

SOFTWARE METAPAPER

ImageJ Plugin for Analysis of Porous Scaffolds used in Tissue Engineering

Morteza Haeri¹ and Mohammad Haeri²

¹ Center for Biomaterials, University of Connecticut Health Center, Farmington, CT, USA

² Pathology & Immunology and Molecular & Human Genetics, Baylor College of Medicine, Houston, Texas, USA

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

(1) Overview

Introduction

ND, which stands for nearest distances, is an ImageJ plugin that was developed to calculate the average size and distance between pores and their nearest neighbors in a porous scaffold by using images of the scaffold as input. The architecture of the porous scaffolds including pore size distribution and density dramatically affects their interaction with biological cells and mechanical properties of the scaffold [1–5]. Therefore, characterization of fabricated scaffolds using microscopy techniques and subsequent processing of the images is an essential step in correlating a specific cell-scaffold response with an effective scaffold design. Although ImageJ is taking center stage as a reliable open source and user-friendly tool for image processing in the biomedical field, image analysis is done either manually or by using propriety software associated with microscope companies or other commercial software whenever specialized plugins for ImageJ are not available. ND is an example of a specialized plugin that builds upon the built-in functionalities of ImageJ, and contributes to the collective efforts to enrich its library of available plugins and advances its versatility in image processing.

The main usage of the ND plugin is to analyze the wall thickness and average spacing of particles/pores/tubules located randomly in a sample. The plugin uses an image of the sample as input and outputs a result table listing the average distance of each particle from its neighbors, nearest neighbor distance, and the average wall thickness of its neighbors. The number of neighbors

to be accounted for in calculating the averages is set by the user.

Implementation/Architecture

The plugin was written in Java, using the ImageJ built-in Java compiler.

Installation

To install the plugin, ImageJ must be installed first. ImageJ can be downloaded from <http://rsbweb.nih.gov/ij>. Variants like Fiji (<http://fiji.sc/>) may also be used. ND_.java file should be downloaded to the plugins folder of the ImageJ installation. The ND_.java file should be compiled for the first time using Plugins -> Compile and Run On restarting ImageJ, the plugin will appear under the menu Plugins. Alternatively, the already compiled file (ND_.class) can be downloaded and placed in the Plugin folder of ImageJ installation. Prior to running the ND plugin, an image should be opened first and the Results table must have records including centroid coordinates of the particles that need to be analyzed. (Centroid and Fit ellipse must be checked under Analyze -> Set Measurements). The user may run Analyze -> Analyze Particles built-in Plugin to generate a Results table needed by the ND Plugin. Messages will be displayed in case any of the above conditions are not met and users will be directed on how to proceed.

Algorithm

In a close packed configuration of particles/fibers having a circular cross section in 2D space there are 6 immediate neighbors surrounding each particle. In randomly packed

