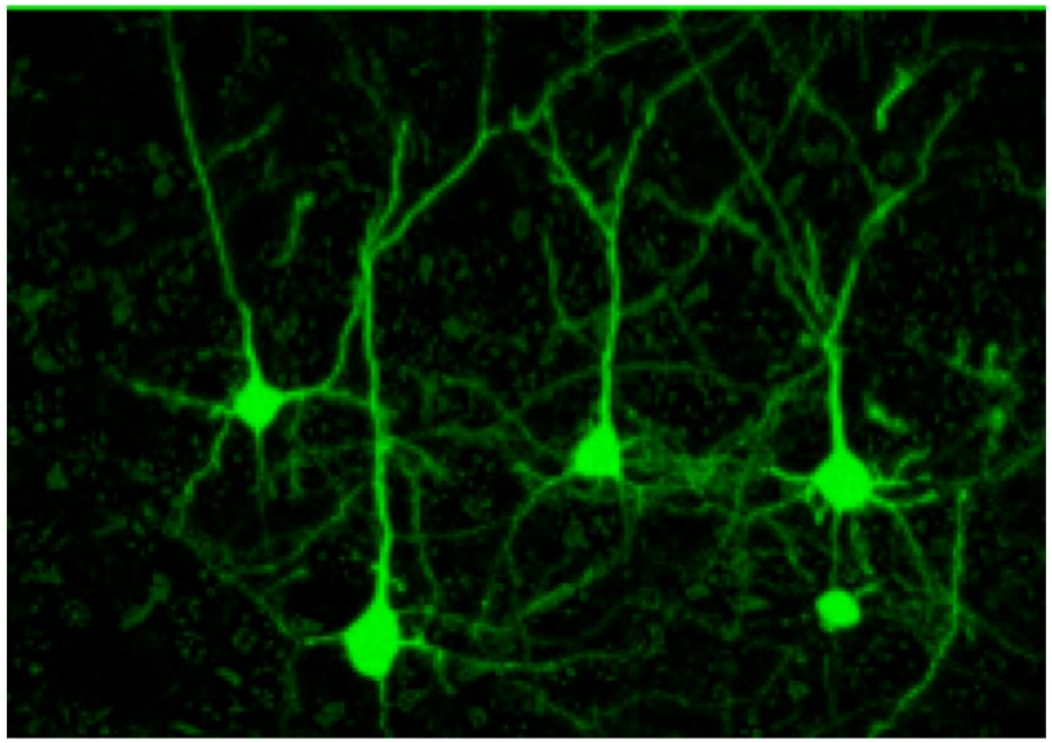
2 columns

20 rows

71	61
86	88
71	97
72	70
82	100
66	96
99	70
69	91
63	99
62	70
99	90
77	83
69	100
92	75
67	71
65	74
65	80
84	96
70	62
94	65

Examples (images)

2D arrav

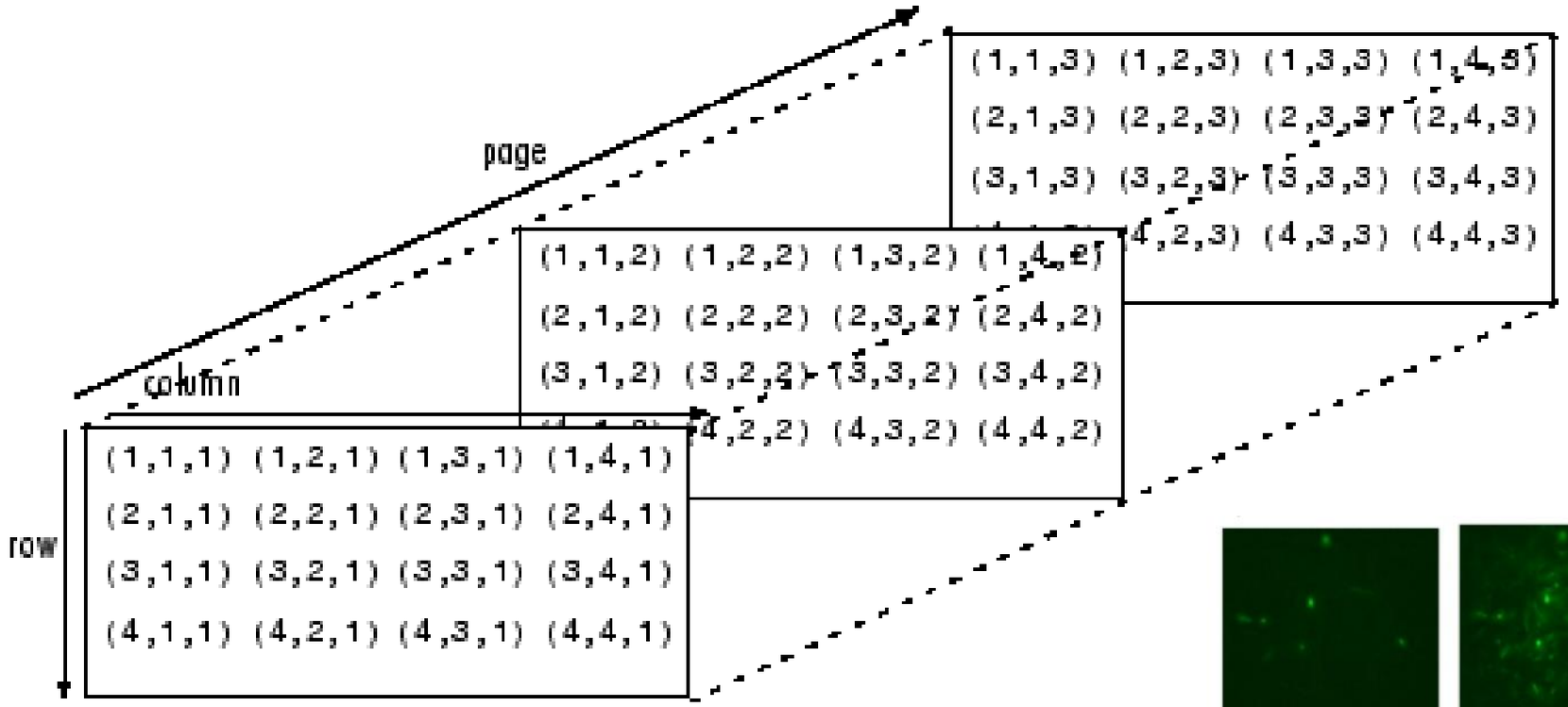


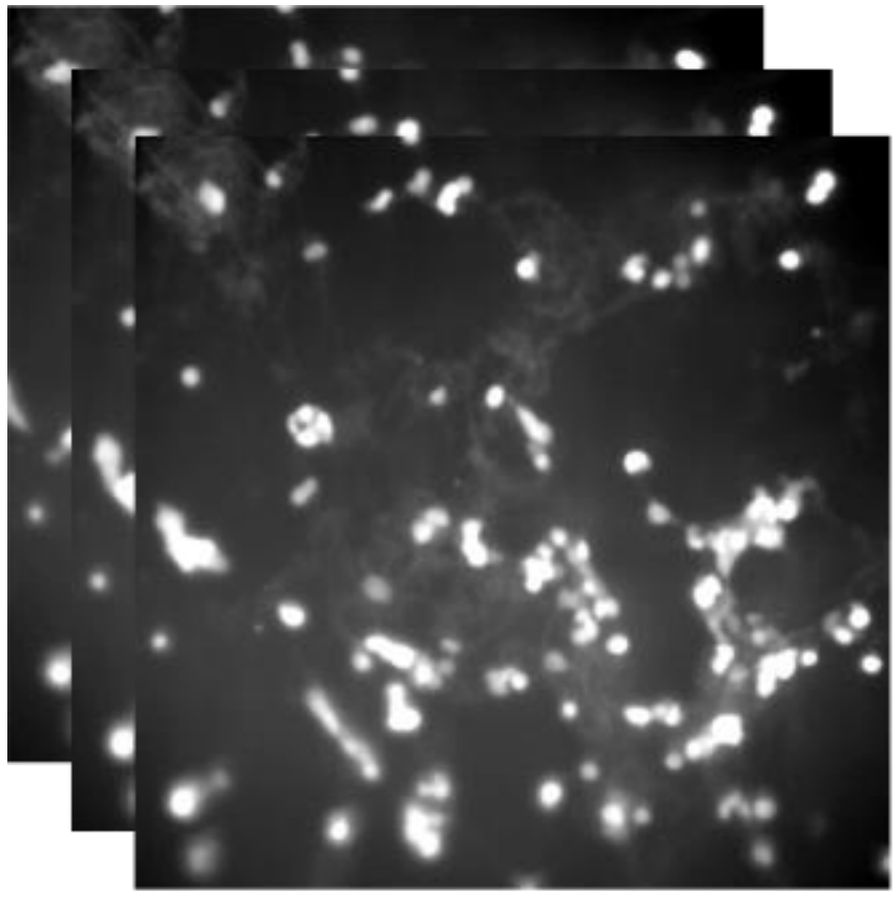
10 by 10 pixel – upper corner of the image

