Electronic Supplementary Material (ESI) for Nanoscale

Sub-picometer structural information of graphene hidden in a 50pm resolved image

Steffen B. Petersen, [1,2] Gnana Prakash Gajula, [1] Maria Teresa Neves-Petersen[1]

[1]International Iberian Nanotechnology Labaratory, Braga, Portugal
[2]Department of Health Science and Technology, Aalborg University, Denmark

Supporting Information

Repeating an experiment many times is a fundamental way of understanding uncertainties in the experimental determination of one or more parameters of a physical system. In its simplest form it is used to enhance the signal/noise in a spectrum or an image. If everything is kept constant the signal/noise will improve with sqrt(N), where N is the number of observations. It is implicit that the noise is random and has a zero mean.

In 1777 it was realized by Georges-Louis LeClerc, Comte de Buffon, that the if a needle of length ´t´ falling onto a floor consisting of parallel planks of width ´t´ (Fig.S1) the probability that the needle crosses a plank boundary is given by

$P=2/\pi$

Curiously this insight makes it possible to determine π with very simple means. One simply needs to repeat the experiment of the falling needle a large number of times, and divide the number of times the needle crosses a boundary with the total number of experiments and equal this result with $2/\pi$. The more times the experiment is repeated the more precise the experimental value for π will become.

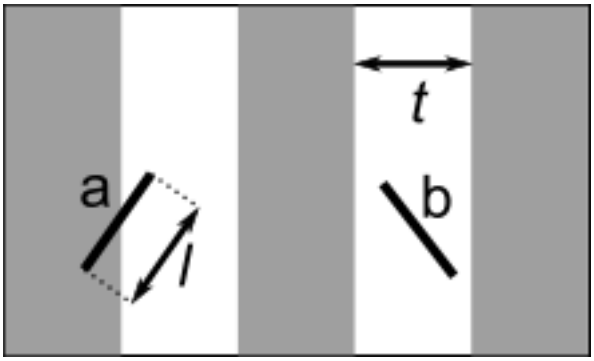


Fig. S1: Schematic diagram of ´l´ length needle falling (a,b,.... times) onto a floor consisting of parallel planks of width ´t´ (http://en.wikipedia.org/wiki/Buffon%27s_needle).

When we have simple linear objects, for which we know the precise length, a low resolution digital image will only reveal the length of the object in terms of the apparent locations of the two end points of the object on the integral grid of the pixels of the image. We can use Pythagoras equation to measure the length of the object as seen in the image. If we throw this linear objects 1000 times, take a digital image of each and deduce the length of the object in each image we will obtain a noisy data set as is shown in the Fig.S2 . If instead we plot the mean as a function of the number of throws we obtain a much more interesting result.

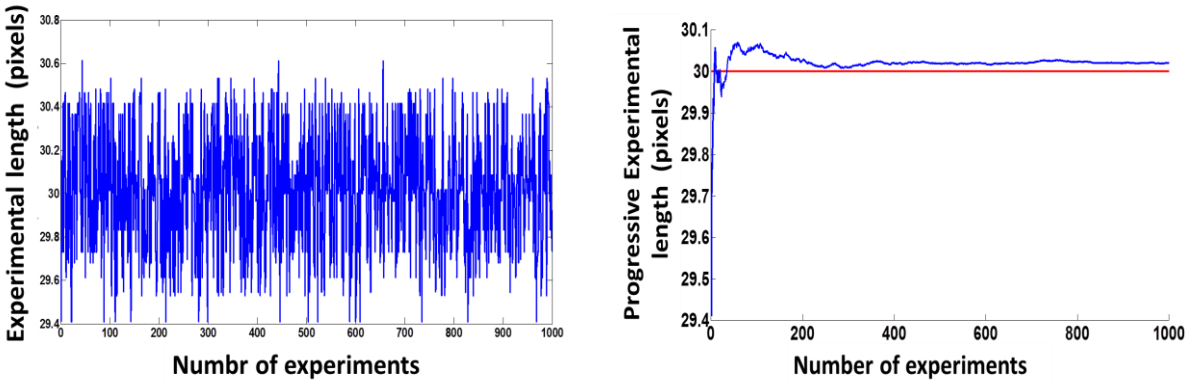


Fig. S2: (A) Noise data of a digital image object measurements for 1000 measurements. (B) graph of a mean of number of measurements indicating the good average value after 100 measurements.

