UNIVERSITY OF PALERMO

FACULTY OF ENGINEERING

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE

STRUCTURAL DETERMINISTIC MODELING DESIGN AND FABRICATION OF ELECTROSPUN SCAFFOLDS FOR SOFT TISSUE ENGINEERING

Ph.D Thesis of:

Antonio D'AMORE

Supervisors:

Prof. Giovanni PETRUCCI

Prof. Michael S. SACKS

Prof. William R. WAGNER

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY XXII PhD PROGRAM CYCLE
ACADEMIC YEARS 2008 - 2010

1ST DRAFT

NOVEMBER 15TH 2010

TABLE OF CONTENTS

Acknowledgments……pag 6

Abstract……pag 7

1. Chapter 1: **On the mechanical function of scaffolds for soft tissue engineering**
   1.1. Scaffolds for soft tissue engineering: overview……..……pag 9
   1.2. Native extra cellular matrix isolation for therapeutic applications……pag 12
   1.3. Synthetic extra cellular matrix and processing methods for synthetic scaffolds……..pag 17
      1.3.1. Non fibrous 3D synthetic scaffolds……..pag 19
      1.3.2. Fibrous synthetic scaffolds……..…pag 23
         1.3.2.1. Self assembly ……..pag 24
         1.3.2.2. Hydrogels……..pag 27
         1.3.2.3. Needled non – wovens ……pag 29
         1.3.2.4. Electrospinning…….……pag 31
   1.4. Engineered tissues as models systems……pag 33
   1.5. Emulating native tissue mechanical behavior ……...pag 43

1.6 Mechanical modeling ……pag 49

1.7 Aims of this work……pag 54

2. Chapter 2: **Characterization of the complete fiber network topology of planar fibrous tissues and scaffolds**……pag 57
   2.1. The state of the art…….…. pag 57
   2.2. A novel methodology, the algorithm description……pag 57
      2.2.1. Material fabrication and imaging……pag 59
      2.2.2. Image pre-processing ……..…pag 60
      2.2.3. Fiber network detection……. pag 61
      2.2.4. Fiber diameter detection……..pag 64

2.2.5. Extraction of architectural descriptors and validation…… pag 67
2.3. Image analysis results……..… pag 69
2.3.1. Application for ES-PEUU scaffolds…….. pag 74
2.3.2. Application to collagen gel and decellularized tissues…….……pag 76
2.4. The presented image analysis technique: advantages and limitations……pag 78
3. Chapter 3: **Elastomeric electrospun polyurethane scaffolds: the interrelationship between fabrication conditions, fiber morphology, and mechanical properties**…….pag 84
3.1. The state of the art …….…….pag 84
3.2. Material fabrication……..……...pag 87
3.3. Material characterization…….…pag 89
3.4. Fabrication study…….……....pag 90
3.4.1. Effect of rastering on dry ES-PEUU…….. …….pag 90
3.4.2. Effect of concurrent medium spraying…….…pag 93
3.4.3. Effect of cellular and particulate integration……....pag 95
3.5. Interrelationship between fabrication conditions, fiber morphology, and mechanical properties: results summary…….....pag 98
3.5.1. Rastering ……..……pag 99
3.5.2. Concurrent medium spraying…….……pag 100
3.5.3. Cellular and particulate integration……....pag 101
3.6. Implications for future studies……...pag 102
4. Chapter 4: **Structural deterministic mechanical models for electrospun scaffolds**……pag 104
4.1. Biomechanical modeling of soft tissue engineering scaffolds…….pag 104
4.2. Biomechanical model of ES-PEUU: Method……pag 110
4.2.1. Experimentally derived mesh generation……...…pag 111
4.2.2. Finite element mechanical modeling……..…pag 122
4.3. Biomechanical model of ES-PEUU: Results……pag 130
4.3.1. Macro scale……pag 130
4.3.2. Meso scale……..pag 133