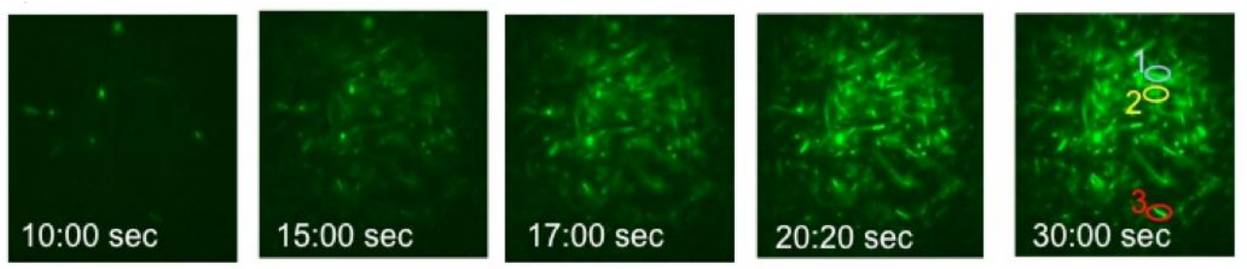
85	78	75	79	82	81	80	81	81	8
74	65	57	56	55	52	49	50	71	74
73	63	56	56	58	57	58	61	66	62
64	56	52	55	59	61	63	66	61	54
68	60	54	56	57	55	53	54	53	47
73	64	57	56	57	54	53	54	51	4
66	55	45	43	45	47	50	54	51	54
80	66	52	47	47	49	54	60	47	55
83									52
74									52