The mean value after 1000 throws is within a few thousands of a pixel separated from the true value. We conclude that we can improve our ability to determine the precise size of an object if we repeat the experiment a large number of times. It is crucial that the object is mapped onto the image grid differently in every image. We cannot improve our spatial definition of the object by repeating an analysis of identical images. It is clear that the mean value is the best estimate we have of the true value.

Intuitatively it is clear that the ratio between the pixel size and the true object length is imortant in the experiment. In Fig.S3 and Fig.S4 show how the ratio between object length and pixel ratio influence the observed difference between the true length and the mean measured length after 10000 experiments. It appear that if the ratio of object length, over pixel size above 30 is optimal. Interestingly the data leading to the curve depicted in Fig.S3 behaves as a scale free data set if the log of the object length/pixel size is plotted against the log of the experimental deviation. The mean length is a very good approximation for the true length of the object if the length exceeds 30 pixels.

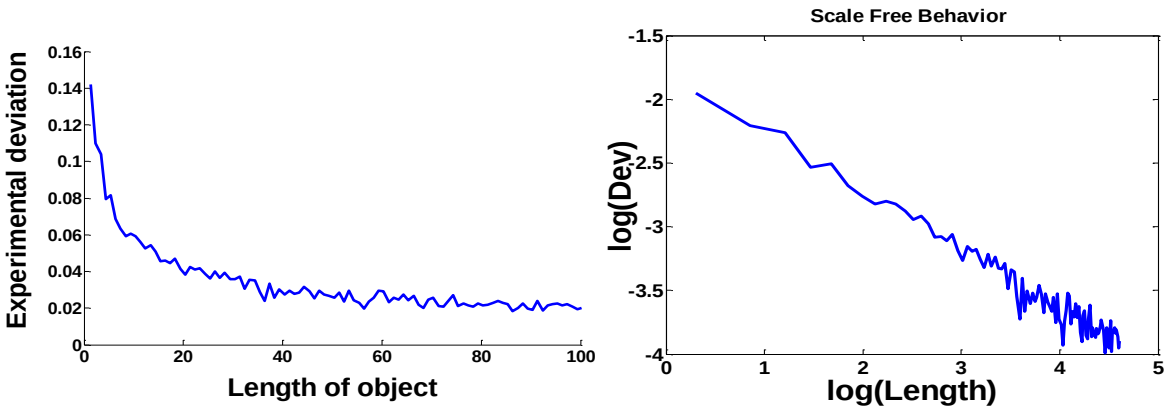


Fig. S3: (A) Plot between the length of object versus experimental deviation from true length indicates the reasonable stability after 30. (B) Log (length) Vs Log (Dev) Plot shows the Scale free behavior

In order to investigate the precision at which the mean value is determined we simulated a situation, where we threw a stick of length 30 pixels 10000 times. The histogram and standard deviation of the measurements are given in Fig.S4.