4.3.3. Micro scale…….pag 139
4.4. Biomechanical model of ES-PEUU: Conclusion……..pag 142
5. Chapter 5: **ES-PEUU for tissue engineering heart valves: a feasibility study on an ovine model**……...…….pag 144
5.1. Material selection……..…pag 144
5.2. In vivo animal model and surgery…….pag 146
5.3. Explants analysis……...……pag 149
6. Chapter 6: **Conclusion and future works**……...……pag 153
6.1. Using the developed structural deterministic model to assist material design and fabrication for soft tissue engineering applications……...pag 153
6.2. Future works……..pag 155
6.3. Conclusion……pag 157
7. Appendix……...pag 160
7.1. Appendix A): Image analysis code (Matlab)…… pag 160
7.2. Appendix B): Mesh generator code (Matlab)……pag 211
7.3. Appendix C): Real – Simulated Scaffold Geometry Optimization code (Matlab)……...pag 257
7.4. Appendix D): FEM analysis code (Ansys)……pag 282
8. Bibliography……pag 291

ACKNOWLEDGMENTS

I would like to thank my supervisors, Prof Giovanni Petrucci, Prof Michael S. Sacks and Prof William R. Wagner for the support and the precious guide during these three years of doctoral studies.

I would like to acknowledge our founding sources: the NIH (grant R01 HL-068816) and the Italian Ministry for University and Research (MIUR). My gratitude goes also to the Bioengineering Department and the McGowan Institute for Regenerative Medicine, University of Pittsburgh for providing an ideal environment to formulate and develop novel research ideas. The Cardiovascular Biomechanics Laboratory and Cardiovascular Engineering laboratory members have all played a fundamental role over these two past years, in particular I would like to mention my co-authors: John Stella, Nicholas Amoroso, Dr Yi Hong, and Christopher Hobson.

The Dr John E. Mayer, Jr. team at the Children Hospital, Harvard Medical School offered its invaluable insight and expertise operating the implant and explants of the Tissue Engineering Heart Valve. Dr Ahmed Bayoumi was the resident performing the surgeries. Dr. Robert F Padera and Frederick J. Schoen teams at the Department of Pathology, Harvard Medical School performed the histologies.

I would like to conclude this section thanking my treasured family and close friends for their unconditioned love and for sharing and sailing at my side under different conditions: Anna and Giulio D'Amore, Maria Oddo, Daniele Scorsone, Giuseppe Scaglione.

Finally, this Ph.D dissertation is dedicated to paediatric patients affected by cardiovascular diseases and their families.

ABSTRACT

The research fields of tissue engineering, biomechanics and regenerative medicine continue to evolve in response to the ever growing need for tissue replacement options. These fields aim to restore, maintain, or improve tissue or whole organ function.

This doctoral studies focus on the development and experimental validation of a structural deterministic modeling strategy to: A) guide tissue engineering scaffold design, B) provide a better understanding of cellular mechanical and metabolic response to local micro-structural deformations. Targeted clinical application was the pulmonary heart valve. Electrospinning was selected as the optimal platform technology to implement, validate and test the presented designing strategy.

An innovative custom made software was developed and tested on Electrospun poly (ester urethane) urea scaffolds (ES-PEUU), decellularized native tissues and collagen gels to fully characterized engineered constructs morphology. These structural information were adopted to feed and assist the mechanical modeling

Two previously unevaluated fabrication modalities were investigated throughout both mechanical testing and image analysis in order to explore further how the electrospinning fabrication process can alter the structure and mechanical response: variation of mandrel translation velocity and concurrent electrospraying of cell culture medium with or without cells or rigid particulates. These fabrication parameters were studied to enrich control in the electrospinning process.

The detected material topology and mechanical equi-biaxial data were adopted to generate statistically equivalent scaffold mechanical models. The structural determinist approach was applied to ES-PEUU scaffolds, validated and mechanical response at organ and cell level was produced through FEM simulation. Prediction included: membrane tension vs. stretch relation, elasticity moduli, Nuclear Aspect Ratio vs. stretch relation for the cells micro-integrated into the scaffold.