0	**1**	**0**	**0**	**1**	**0**	**0**
0	1	0	0	1	0	1
0	1	0	0	1	0	1
0	1	1	0	1	1	1

3d Array
Video files



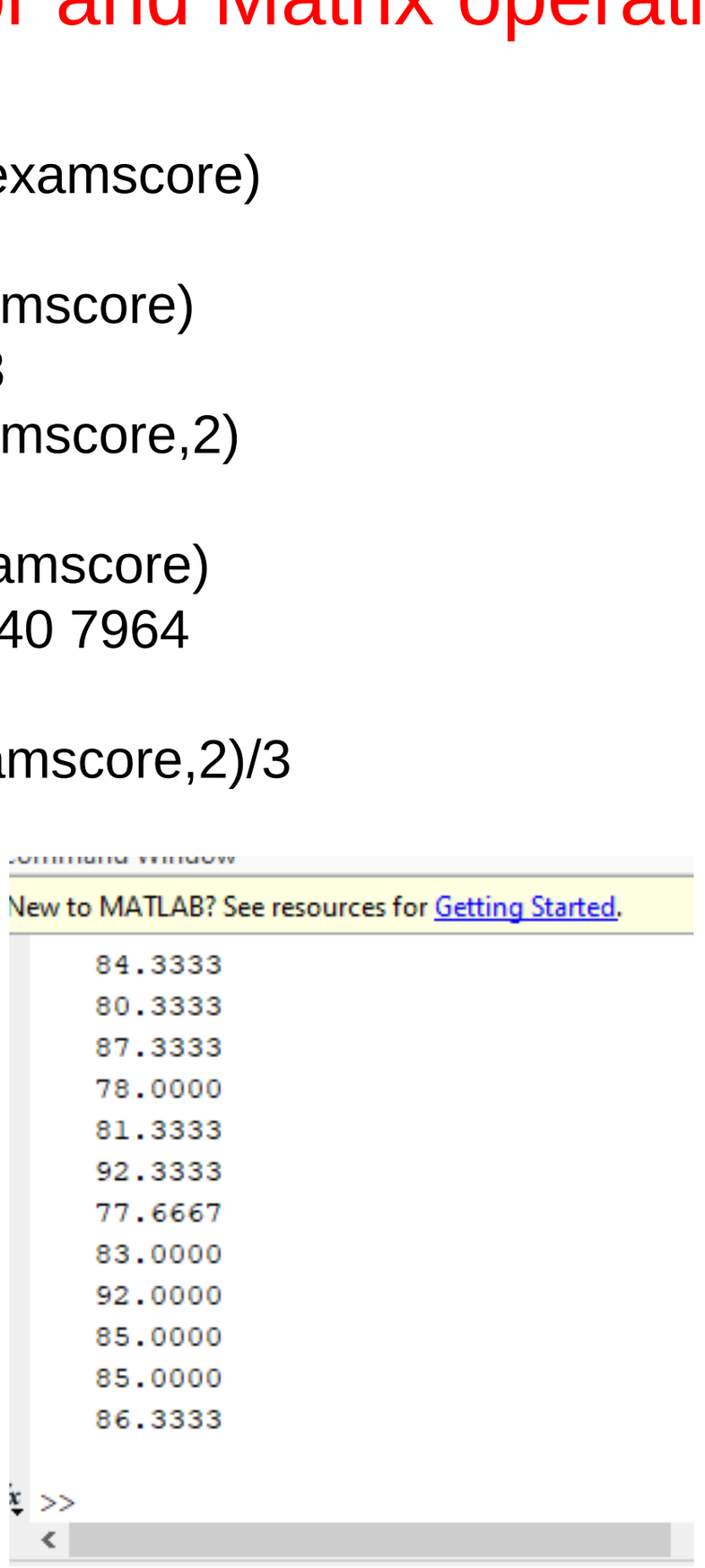


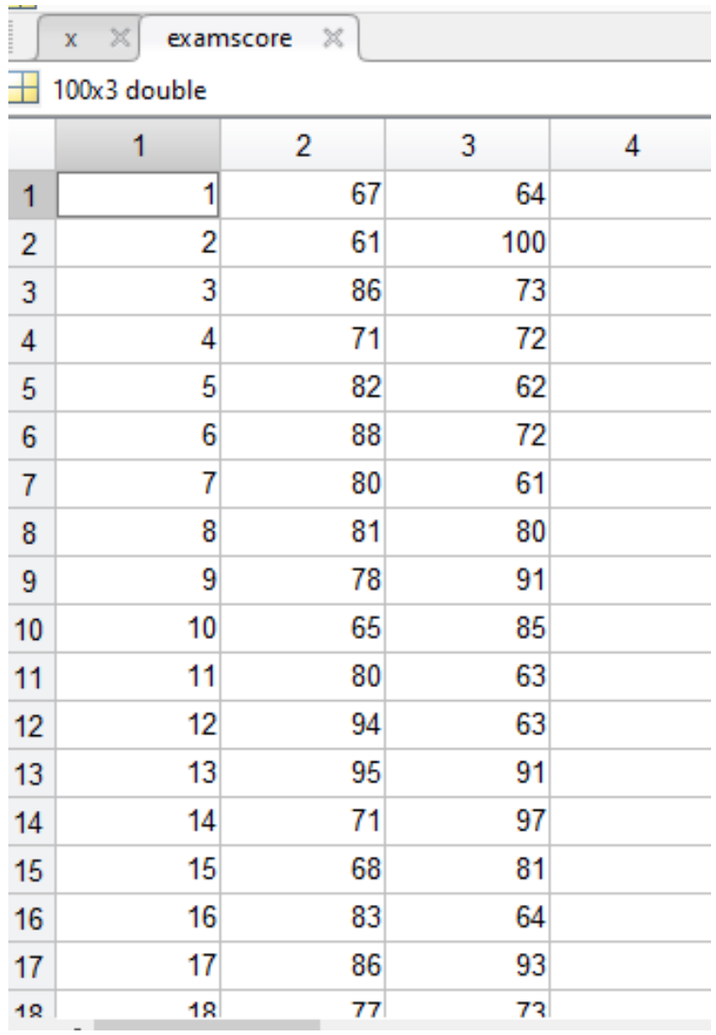


Vector and Matrix operations

```
length (examscore)
100
size(examscore)
100      3
size(examscore,2)
3
Sum(examscore)
5050 8340 7964

sum(examscore,2)/3
```

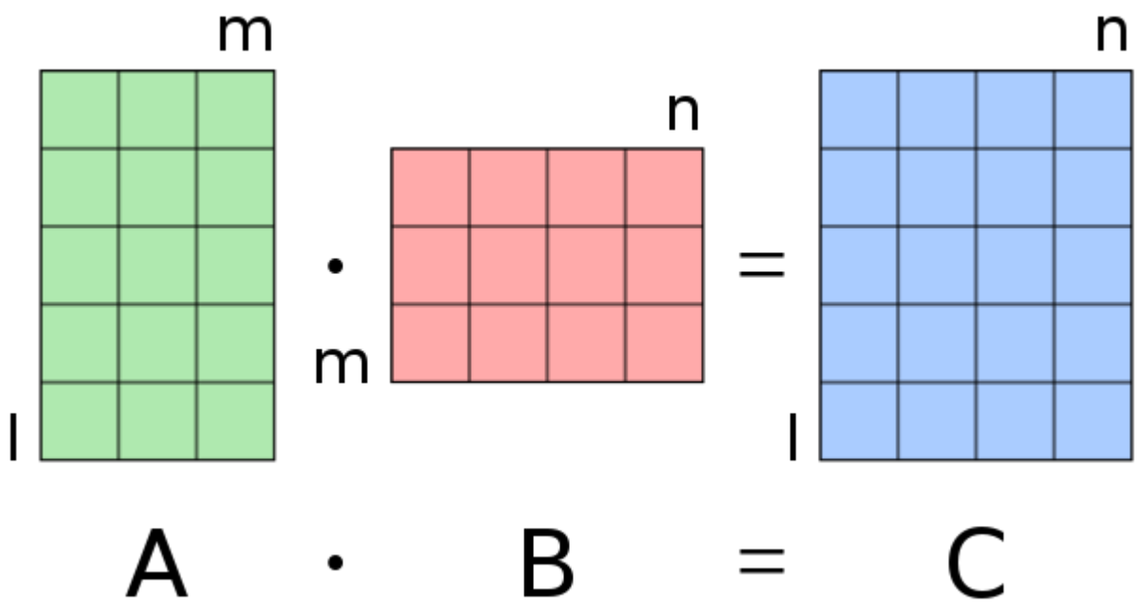
?

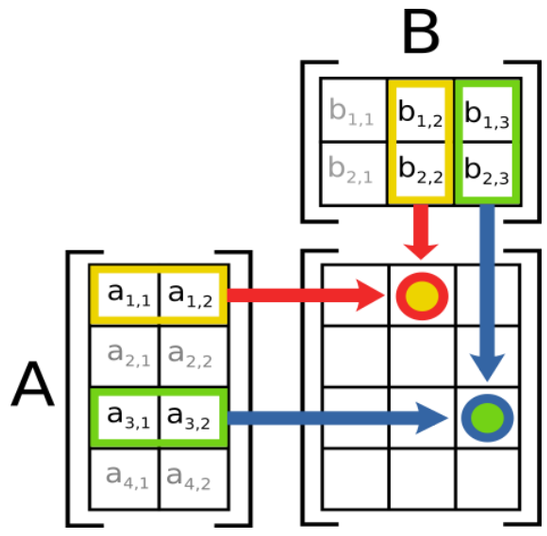




	1	2	3	4
1	1	67	64	
2	2	61	100	
3	3	86	73	
4	4	71	72	
5	5	82	62	
6	6	88	72	
7	7	80	61	
8	8	81	80	
9	9	78	91	
10	10	65	85	
11	11	80	63	
12	12	94	63	
13	13	95	91	
14	14	71	97	
15	15	68	81	
16	16	83	64	
17	17	86	93	
18	18	77	73	

Vector and matrix operations





- * multiplication, scalar or matrix-matrix or scalar-matrix
- .* element by element multiplication
- ^ power of a scalar
- .^ power element by element
- ‘ transpose of an array
- .’ transpose of an array

x=[1,2,5;3,4,6]
y= [1,2;3,4;5,6]
z=x*y

t=x.*x
k=x.^2

```
37
38      %%
39      %% Vector and matrix operations
40 -    x=[1,2,5;3,4,6]
41 -    y= [1,2;3,4;5,6]
42 -    z=x*y
43 -    t=x.*x
44 -    k=x.^2
45
46
```

New to MATLAB? See resources for Getting Started.

```
x =

     1     2     5
     3     4     6


y =

     1     2
     3     4
     5     6


z =

    32    40
    45    58

fx >>
```

```
t =

     1     4    25
     9    16    36


k =

     1     4    25
     9    16    36

fx >>
```

Transpose of matrix

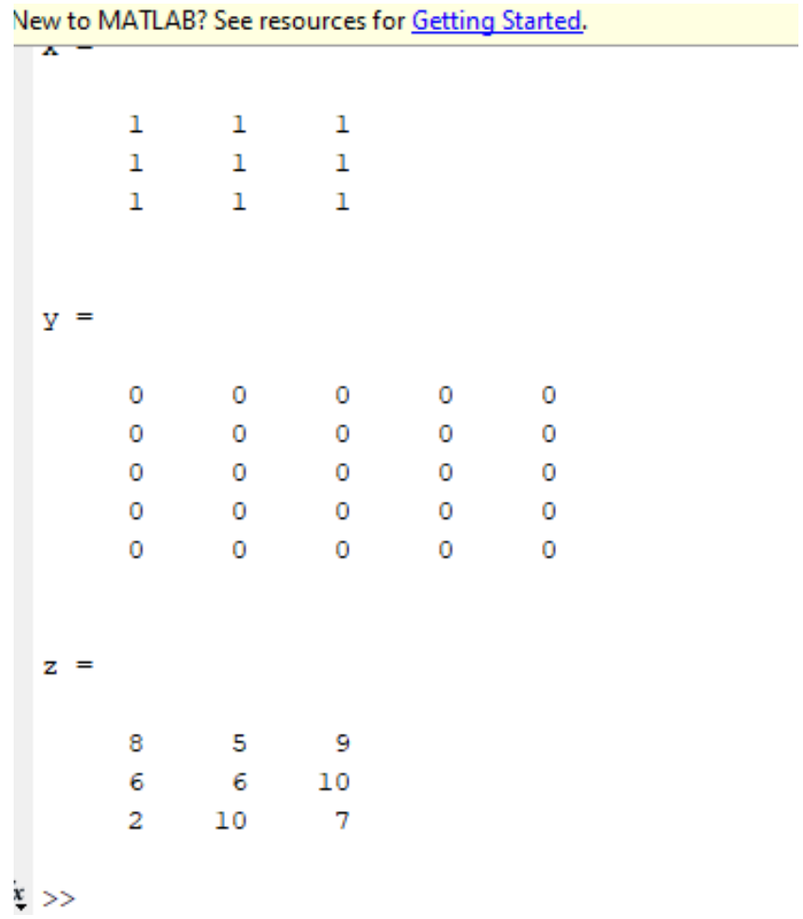
```
x=[1 2;3 4;5 6]
    x'
```

```
x =

     1     2
     3     4
     5     6
```