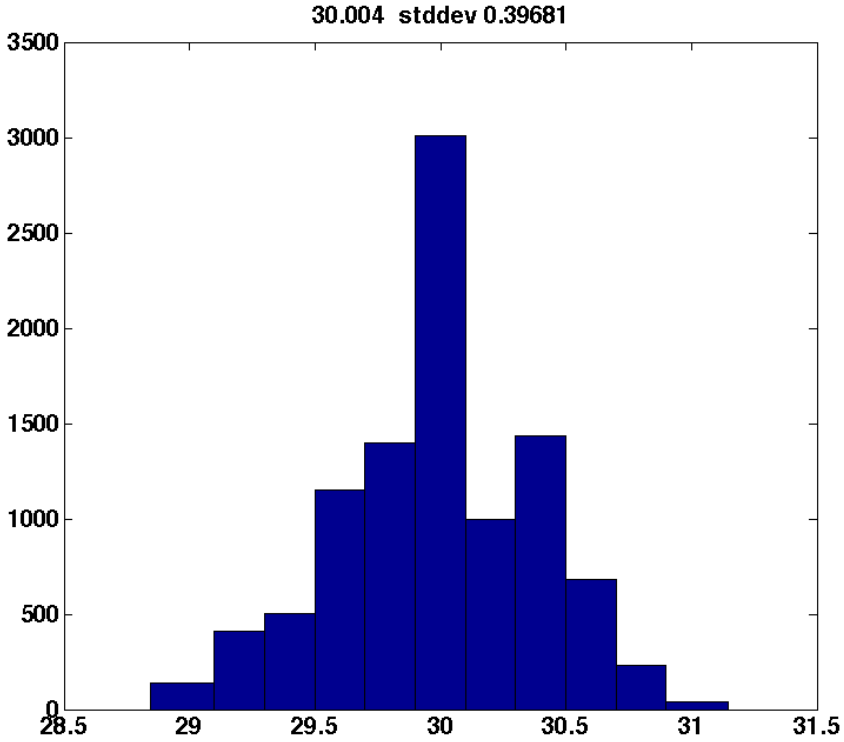


Fig.S4: Histogram of the 30 pixels object for 10000 times measurements.

The mean value 30.004 is very close to the true value, but the standard deviation is around 0.4 pixels, thus casting doubt on the true precision of the determination. We therefore designed a combined experiment where we determined the length of the 30 pixel stick in 100 throws, and repeated the experiment 10000 times. We then computed the mean of the mean value obtained in each of the 100 throws, as well as the std deviation of these values:

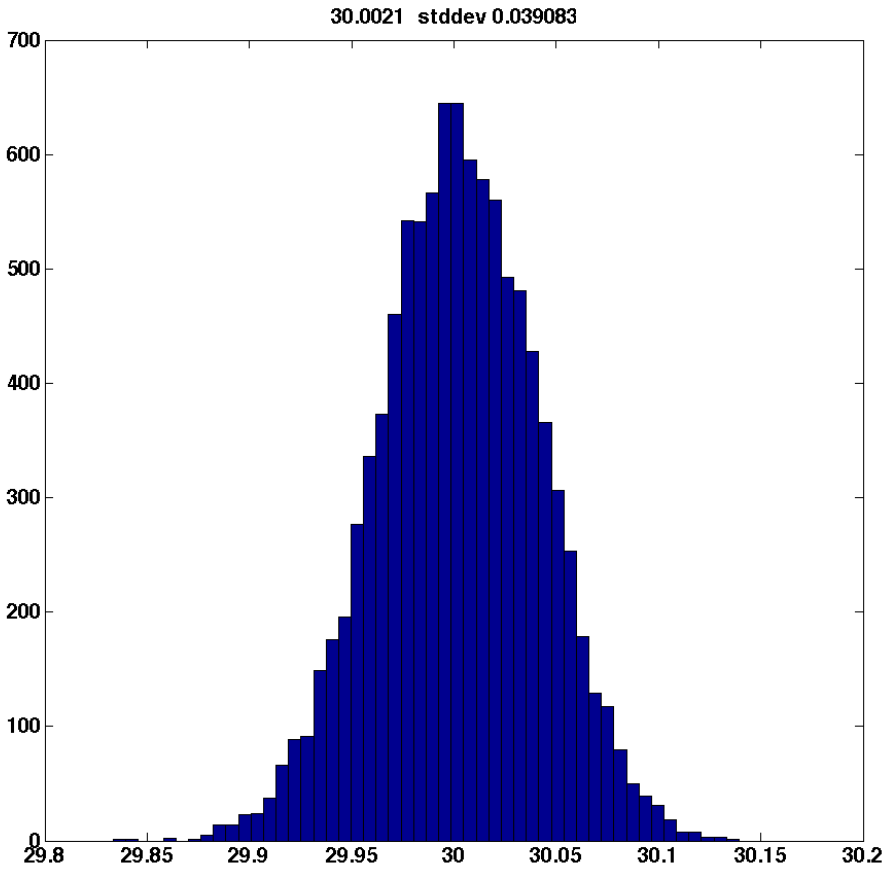


Fig.S5: Histogram of the measurement for mean value of 30 pixel length stick for 100 times and 10000 repeated values

The mean of the mean value is 30.0021 with a standard deviation of 0.04 or 0.1%, a remarkable precision, very close to the true value of 30.00000.