A three weeks in vivo - study on an ovine model was performed to demonstrated the feasibility of the adoption of ES-PEUU for TEHVs and more generally this material potential for soft tissue regeneration. Explants analysis showed surgical feasibility and acceptable valve functionality.

The developed design strategies combining image analysis and structural deterministic modeling enabled the material topology to be both quantified and reproduced. Material fabrication parameters were related to material micro-architecture Similarly, the micro-architecture was related to macro scale mechanical responses such as in-plane reactions or flexural rigidity, and complex meso – micro scales mechanisms such NAR – stretch relation. In conclusion, the modeling approach introduced in this work bridges for the first time the scaffold fabrication parameters with the mechanical response at different scale length. The developed paradigm will be utilized to identify the optimal scaffold for a given soft tissue engineering application.

CHAPTER 1

ON THE MECHANICAL FUNCTION OF SCAFFOLDS FOR SOFT TISSUE ENGINEERING

1.1 - SCAFFOLDS FOR SOFT TISSUE ENGINEERING: OVERVIEW

The research fields of tissue engineering and regenerative medicine continue to mature in response to the ever growing need for tissue replacement options. These fields aim to restore, maintain, or improve tissue or whole organ function. It is estimated that $400 billion is spent each year in the United States on patients suffering from organ failure or tissue loss [1]. This general lack of replacement options comes at the further expense of approximately 4,000 human lives who are awaiting organ transplants and an additional 100,000 patients who expire without qualifying for the implant waiting lists each year. To meet this need, a multidisciplinary approach combining biology, medicine, and engineering is required to overcome the significant challenges preventing the successful repair or replacement of biomechanically functioning tissues. Many of the reasons for poor implant performance or failure in academic or clinical settings remain ill defined. Often they are a combination of inadequate or miss-matched mechanical properties and biological complexities. As such, a great deal of effort focuses on gaining a deeper insight into the structural and behavioral characteristics of native tissues to guide design criteria for the development of tissue surrogates.

Despite the multitude of challenges, many early approaches and technologies have shown promising results. For example, valvular implants sourced from bovine pericardium or porcine valve leaflets have long been used to enhance survival and improve the quality of life of patients presenting with a variety of valvular maladies. Similarly, engineered dermal grafts have successfully been used clinically to treat severe burns or wounds that would otherwise be unable to close and heal properly. Recent work by Badylak et al. has also shown successes in regenerating organized tissue after severe tissue loss or injury. In addition to providing invaluable educational experience to guide future efforts, this incremental progress moves the field ever closer towards the ultimate goal of developing technologies for safer and more efficacious tissue repairs and replacements. In order for an engineered tissue to perform a predominantly load bearing function and sufficiently recapitulate the mechanical behavior of native tissues, advancements in the current technologies are necessary to attain more complex biological functionality as well as biomechanical stability. It is generally accepted that both chemical and mechanical factors modulate cell biosynthesis when producing extracellular matrix [2-5]. Healthy native tissues undergo intricate multi-scale modes of deformation which work synergistically with biochemical stimuli to determine physiologic responses. Efforts to produce viable tissue replacements which recapitulate mechanical behaviors of native tissues are confounded by the complex multi-scale architectures, hierarchical biological phenomenon and intricate modes of deformation typically observed in collagenous tissues. In order to mimic native tissue structure and organization it

is first necessary to develop techniques to produce scaffolds in a controlled manner with characteristic lengths on a scale comparable to those observed in nature (Figure 1).