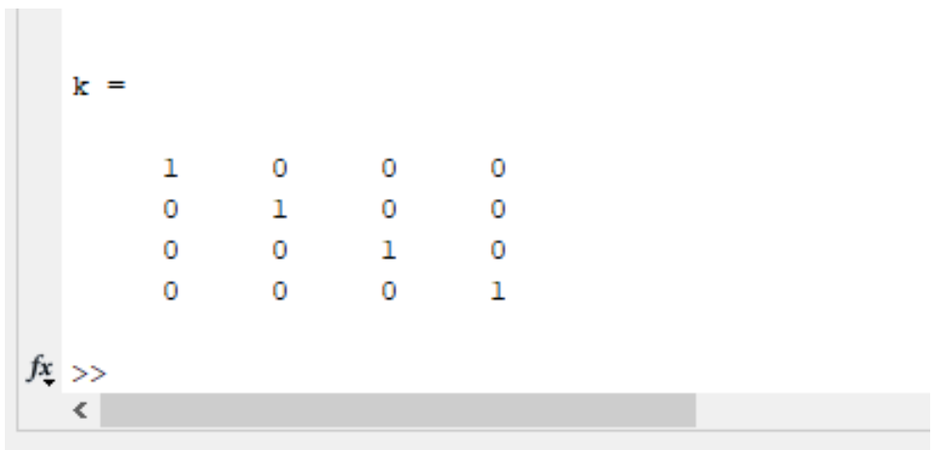
```
ans =

     1     3     5
     2     4     6
```

Special matrices

x=ones(3,3)
y=zeros(5,5)
z=randi([1,10],3,3)



k=eye(4,4)



Creating arrays with other arrays

```
x=ones(3,3)
y=zeros(3,3)
m=[x,y]
```

```
Command Window
New to MATLAB? See resources for Getting Started.

x =

     1     1     1
     1     1     1
     1     1     1

y =

     0     0     0
     0     0     0
     0     0     0

m =

     1     1     1     0     0     0
     1     1     1     0     0     0
     1     1     1     0     0     0

fx >>
```

```
x=ones(3,3)
y=zeros(3,3)
m=[x;y]
```

```
New to MATLAB? See resources for Getting Started.

     1     1     1
     1     1     1
     1     1     1

y =

     0     0     0
     0     0     0
     0     0     0

m =

     1     1     1
     1     1     1
     1     1     1
     0     0     0
     0     0     0
     0     0     0

fx >>
```

Ones and Zeros Matrix

A matrix having all its numbers as 1 or 0 make up a ones and zeros matrix, respectively:

```
a=ones(3,3)
b=zeros(2,2)
```

```
a =

     1     1     1
     1     1     1
     1     1     1

b =

     0     0
     0     0
```

3D array

```
c=ones(3,3,3)

c(:,:,1) =

     1     1     1
     1     1     1
     1     1     1

c(:,:,2) =

     1     1     1
     1     1     1
     1     1     1

c(:,:,3) =

     1     1     1
     1     1     1
     1     1     1
```

Concatenation Along a Dimension (cat function)
Rearrange arrays as a row, column or 3d matrix

t1=cat(1,x,y)
t2=cat(2,x,y)
t3=cat(3,x,y)

```
New to MATLAB? See resources for Getting Started.

t1 =

     1     1     1
     1     1     1
     1     1     1
     0     0     0
     0     0     0
     0     0     0


t2 =

     1     1     1     0     0     0
     1     1     1     0     0     0
     1     1     1     0     0     0


t3(:,:,1) =

     1     1     1
     1     1     1
     1     1     1


t3(:,:,2) =

     0     0     0
     0     0     0
     0     0     0

fx >>
```

Concatenation Along a Dimension

```
a=[1 2;3 4]
b=[5 6;7 8]
```

Method 1

```
c=cat(1,a,b)
c=cat(2,a,b)
```

Method 2

```
d=[a,b]
e=[a ; b]

f=[a a.*2]
```

```
c =
     1     2
     3     4
     5     6
     7     8
c =
     1     2     5     6
     3     4     7     8
d =
     1     2     5     6
     3     4     7     8
e =
     1     2
     3     4
     5     6
     7     8
```

```
f =

     1     2     2     4
     3     4     6     8
```

Build-in Functions

What is the average of first and second midterm of each students?

```
ans =

mean(examscore,2)

   81.0000
   82.0000
   80.0000
   80.5000
   70.5000
   93.0000
   79.0000
   90.0000
   87.5000
   82.5000
   75.5000
   67.0000
   91.5000
   84.5000
   87.0000
   82.5000
   80.5000
   76.0000
   82.0000
   81.0000
```

```
   80   82
   91   73
   89   71
   74   87
   68   73
   86  100
   77   81
   82   98
   94   81
   65  100
   62   89
   74   60
   84   99
   90   79
   76   98
   78   87
   80   81
   76   76
   97   67
   63   99
```

mean(examscore,1)

What is the average of first and second midterm?

```
   79.3000   84.0500
```

What is the average exam score of all exams?

```
mean(mean(examscore))

    ans =

    81.6750
```

Other build-in functions

```
80   82
91   73
89   71
74   87
68   73
86  100
77   81
82   98
94   81
65  100
62   89
74   60
84   99
90   79
76   98
78   87
80   81
76   76
97   67
63   99
```

```
max(examscore)
min(examscore)
examsum=sum(examscore)

% how many students i have
size(examscore)
examsum/size(examscore,1)

length(examscore)
```

```
ans =
97   100
ans =
 62    60

examsum =
1586        1681
   ans =
    20     2

ans =
    20
```

Vector slicing

you can find a small section of array

```
b=examscore(1:10,1)

c=examscore(5:end,1)
c=examscore(5:20,1)
c=examscore(5:21,1), error

d=examscore(1,:)
```

```
b =
   78
   73
   64
   80
   80
   76
   60
   89
   86
   76
```

```
c =
   80
   76
   60
   89
   86
   76
   99
   70
   61
   87
  100
   82
   83
  100
   82
   96
```

```
d =
   78   88
```

Finding and selecting elements in a matrix

```
z=magic(4)
z([1,2],[3,4])
z(1,1)
z(2)
```

z =

16	2	3	13
5	11	10	8
9	7	6	12
4	14	15	1

ans =

3	13
10	8

ans =

16

ans =

5

z =

16	2	3	13
5	11	10	8
9	7	6	12
4	14	15	1

Index number 2

Data sorting

sort the elements of each column in a particular order.

examscores =

98	76	71	83	70	85	89	83	71	63

sort(examscores,'ascend')

ans =

63	70	71	71	76	83	83	85	89	98

sort(x,'descend')

ans =

98	89	85	83	83	76	71	71	70	63

Sorting rows

examscores =

94	60
65	88
80	82
100	77
67	81
95	70
62	97
65	88
76	74
60	65

sortrows(examscores,1)

ans =

60	65
62	97
65	88
65	88
67	81
76	74
80	82
94	60
95	70
100	77

sortrows(examscores,2)

ans =

94	60
60	65
95	70
76	74
100	77
67	81
80	82
65	88
65	88
62	97

Sort for different rows or columns

y =

2	10	9
1	5	0
6	-4	3

sort(y,1)

ans =

1	-4	0
2	5	3
6	10	9

sort(y,2)

ans =

2	9	10
0	1	5
-4	3	6

Find an information in an array

returns the row and column indices of non-zero entries in a matrix.

```
1   2   2
4   6   9
1   10  9
```

```
find(x>7)

ans =

     6
     8
     9
```

```
[ans]=find(x>7)
```

```
find(x==10)
```

```
[r,c,l]=find(x>7)

row =

     3
     2
     3

col =

     2
     3
     3

v =

3×1 logical array

   1
   1
   1
```