Flow chart of the ´BNIP Hexagonal Analysis´ program

The first step is to load the relevant image. Most standard image formats are acceptable. In some cases it is desirable to crop the image to remove unwanted features such as text, etc. After cropping the image is converted to gray scale and subsequently to binary format (black and white) using a user selected threshold. The binary image can now be manipulated with one or more morphological image processing functions. The most important one is "clear boundary". This command removes all objects that are in direct contact with the image boundary thus it prevents the subsequent analysis to be polluted with partial information. In some cases single pixel regions can be removed with the command "Erosion". Finally, the image is analyzed with respect to the number and location (centroids) of particles. The above mentioned steps are shown graphically below.

The output from the image pre-processing program is the gray-scale image, the binary image and the centroids for all particles found. Since the centroids are known we can query the data with respect to which and how many neighbors each particle has. We are only interested in investigating the first sphere neighbors. The "Hex Analyze" program asks the user to define a minimum distance as well as a maximum distance within which the user wants to investigate the data. Typically the user will request all neighbors up to a certain level and investigating the resulting histogram he can identify the proper distance threshold for the nearest neighbors. For each particle the program counts the number of nearest neighbors. Clearly, in the present context we are interested in all cases where the number is 6, i.e., hexagonal packing. The program now performs a translation using the centroids information for the

selected objects. The program now displays a superposition of all hexagonal environments in a Cartesian plot. This plot is transformed to a polar plot where radial and angular information are clearly separated. The user now selects a reference hexagon. This may either be one selected from the reference dataset or one modeled hexagon. At this stage the program will attempt to optimize the superposition of the objects onto the reference structure by rotating the object until a minimal deviation is found. The program will now report the Cartesian superposition, the polar plot and upon user request the Voronoi plot as well as the actual structural details of the individual consensus hexagon. Finally, the null hypothesis matrix is provided which contains information about the statistical dependencies of the observations. The above mentioned steps are shown graphically below.

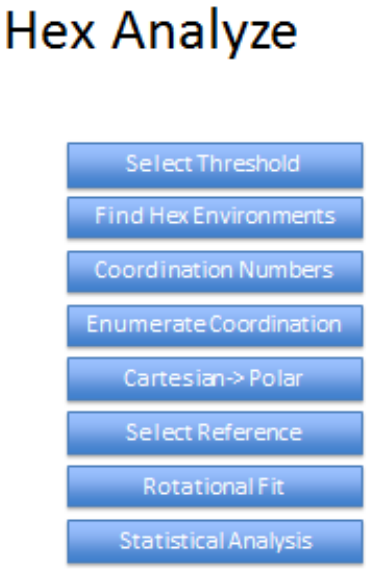


TABLE S1: Values for experimental graphene

	Theta	Rho	Av. - Rho_x	pm
1	-142.7272 (±1.4505)	10.5270 (± 0.2672)	0.0315	0.73
2	-82.0386 (±1.4409)	10.5160 (± 0.2754)	0.0425	0.98
3	-22.3431(±1.4951)	10.6316 (± 0.2602)	0.0302	0.703
4	37.2558 (± 1.4558)	10.5283 (± 0.2685)	0.0412	0.96
5	97.9609 (± 1.4431)	10.5173 (± 0.2744)	0.0725	1.69
6	157.6610 (± 1.4985)	10.6310 (± 0.2584)	0.0731	1.70
Av.		10.5585		

SET of Objects (1, 2,4 & 5)

1. 10.5270 – 10.52215 = 0.00485
3. 10.5160 – 10.52215 = 0.00615
2. 10.5283 – 10.52215 = 0.00615
4. 10.5173 – 10.52215 = 0.00485