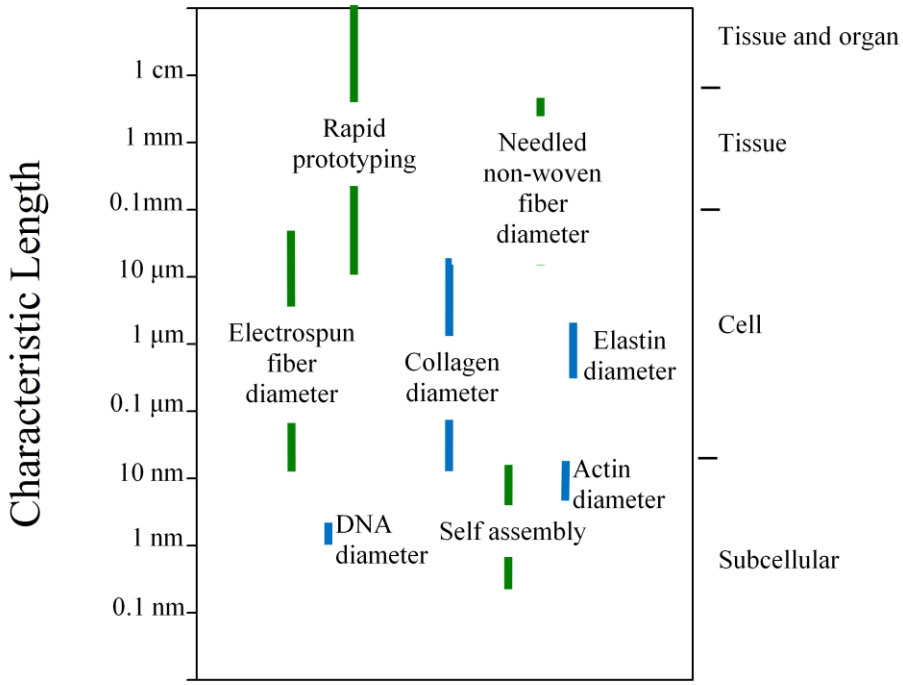

Figure 1. Ability to produce scaffolds which mimic native tissue constituent scales. Viable tissue replacements are confounded by complex multi-scale architectures, hierarchical interactions, and modes of deformation characteristically observed in native tissues. Overcoming the limitations of current medical therapies necessitates new production methods or adaptations to current techniques to produce scaffolds in a controlled manner with characteristic lengths comparable to those observed in nature.

The ability to create engineered tissue replacement options requires a detailed command of the complex, dynamic, and reciprocal interactions which occur at the cell-ECM interface. This includes how mechanical cues from the tissue or organ level are transmitted to the cell or cell components and elucidation of the signaling pathways which comprise the mechanotransduction pathways responsible for the cellular processes observed experimentally. Furthermore, the use of new technologies in the production of engineered scaffolds necessitates a detailed understanding of the structure-function relationship relationships unique to these materials. Currently, the exact microstructural characteristics of engineered scaffolds (that experienced by cellular inclusions) often remain ill defined and presumably will have a profound influence on cellular function. Long term efficacy of tissue replacements or regenerative therapies will rely on the critical processes of cell proliferation and differentiation, the production of organized matrix, and concurrent tissue remodeling or growth. The major focus of this chapter is to present current technologies in use for engineered tissue applications and explore the structure-function relationships critical to their mechanical behavior and their potential ability to perform in a mechanically demanding environment.

1.2 – NATIVE ECM ISOLATION FOR THERAPEUTIC APPLICATIONS

One of the most obvious and earliest implemented approaches for tissue replacement was the use of allograft or homograft tissues. These tissues are typically obtained from cadavers and cryopreserved without chemical cross-

linking to control biological function for transplantation into another individual. As such, there is typically a limited supply of tissue far outnumbered by patients requiring medical intervention. These tissues typically exhibit low infection rates and adequate initial mechanical function but tend to lack long-term efficacy. For example, cryopreserved allograft heart valves exhibit good hemodynamic profiles and require little or no chronic anticoagulation therapy but are plagued by progressive degradation characterized by leaflet distention, fibrosis, possible leaflet calcification or even valve stenosis attributed to somatic growth of the surrounding tissues. Similarly, xenograft tissues could be used with proper tissue processing to overcome immunogenetic responses and stabilize the extracellular matrix. Chemical processing (i.e. cross-linking) of xenograft tissues has long been used in therapeutic applications to stabilize collagen architecture while significantly reducing the risks of immuno-rejection. In fact, bioprostheses comprised of glutaraldehyde fixed porcine aortic valve or bovine pericardium is implanted in at least half of valve replacement surgeries. The use of both allograft and xenograft tissues is a rather intuitive approach as native ECM is by definition an ideal biological scaffold.