Rounding the elements of an array to the nearest integer

```
x=1.5001
floor(x)
ceil(x)
```

```
ans =

     1
```

```
x=rand(3)
ceil(x)
```

```
x =

    0.4254    0.9915    0.1293
    0.9842    0.7764    0.7471
    0.9800    0.3136    0.6842
```

```
ans =

     1     1     1
     1     1     1
     1     1     1
```

```
round(x)
```

```
x =

    3.3468    3.0285    1.0143
    2.2344    3.6283    3.4876
    2.2025    1.8237    1.0050
```

```
ans =

     3     3     1
     2     4     3
     2     2     1
```

Reshaping a Matrix

The number of rows and columns in a matrix can be changed provided the total

number of elements remains the same.

a=randi([1,10],3,3)

b=reshape(a,9,2,1)

2	2
7	3
1	8
8	10
9	5
8	9
7	10
10	

b=reshape(a,1,9)

8	5	7						
10	8	10						
9	9	10						
8	10	9	5	8	9	7	10	10

Data types (2)

Logical

boxcar_kernel	*1x51 double*	double	408
c	*3x3 logical*	logical	9
centers	*1x1 cell*	cell	47872

```
a =
     4     3     5
    10     9     4
     6    10     9
b =
     8     5     7
     1     8     7
     9     6     1
c =
3×3 logical array
   0   0   0
   1   1   0
   0   1   1
```

3x3 logical

	1	2	3
1	0	0	0
2	1	1	0
3	0	1	1
4			

```
a=randi([1,10],3)
b=randi([1,10],3)
c=b<a
```

Practical Skills 1

If you write a code at large scale (moren than 50 lines), you should use meaningful names.

For codes having few lines you can user very short names.

x,y,z usually used for double

i,j,k usually used for integers

proteinexps

drugdosage

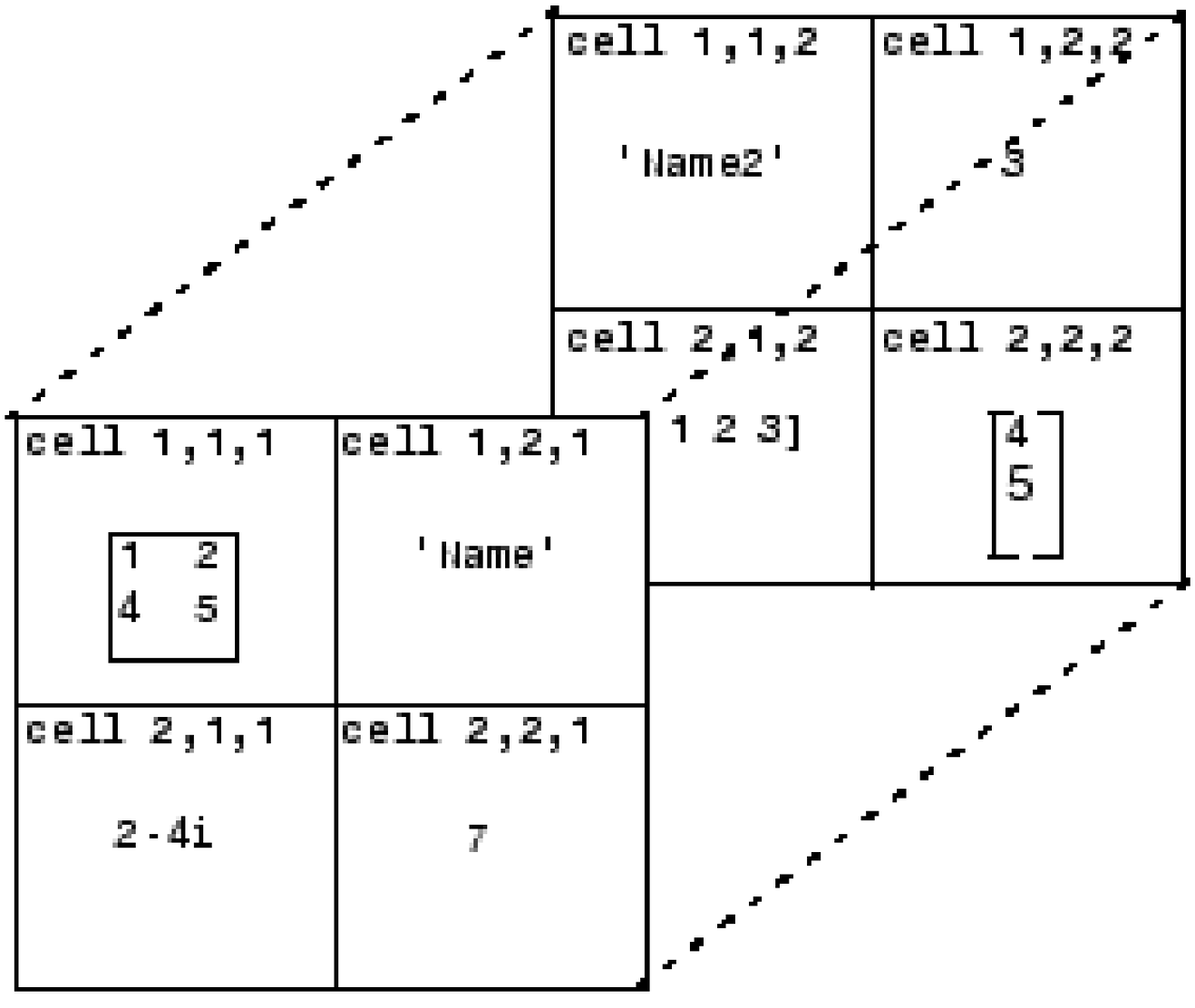
Remember that some programs have millions of lines of code

if you do use useful names, noone including yourself will understand what your code is

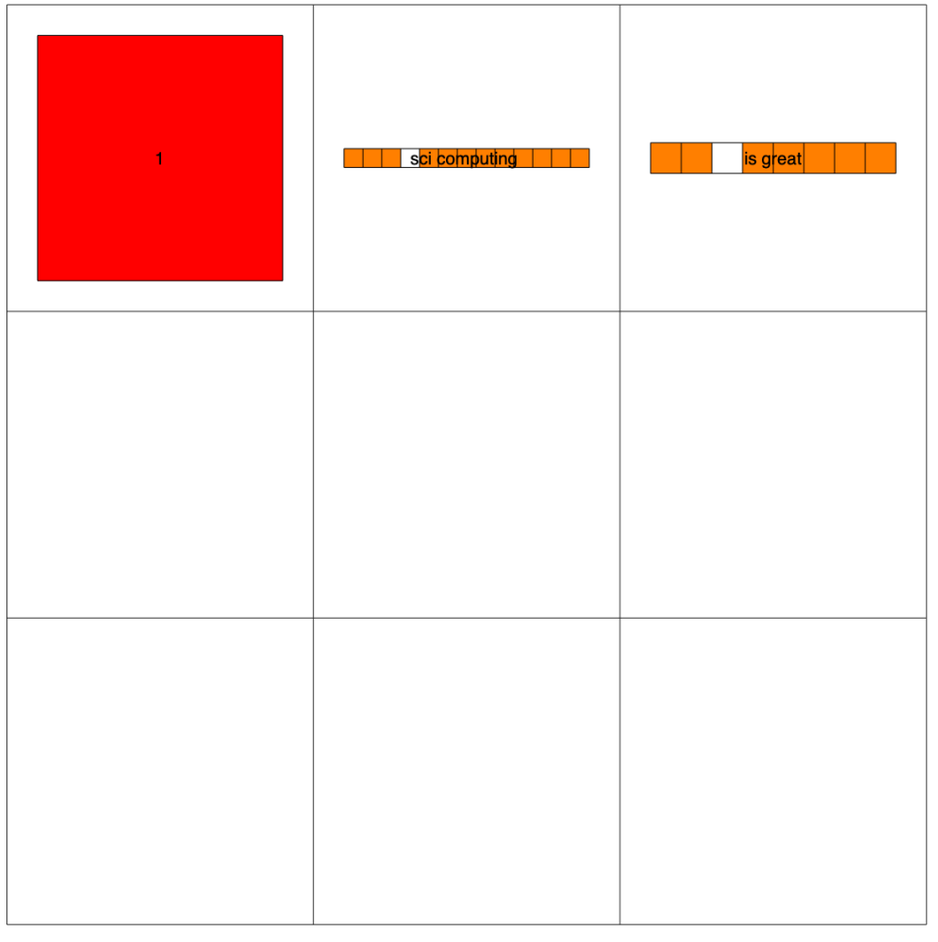
Cell Arrays

Cell arrays are arrays of cells where each cell stores an array. Within a cell, elements