5. 10.6310 – 10.6313 = 0.0003
6. 10.6316 – 10.6313 = 0.0003

SET of Objects (1&4, 2&5, 3&6)

5. 10.6310 – 10.6313 = 0.0003
6. 10.6316 – 10.6313 = 0.0003

1. 10.5270 – 10.52765 = 0.00065
2. 10.5283 – 10.52765 = 0.00065

3. 10.5160 – 10.51665 = 0.00065
4. 10.5173 – 10.51665 = 0.00065

Creating an ideal hexagonal lattice

Now we create an ideal lattice of hexagons consisting of a total of 1975 'objects'. In the example investigated we assumed a bond length of 10.5585 pixels, i.e. a non-integral length which is suboptimal according toFig.S3. The centroid was computed for each object. A total of 1777 hexagons were identified by an automated procedure based on morphological image processing.

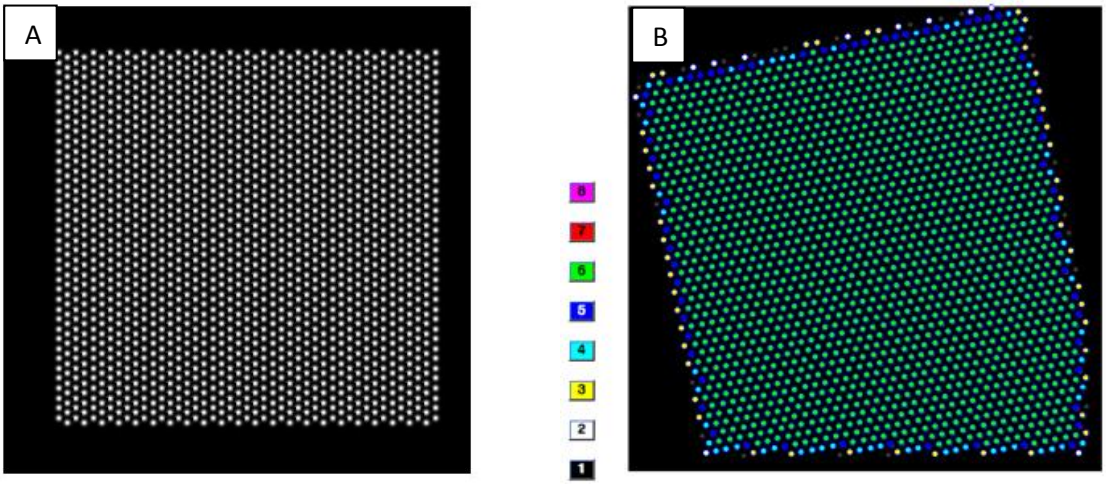


Fig.S6: (A) Ideal digital image with a perfect hexagonal environment. (B) Co-ordination plot of the generate perfect hexagonal structure rotated 11^0. The number of nearest neighbors is the co-ordination number for each individual graphene core object. The color box also present the co-ordination number for each graphene core.

Since the ideal hexagonal lattice was digitized, some side-effects emerge. Please note the 2 objects with 2 red dots and the 4 objects with 4 dots (Fig. S7). Each red dot denotes the location of a object in at least one hexagon. The blue cross embedded in a circle denotes the weighted mean position for all objects at this location. The fact that we see multiple dots for each object is an effect of the digitization. It is now relevant to address which bond lengths can be extract from the 'ideal' lattice.