The mechanical response of native and glutaradehyde fixed porcine aortic valve leaflets and bovine pericardium have been systematically evaluated in their application to valve replacement therapies [6-10]. After chemical cross-linking, the highly mobile fiber kinematics typical of native valvular tissues is altered significantly owing to altered mechanical behavior such as becoming less extensible [6]. Furthermore, it has been shown that cross-linked tissues for valve

replacement are susceptible to fatigue induce changes which lead to compromised mechanical behavior. These injuries can present in both locations of calcific nucleation and without calcification [11-13]. Lastly, aldehyde cross-linking of tissues with glutaradehyde, which is known to exhibit cytotoxic properties, has been shown to exhibit poor host cell infiltration. This lack of viable cells severely inhibits their growth and remodeling capabilities resulting in an inability to maintain tissue homeostasis and repair structural injury [14]. Unfortunately, the processing techniques required for storage (cryopreservation) or matrix stabilization (cross-linking) of valvular tissue replacements significantly alters tissue mechanical behavior resulting in a continuing risk for morbidity and mortality [15].

More recently, non-crosslinked terminally sterilized biologic scaffolds of naturally occurring extracellular matrix have shown promise as a therapeutic option. For example, porcine small intestinal submucosa (SIS) has been successfully used in multiple applications to treat damaged or diseased tissue in human patients (i.e. musculotendinous, dermal, cardiovascular, gastrointestinal, lower urinary tract). This approach represents a "top down" method to obtain materials for therapeutic purposes whereby undesirable constituents such as cellular materials are stripped away leaving potentially viable ECM. Cellular material such as DNA and membrane proteins have been implicated in the foreign body immune response typical of transplanted tissues and must be removed while organized ECM proteins are well conserved across species causing lower rates of rejection [16]. The work by Badylak et al. has contributed

significantly to this area of research and has focused on the innate and acquired host immune responses elicited by the use of sterilized biologic scaffolds and their associated downstream remodeling events. These materials can exhibit bio-inductive properties facilitating constructive remodeling characterized by cellular infiltration, tissue ingrowth, and production of organized connective tissue [17-18]. Furthermore, the rapid and complete degradation of non-crosslinked SIS is beneficial as it reduces the duration of potential host exposure to antigens. Though there are many variables affecting the viability of these materials, such as tissue source (presence of antigens), rate of scaffold degradation, and manufacturing methods, they have been shown to exhibit adequate mechanical function in load bearing applications.

The mechanical response of these materials is dictated by their fibrous constituent arrangements and kinematics. Systematic evaluations of decellularized tissues are currently limited with the exception of SIS, urinary bladder matrix, and aortic valve leaflets. SIS and valvular tissue sources exhibit anisotropic mechanical behaviors owing to a preferred fiber alignment while UBM (submucosa and tunica propria) has a more isotropic fiber alignment and mechanical behavior [19]. The mechanical resiliency of SIS is typically inadequate but the material mechanical response can be tailored to particular applications by the use of multiple layers. The Restore® device (DePuy, Warsaw, IN) is one example of a commercially available SIS based material for orthopedic tissue reconstruction and is comprised of 10 layers of SIS arranged in such a manner that the final mechanical response is isotropic. In a related study,

Gilbert et al. used a multi-layer SIS device in the repair of canine Achilles tendon. The implant showed the ability to support the remodeling response of host cell infiltration and tissue ingrowth, leading to an organized, dense collagenous SIS-ECM similar to normal tendon tissue without catastrophic loss of function [20].