must be the same type (because cells store arrays), but two cells may have different types.



y{1,1}=([1])
y{1,2}='sci computing'
y{1,3}='is great'

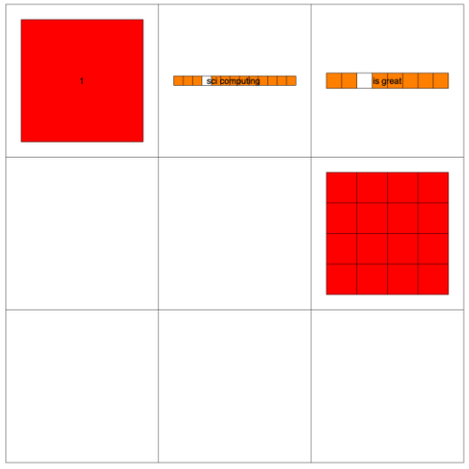


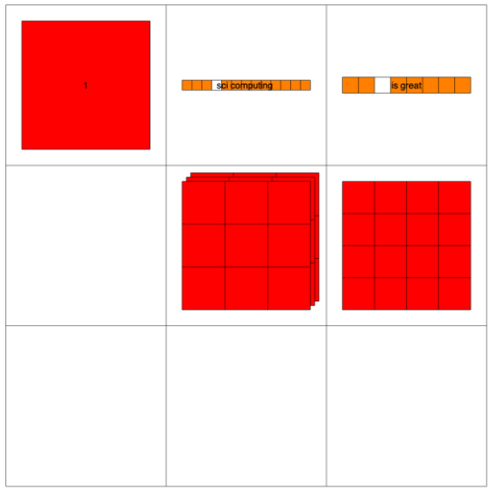
Cell Arrays

Add 3D array into a cell

y{2,3}=randi(4,4)

y{2,2}=ones(3,3,3)






```
C = {1,rand(1,10,2),'sci computing';
     "matlab",[3,4],{11; 22; 33}}
```

cellplot(C)

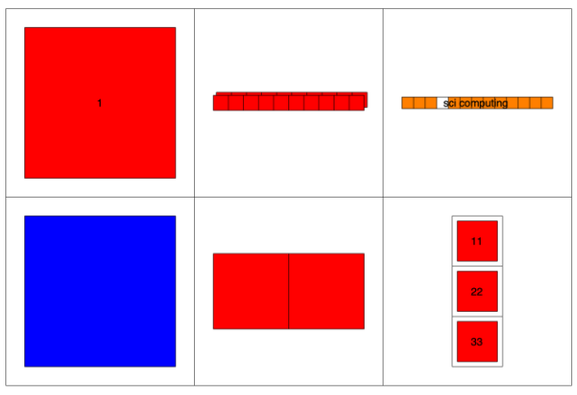


Table array

```
A = table([18;13;25],[38;43;45],...
                    'VariableNames',{'healthy' 'disease'},...
                    'RowNames',{'genex' 'geney' 'genez' })
```

```
3×2 table

          healthy    disease
          _______    _______

  genex      18         38
  geney      13         43
  genez      25         45
```

```
A = table({'chr1';'chr2';'chr3'}, [18;13;25],[38;43;45],...
    'VariableNames',{'locus','healthy' 'disease'},...
    'RowNames',{'genex' 'geney' 'genez' })
```

```
A =

  3×3 table

            locus       healthy    disease
          _________     _______    _______

    genex    {'chr1'}      18         38
    geney    {'chr2'}      13         43
    genez    {'chr3'}      25         45

fx >>
```

Table slicing

```
>> A.locus

ans =

  3×1 cell array

    {'chr1'}
    {'chr2'}
    {'chr3'}
```

```
>> A.healthy(2)

ans =

    13

>> A.genex(2)
Error using tabular/dotParenReference (line 76)
Unrecognized table variable name 'genex'.
```

```
>> A(2,:)

ans =

  1×3 table

              locus      healthy    disease
            _________    _______    _______

    geney    {'chr2'}      13         43
```

```
387
388 -  A = table([18;13;25],[38;43;45],...
389        'VariableNames',{'healthy' 'disease'},...
390        'RowNames',{'genex' 'geney' 'genez' })
391
392
393 -  A1 = table({'chr1';'chr2';'chr3'}, [18;13;25],[38;43;45],...
394        'VariableNames',{'locus','healthy' 'disease'},...
395        'RowNames',{'genex' 'geney' 'genez' })
396
397
398 -  A.healthy(2)
399 -  A.disease(1)
400 -  A(1,:)
401 -  A(2,:)
402 -  A(3,:)
403
```

Example data1

- Most microarrays data are the table format
- subdata1(1:100,:)=table2array(data1(1:100,2:34));

27130x34 table

	1 Gene	2 ARNA	3 TRNA	4 ARNA1	5 TRNA1	6 ARNA2	7 TRNA2	8 ARNA3	9 TRNA3	10 TRNA4
1	"LOC1024...	0	1.0045	4.0185	0.9162	2.9799	3.3377	1.3212	2.1511	1.0805
2	"ZBTB42"	27.8394	37.1676	55.2547	30.2348	42.7112	32.2643	54.1705	48.3991	35.6578
3	"FCAMR"	1.1136	0	1.0046	0	0.9933	0	0	0	0
4	"ZNF503-...	41.2024	35.1586	40.1853	16.4917	35.7582	32.2643	60.7767	23.6618	47.5438
5	"NFU1"	111.3578	123.5573	118.5465	133.7663	91.3821	86.7800	76.6315	120.4600	100.4903
6	"ELSPBP1"	0	0	0	0	0	0	0	0	0
7	"ZRANB3"	190.4218	183.8291	141.6531	169.4984	155.9455	92.3428	118.9109	121.5355	152.3563
8	"MECR"	259.4637	291.3139	188.8708	237.2977	289.0455	189.1358	257.6403	253.8264	347.9341
9	"LOC1057...	0	0	0	0	0	0	0	0	0
10	"LINC003...	2.2272	2.0091	3.0139	4.5810	6.9530	0	0	0	1.0805
11	"AARSD1"	1.1136	0	0	0	0	2.2251	0	0	0
12	"DEXI"	485.5200	435.9663	492.2695	308.7619	511.5410	493.9782	478.2860	357.0779	467.8742
13	"DCHS1"	1.1559e+03	1.0035e+03	1.2116e+03	1.3138e+03	1.3479e+03	1.4652e+03	1.8788e+03	1.4842e+03	1.7224e+03
14	"PSMD2"	1.2550e+03	1.8232e+03	1.3914e+03	1.4861e+03	1.5545e+03	1.3206e+03	1.5630e+03	1.1282e+03	1.4382e+03
15	"GABRR1"	3.3407	4.0181	2.0093	1.8324	5.9597	8.9005	6.6062	2.1511	2.1611