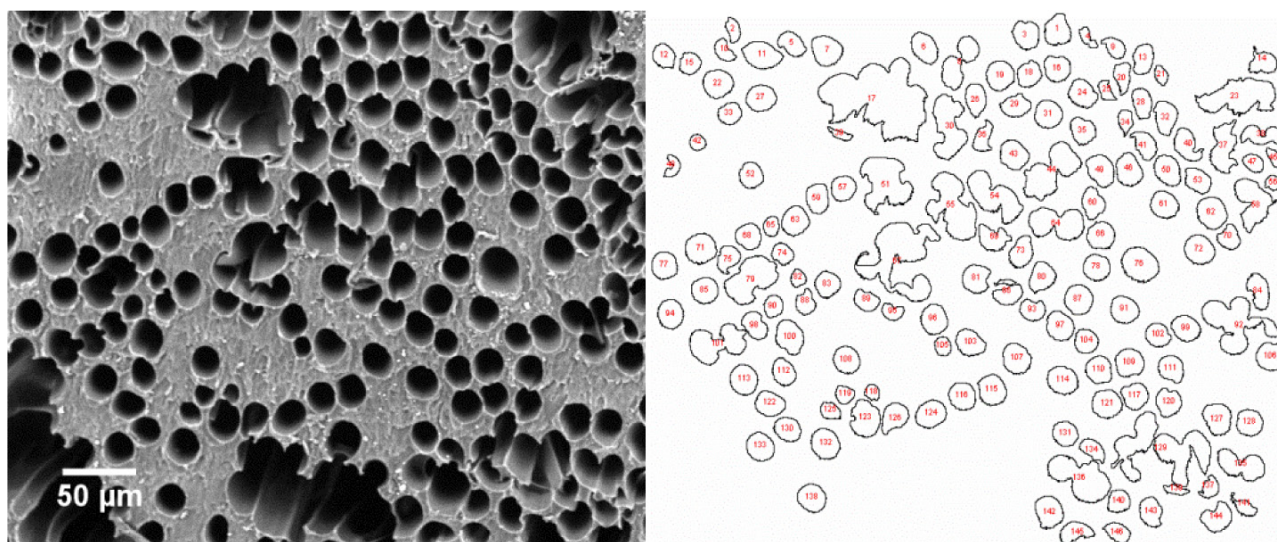
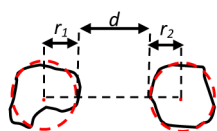


Fig. 1: Scanning electron micrograph of a tubular scaffold used in tissue engineering (left) and its segmentation after using built-in Analyze Particles ImageJ plugin (right).

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 Analyze 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 of the ND plugin is 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 n closest 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 distance for each tubule is shown in **Fig. 2**.