Despite these intriguing in-vivo results, the manufacturing protocols required to eliminate cellular debris from these tissues can cause profound alterations in mechanical behavior. Chemical cross-linking, as mentioned above, alters the collagen fiber architecture owing to a change in mechanical behavior. In a similar light, the harsh chemical processing and sterilization methods likely alters or disrupts critical matrix micro-structure. In a recent study by Liao et al., the effects of chemical decellularization on mechanical and structural properties were investigated. In short, an anionic detergent, enzymatic agent, and a non-ionic detergent (sodium dodecyl sulfate, Typsin, Triton X-100 respectively) were used to process porcine aortic valve leaflets prior to mechanical testing or structural examination [21]. It was determined that the processing chemicals were responsible for profound increases in tissue extensibility and reductions in flexural rigidity all while no measureable change in gross fiber orientation was observed. Common methods for terminal sterilization of biological materials include ethylene oxide, gamma radiation, and electron beam irradiation and each have unique effects on material behavior. When SIS is terminally sterilized, each of the above sterilization methods has been shown to reduce maximum material stiffness. Both methods of irradiation have been shown to reduce maximum

force at failure and maximum tangential stiffness while ethylene oxide treatment induced marked increases in extensibility without significant alterations to force at failure [22]. It is likely that structural disruptions occurred on a sub-fiber level which could compromise tissue durability possibly precluding them from use in demanding dynamic mechanical environments. The mechanism(s) by which the matrix architecture is altered during manufacturing is currently ill defined and requires further investigation to improve the effectiveness of this technology.

1.3 - SYNTHETIC ECM AND PROCESSING METHODS FOR SYNTHETIC SCAFFOLDS

Though native tissues represent ideal scaffolds in their in-vivo form and function, the use of native ECM based scaffolds can have significant drawbacks. As mentioned above, host recognition of certain foreign features can induce an acute or chronic immunogenic host response compromising implant function. In addition, the processing methods required for the production of these materials can be a source of inconsistencies between specimens resulting in varied degradation behaviors and mechanical responses. The use of biodegradable synthetic scaffolds can circumvent the pathogen transmission and immuno-rejection concerns associated with collagen based tissues from animal or cadaver sources.

Designing scaffolds for tissue repair represents a “bottom up” approach whereby scaffolds are produced with desired chemical, physical, and mechanical characteristics in a controlled and reproducible manner. Common physical

characteristics of interest include: surface texture to promote cell attachment, highly porous microstructure to allow tissue ingrowth, and interconnected porous network to allow adequate nutrient transport and cytokine activity all while maintaining a desired mechanical behavior (Table 1).

Scaffold design criteria	**Resulting function in engineered tissue**
Biologic compatibility	Non-toxic and minimal inflamatory response
Non-thrombogenic	
Non immunogenic	
Low or zero toxicity (degredation products)	
3D matrix architecture	Physiologicly relevant environment for cell function
	Known multi-scale architectural features mediating macro-micro transmission of force
Void space	Highly porous and interconnected pores allow cell infiltration, transport of nutrients, humoral factors, and waste products
Surface chemistry and topography	Cell attachment and cell-matrix interactions
Degredation rate	Scaffold gives way to functional matrix formation
Fiber orientation	Anisotropic mechanical behavior
	Influence orientation of cells and ECM deposition
Sound mechanical behavior	Seamless integration with surrounding tissue(s),
	Able to Withstand in-vivo forces

Table 1. Structural scaffold design criteria and their corresponding function for engineered tissues.

Bulk biocompatible polymers are unable to provide an adequate three dimensional environment which encourages cell viability. Instead, bulk polymers only allow surface interactions and typically exhibit isotropic (or near isotropic) mechanical responses. As such, further processing techniques are implemented to produce 3D scaffolds with desired material behaviors which

encourage cell bioactivity. For the purposes of this article, scaffold morphologies will be segregated into two types: non-fibrous and fibrous. Depending upon their intended application, both the non-fibrous and fibrous scaffolds have unique manufacturing processes and characteristics allowing them to function in a wide range of mechanical environments. Non-fibrous scaffolds typically exhibit relatively poor tissue-like tensile mechanical behavior but can function adequately in environments which experience primarily compressive strains. Fibrous scaffolds on the other hand, have the ability to bear significant tensile loads and can be manufactured to exhibit varying mechanical behaviors.

