ND is an image processing plugin that can be used to calculate the average size and distance between particles and their closest neighbors in many-particle systems. It is written in Java and implemented in ImageJ, the open source Java-based software developed by National Institute of Health (NIH). ND is particularly useful in analysis of porous synthetic and natural constructs (known as scaffolds) that are commonly used in the field of regenerative medicine and tissue engineering. Architecture of these scaffolds including pore size and density significantly affects the behavior and fate of cells cultured on them. ND adds to the built-in functionalities of ImageJ to provide a fast and user-friendly method to better characterize the porosity of these scaffolds.

Keywords: ImageJ; porosity; scaffold; nearest neighbor distance
systems, coordination number depends on the visual perception and can be lower or higher. Estimation of particle spacing of a particle with its neighboring particles is performed as follows:

1. The centroid coordinates of each particle \((X,Y)\) is derived from the result table of the built-in Analyse Particles plugin.
2. A circle is fit on each particle with the center \((X,Y)\) and radius \(r\).
3. The spacing (wall thickness) between a pair of particles \((d)\) is calculated as:
   \[
   d = \sqrt{(Y_2 - Y_1)^2 + (X_2 - X_1)^2} - (r_1 + r_2)
   \]
4. The distances of each particle with all the other particles is stored in an array and sorted.
5. Depending on the coordination number of interest, average of the distances is calculated.
6. Results are shown in a new result table, which contain the distance of the closest neighbor to each particle, the average wall thickness, and the average distance of the closest neighbors.

**Quality Control**

Functional testing, load testing and end-to-end testing were carried out in Microsoft Windows 7-64 bit. The ND Plugin functions as expected and results were manually verified. The level of accuracy/precision of the software when compared to manual verification is solely dependent on the threshold settings and the parameters set in Analyze Particle built-in plugin before running the ND plugin. Therefore, the limiting factor in the accuracy/precision data generated in the results table. Therefore, with regard to values available in the results table, the ND plugin compared with manual calculation is completely accurate. One scenario that may lead to an inaccurate estimation of particle/pore distances might occur when particles/pores are too close to each other. In this situation a slight change in the values used for thresholding the image might lead to merging of two neighboring particles/pores.

**Example of Use**

*Measurement of tubule/fiber diameter and their spacing in tubular scaffolds or fiber reinforced composites*

The main usage of the ND plugin is to analyze the size and spacing of particles/pores/tubules located randomly in a sample. While the size of particles can be estimated using the built-in “Analyze Particle” plugin in ImageJ, the spacing between the features cannot be estimated. Fig. 1 shows an example of a tubular scaffold used in tissue engineering, which was analyzed using ND plugin. ND provides this ability for the user to calculate the spacing between each particle/pore and its nearest neighbors with \(n\) being the coordination number of interest (Fig. 2). Additionally, the histogram of the average distances of the closest neighbors provides a quantitative measure of how uniformly the feature of interest is spatially distributed. Initially, preliminary steps need to be taken for the plugin to do its function. These steps are as follows:

- a) Open the image (File -> Open)
- b) Threshold the image to make the feature of interest evident (Image -> Adjust -> Threshold)
- c) Run the built-in Analyze Particles (Analyze -> Analyze Particles)
- d) Run the ND plugin (Plugins -> ND)
- e) The result table which outputs the average distance between tubules, the average wall thickness, and the distance of the closest neighbor for each tubule is shown in Fig. 2.
(2) Availability

**Operating system**
ImageJ runs and is used on different versions of Unix, Mac OS X, and Windows.

**Programming Language**
Java

**Additional system requirements**
None at runtime

**Dependencies**
At runtime, the plugin requires ImageJ to be running

**List of contributors**
Morteza Haeri

**Archive**

**Code Repository Name**
Github

**Identifier**
https://github.com/sedmorteza/ND.git

**Publisher**
Morteza Haeri

**License**
GNU General Public License version 3

**Date published**
2013–09-24

(3) Reuse Potential

**Measurement of pore size distribution and fiber thickness in electro-spun fiber mats**
Alternatively, in electro-spun fiber mats, ND can be used to estimate the average pore size as well as the average fiber size and distribution. In an electro-spun fiber mat, nano or micro sized fibers are stacked to form a mesh structure. The structure is highly porous with varying pore sizes. By using the procedure mentioned above to threshold images from electro-spun fiber mats and running the built-in Analyze Particle plugin, one can generate a result table containing the radii of pores. Running the
ND plugin subsequently will estimate the wall thickness between pores, which in this case will be the thickness of fibers forming a pore.

Detection of co-localization
A common measure used in biology is object-based co-localization (OBC). In OBC, the nearest neighbor distances (NND) from one point pattern to the other is plotted as a histogram and then a threshold distance is set (typically, the ‘diffraction-limited resolution’ of ~200 nm), so that all points with a NND less than the threshold is considered co-localized. ND plugin can be modified to provide this functionality. While the current code calculates the distances of all the neighboring particles from each particle, the ND plugin only uses the nearest neighbors up to maximum of 6, which corresponds to a close packed configuration into the subsequent averaging and display windows. However, the data required for OBC is stored in the matrix named “dist” in the code and can be used accordingly for other applications such as OBC.

References