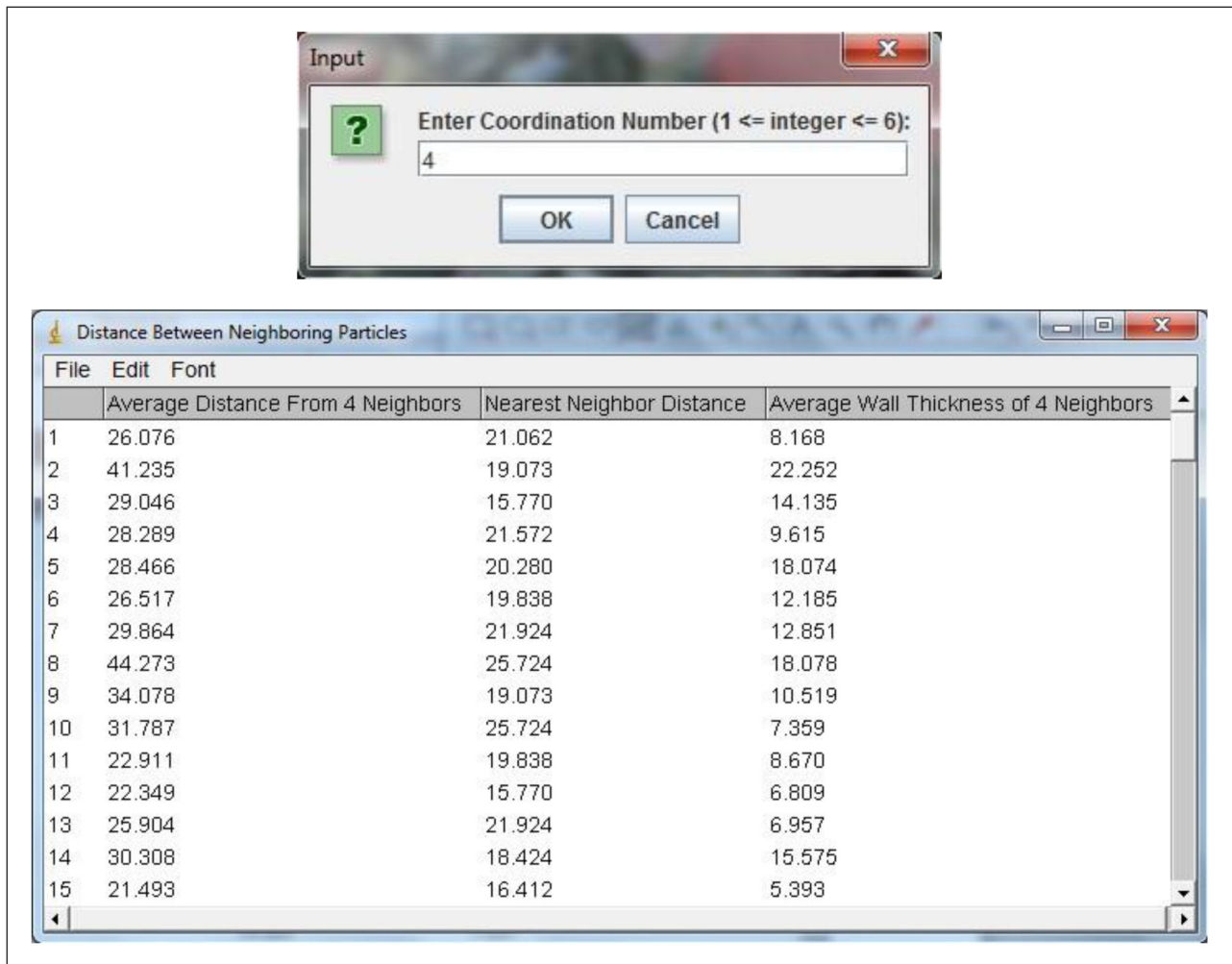


Fig. 2: ND plugin running. Input window to take the coordination number of interest to the user (top) and the generated result table listing the average distance of each particle from its neighbors, nearest neighbor distance, and the average wall thickness of its neighbors (bottom).

(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

Persistent Identifier

https://github.com/sedmorteza/ND/blob/master/ND_.java

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

1. **Scott JB, Afshari M, Kotek R, Saul JM** 2011 The promotion of axon extension in vitro using polymer-templated fibrin scaffolds. *Biomaterials*, **32**(21): 4830–4839. DOI: <http://dx.doi.org/10.1016/j.biomaterials.2011.03.037>.
2. **Martínez-Ramos C, Vallés-Lluch A, Verdugo JM, Ribelles JL, Barcia Albacar JA, Orts AB, Soria López JM, Pradas MM** 2012 Channeled scaffolds implanted in adult rat brain. *Journal of Biomedical Materials Research A*, **100**(12): 3276–3286. DOI: <http://dx.doi.org/10.1002/jbm.a.34273>.
3. **Oh S, Brammer KS, Li YS, Teng D, Engler AJ, Chien S, Jin S** 2009 Stem cell fate dictated solely by altered nanotube dimension. *Proceedings of National Academy of Sciences*, **106**(7): 2130–2135. DOI: <http://dx.doi.org/10.1073/pnas.0813200106>.
4. **Karageorgiou V, Kaplan D** 2005 Porosity of 3D biomaterial scaffolds and osteogenesis. *Biomaterials*, **26**(27): 5474–91. DOI: <http://dx.doi.org/10.1016/j.biomaterials.2005.02.002>.
5. **Peng-Yuan W, Lauren RC, Helmut T, Wei-Bor T, Nicolas HV** 2013 High-throughput characterisation of osteogenic differentiation of human mesenchymal stem cells using pore size gradients on porous alumina. *Biomaterials Science*, **1**(9): 924–932. DOI: <http://dx.doi.org/10.1039/C3BM60026B>

How to cite this article: Haeri, M and Haeri, M 2015 ImageJ Plugin for Analysis of Porous Scaffolds used in Tissue Engineering. *Journal of Open Research Software*, 3: e1, DOI: <http://dx.doi.org/10.5334/jors.bn>

Published: 28 January 2015

Copyright: © 2015 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License (CC-BY 3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/3.0/>.