200x33 double

	1	2	3	4	5	6	7	8	9	10
1	0	1.0045	4.0185	0.9162	2.9799	3.3377	1.3212	2.1511	1.0805	0
2	27.8394	37.1676	55.2547	30.2348	42.7112	32.2643	54.1705	48.3991	35.6578	50.2696
3	1.1136	0	1.0046	0	0.9933	0	0	0	0	0
4	41.2024	35.1586	40.1853	16.4917	35.7582	32.2643	60.7767	23.6618	47.5438	40.8440
5	111.3578	123.5573	118.5465	133.7663	91.3821	86.7800	76.6315	120.4600	100.4903	68.0734
6	0	0	0	0	0	0	0	0	0	0
7	190.4218	183.8291	141.6531	169.4984	155.9455	92.3428	118.9109	121.5355	152.3563	102.6337
8	259.4637	291.3139	188.8708	237.2977	289.0455	189.1358	257.6403	253.8264	347.9341	271.2462
9	0	0	0	0	0	0	0	0	0	0
10	2.2272	2.0091	3.0139	4.5810	6.9530	0	0	0	1.0805	4.1891
11	1.1136	0	0	0	0	2.2251	0	0	0	0
12	485.5200	435.9663	492.2695	308.7619	511.5410	493.9782	478.2860	357.0779	467.8742	569.7217
13	1.1559e+03	1.0035e+03	1.2116e+03	1.3138e+03	1.3479e+03	1.4652e+03	1.8788e+03	1.4842e+03	1.7224e+03	1.2724e+03
14	1.2550e+03	1.8232e+03	1.3914e+03	1.4861e+03	1.5545e+03	1.3206e+03	1.5630e+03	1.1282e+03	1.4382e+03	1.5730e+03
15	3.3407	4.0181	2.0093	1.8324	5.9597	8.9005	6.6062	2.1511	2.1611	4.1891

How to remove missing data?

subcleandata1= rmmissing(subdata1)

200x33 double

	1	2	3	4	5	6	7	8	9	10
1	0	1.0045	4.0185	0.9162	2.9799	3.3377	1.3212	2.1511	1.0805	0
2	27.8394	37.1676	55.2547	30.2348	42.7112	32.2643	54.1705	48.3991	35.6578	50.2696
3	1.1136	0	1.0046	0	0.9933	0	0	0	0	0
4	41.2024	35.1586	40.1853	16.4917	35.7582	32.2643	60.7767	23.6618	47.5438	40.8440
5	111.3578	123.5573	118.5465	133.7663	91.3821	86.7800	76.6315	120.4600	100.4903	68.0734
6	0	0	0	0	0	0	0	0	0	0
7	190.4218	183.8291	141.6531	169.4984	155.9455	92.3428	118.9109	121.5355	152.3563	102.6337
8	259.4637	291.3139	188.8708	237.2977	289.0455	189.1358	257.6403	253.8264	347.9341	271.2462
9	0	0	0	0	0	0	0	0	0	0
10	2.2272	2.0091	3.0139	4.5810	6.9530	0	0	0	1.0805	4.1891
11	1.1136	0	0	0	0	2.2251	0	0	0	0
12	485.5200	435.9663	492.2695	308.7619	511.5410	493.9782	478.2860	357.0779	467.8742	569.7217
13	1.1559e+03	1.0035e+03	1.2116e+03	1.3138e+03	1.3479e+03	1.4652e+03	1.8788e+03	1.4842e+03	1.7224e+03	1.2724e+03
14	1.2550e+03	1.8232e+03	1.3914e+03	1.4861e+03	1.5545e+03	1.3206e+03	1.5630e+03	1.1282e+03	1.4382e+03	1.5730e+03
15	3.3407	4.0181	2.0093	1.8324	5.9597	8.9005	6.6062	2.1511	2.1611	4.1891

128x30 double

	1	2	3	4	5	6	7	8	9	10
1	27.8394	37.1676	55.2547	30.2348	42.7112	32.2643	54.1705	48.3991	35.6578	50.2696
2	41.2024	35.1586	40.1853	16.4917	35.7582	32.2643	60.7767	23.6618	47.5438	40.8440
3	111.3578	123.5573	118.5465	133.7663	91.3821	86.7800	76.6315	120.4600	100.4903	68.0734
4	190.4218	183.8291	141.6531	169.4984	155.9455	92.3428	118.9109	121.5355	152.3563	102.6337
5	259.4637	291.3139	188.8708	237.2977	289.0455	189.1358	257.6403	253.8264	347.9341	271.2462
6	485.5200	435.9663	492.2695	308.7619	511.5410	493.9782	478.2860	357.0779	467.8742	569.7217
7	1.1559e+03	1.0035e+03	1.2116e+03	1.3138e+03	1.3479e+03	1.4652e+03	1.8788e+03	1.4842e+03	1.7224e+03	1.2724e+03
8	1.2550e+03	1.8232e+03	1.3914e+03	1.4861e+03	1.5545e+03	1.3206e+03	1.5630e+03	1.1282e+03	1.4382e+03	1.5730e+03
9	780.6181	676.0491	640.9550	244.6274	522.4672	406.0857	486.2134	287.1680	504.6126	799.0765
10	309.5747	294.3275	372.7184	721.0553	238.3881	354.9078	395.0484	379.6641	298.2293	273.3407
11	113.5849	91.4123	74.3427	89.7883	100.3216	160.2091	104.3773	73.1364	63.7519	97.3973
12	1.3274e+03	1.6374e+03	1.3000e+03	1.2341e+03	1.2416e+03	1.4953e+03	1.1904e+03	1.0508e+03	1.1216e+03	1.1688e+03
13	1.3285e+03	1.1422e+03	977.5067	1.1333e+03	1.3072e+03	1.1738e+03	969.7844	795.8965	1.1454e+03	1.4641e+03
14	150.3330	145.6569	190.8800	127.3528	113.2343	140.1830	142.6931	95.7227	79.9600	89.0190
15	66.8147	36.1631	46.2131	39.3969	31.7851	52.2905	31.7096	54.8523	32.4162	34.5603

There are many different ways to remove missing data from an array
we will learn them after we learn for loops and decision control (if).
Rmmmissing is a build in function