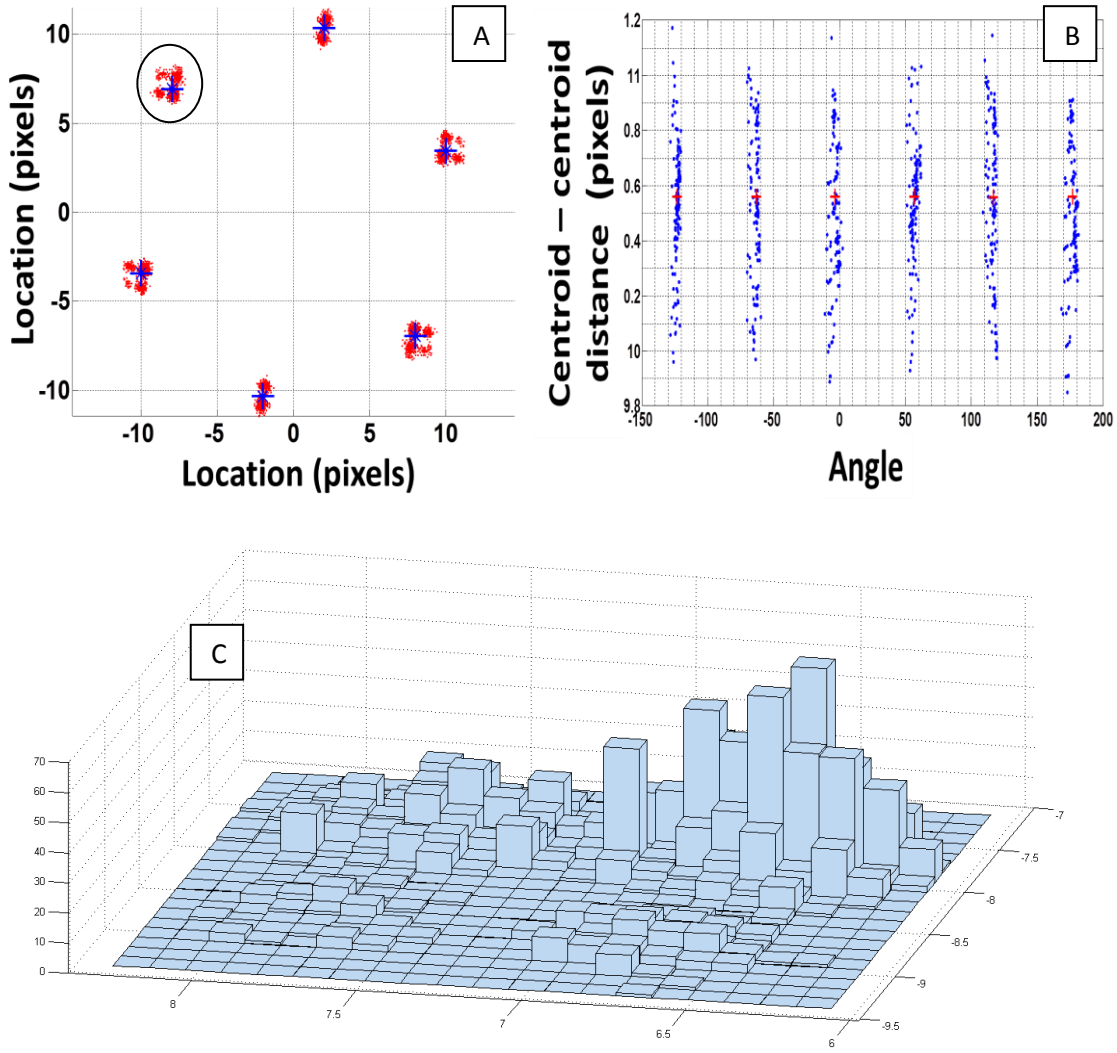


Fig.S7: Cartesian Plot (A) and polar plot (B) for the ideal hexagonal digital image after translational fit. (C) The 3D histogram of the distribution mentioned in (A) as circle.

The values listed below table were the measured bond lengths for each of the 6 positions in the three lattices for perfect hexagonal structure created. Values given in round parenthesis are the calculated standard deviations on the number preceding the parenthesis.