String arrays

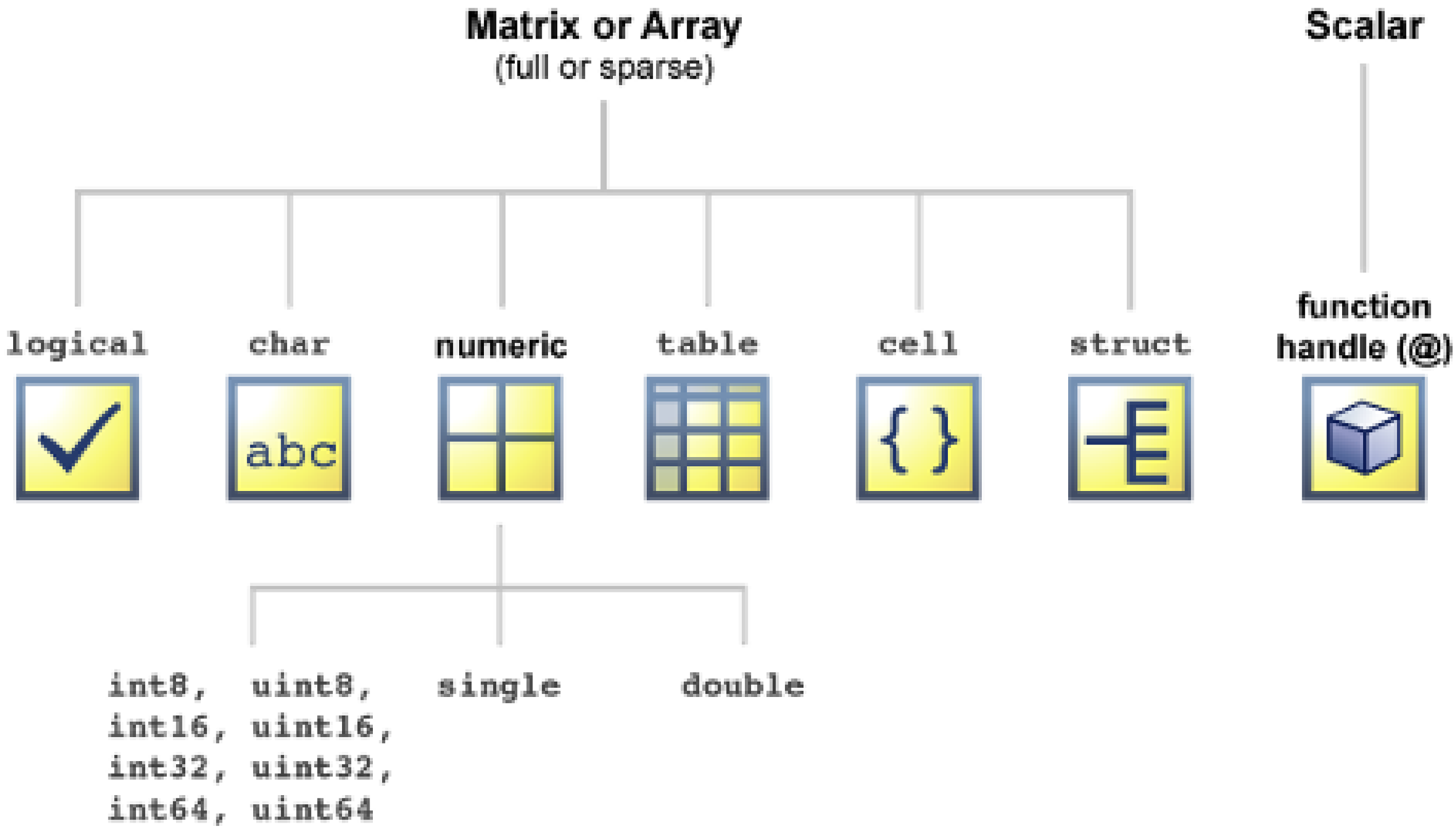
```
%% Create a char variable
% char character
geneA='AGCTCTAGTG'
geneB='AGCTCTAGTA'
geneC='AGTGTGGTGT'
geneD='AGCTCTAGTG'
```

```
%%
geneA='AAAATAGTAGATGATGATGATGTCCATATAT'
geneB='AAAATATGTAATTGTATGGATGTCCATATAT'

[row,col,v]=find(geneA~=geneB)
```

```
%%
geneD=[geneA,geneB]
x='Matlab is great'
```

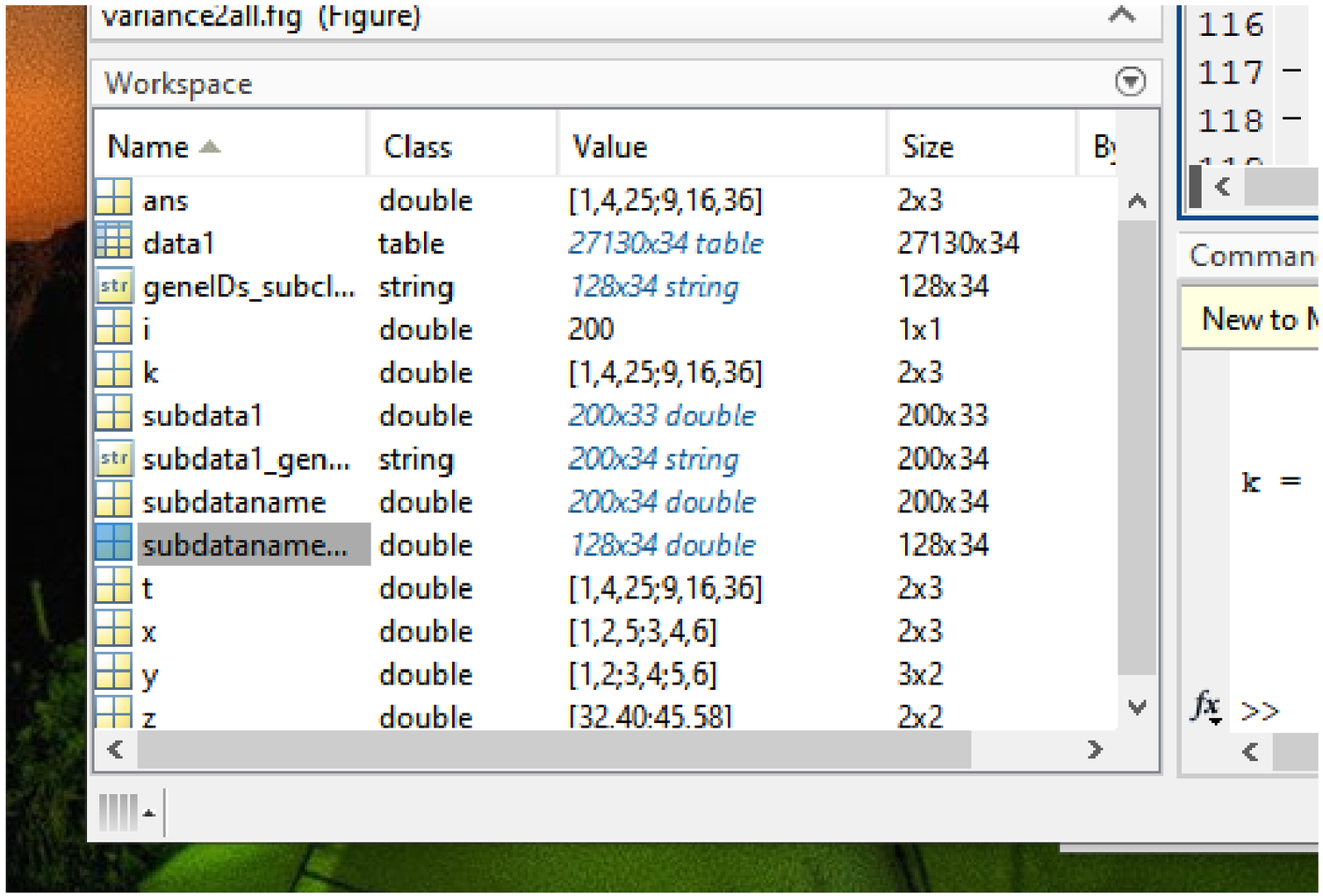
```
%% size length and find fuction for char variable

size(geneA)
length(geneA)
% compare two genes
% simple sequenc alingment
[ro,co,v]=find(geneB==geneD)
```

Array types

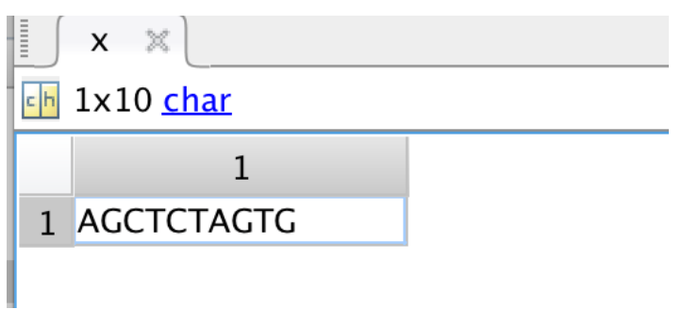

Matrix or Array
(full or sparse)
Scalar
logical
char
numeric
table
cell
struct
function handle (@)
abc
{ }
int8, uint8, single double
int16, uint16,
int32, uint32,
int64, uint64

Data Types

1.int16, int32, and int64 type of data stores numbers as integers, 1,2,3,4, -1,-2 etc.



Char: stores alphabetical characters or strings



x='AGCTCTAGTG'