TABLE S2: *Table of bond lengths in model hexagonal lattices.*

	Angle	Centroid to centroid distance (please see Fig.3B for comparison)
1	-131.2009 (± 1.3342)	10.5607 (± 0.2091)
2	-71.2260 (± 2.4931)	10.5684 (± 0.1788)
3	-11.2177 (± 1.2447)	10.5613 (± 0.4723)
4	48.7961 (± 1.3111)	10.5610 (± 0.2077)
5	108.7682 (± 2.4974)	10.5687 (± 0.1778)
6	168.7855 (± 1.2489)	10.5610 (± 0.4724)

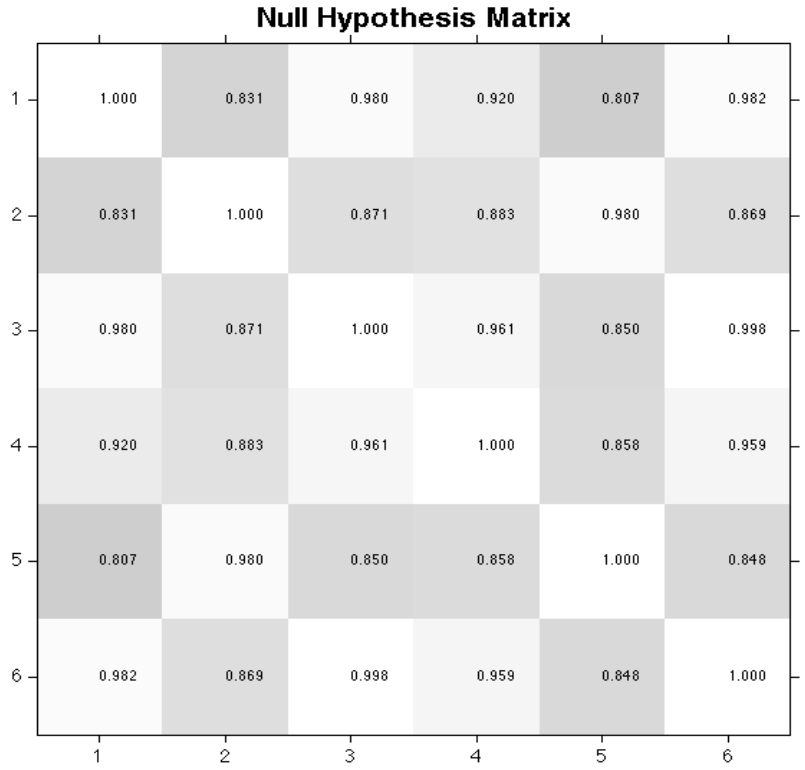
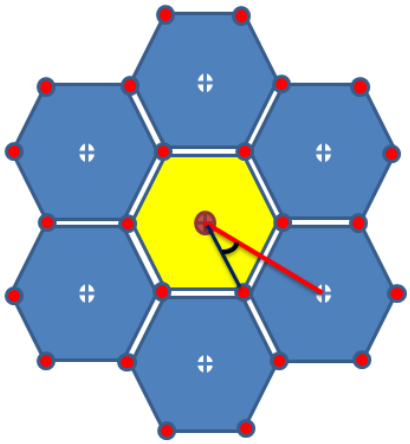


Fig.S8: Null Hypothesis Matrix combinations of the 6 distributios shown in Fig:S7B. In this case, none of the 6 distributions are separable according to the null-hypothesis matrix.

Calculating the pixel size:



- The mean value of centroid to centroid distance (´*Rho*´ shown as a red line) is 10.5585 pixels (average value obtained from analysis).
- The distance between the C-C bond in graphene (´L´ shown as a white line) is 1.42 Å.
- Angle between the black line and centroid-centroid red line is 30°.
- In order to calculate the pixel value, the C-C bond length has to be related to the known centroid-centroid distance in pixel.
- Therefore, $\tan 30^{o} = \frac{L/2}{Rho/2}$
- i.e. $\tan 30^{o} = \frac{L}{Rho}$
- i.e., $1/\sqrt{3} = \frac{L}{Rho}$
- i.e. $Rho = L * \sqrt{3}$

Since, L = 1.42 Å and *Rho = 10.5585 pixels.*

- 10.5585 pixels = $\sqrt{3}$ X 1.42 Å
- Therefore, 1pixel = 0.2329 Å = 23.29 pm.

Fig.S9: Calculating the pixel size – detailed information on how the pixel size in pm has been calculated.