1.3.1 - NON FIBROUS 3D SYNTHETIC SCAFFOLDS

Scaffold processing techniques of biocompatible polymers such as phase separation (Figure 2 A, D), particle leaching (Figure 2 B, E), and high pressure gas foaming (Figure 2 C, F) can all produce highly porous morphologies and provide a 3D environment to support a viable cell population.

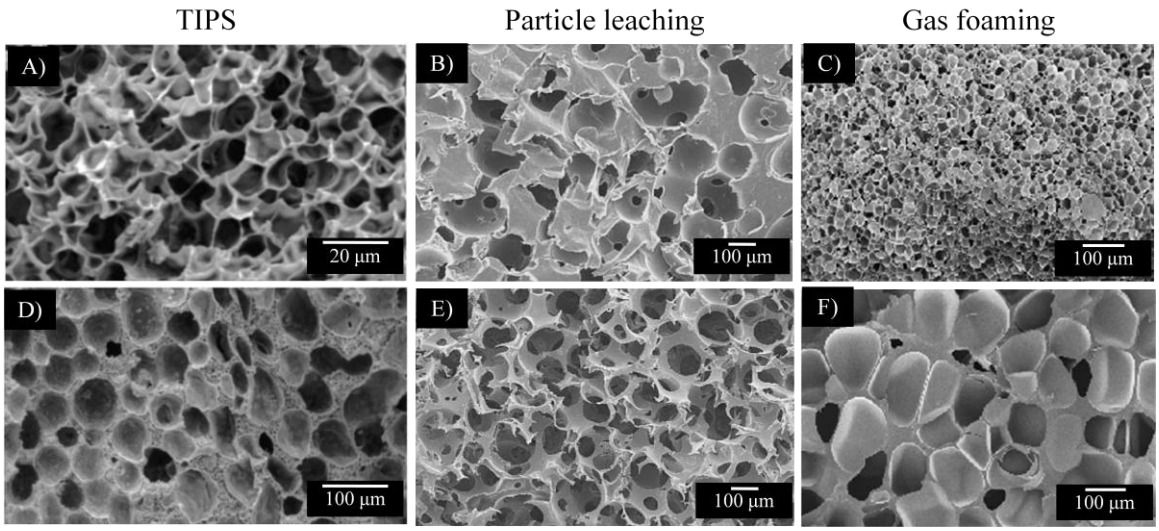

Figure 2. Cross-sectional SEM micrographs of non-fibrous methods to manipulate micro-morphology. (a and d) Porous poly(L-lactic acid) prepared via TIPS at 30 C with metastable state residence times of (a) 5 min, (b) 60 min so as to control pore size. (b and e) Polycaprolactone scaffold created by particle leaching with a poly(ethylmethacrylate) bead (200 lm diameter) porogen. Increased porosity was attained by compressing the beads in a mould prior to injecting melted polycaprolactone. (c and f) Aporous poly(ester amine) sample prepared via gas foaming at 105 C. By increasing the gas saturation pressure, pore density is increased and average pore diameter decreases. Saturation pressure equals 20 bar and 40 bar, respectively.

Each of these processes allows a degree of control over pore size and pore structure; both of which are critical during the culture duration to facilitate cellular ingrowth [23]. For instance, if pore size is too small or are not interconnected, cells will be physically unable to penetrate the scaffold and be limited to surface interactions. A highly porous, interconnected structure also

contributes to the effective transport of nutrients or humoral factors and the removal of waste products necessary to maintain viability of the cell population.

Emulsification/freeze-drying and thermally induced phase separation are two commonly employed phase separation approaches used in the production of scaffolds for tissue engineering. For emulsification/freeze-drying, a synthetic polymer is first dissolved into a suitable solvent and then water is added to obtain an emulsion. Next, the emulsion is cast into a mold and quickly frozen by immersion into liquid nitrogen. Finally, the frozen emulsion is freeze-dried to remove the dispersed water and solvent leaving a solidified, porous polymeric structure. Thermally induced phase separation or liquid-liquid phase separation requires the use of a solvent with a low melting point that can easily be removed through vacuum-drying. By adding a small quantity of water, phase separation is induced creating polymer-rich and polymer-poor phases. Once cooled below the solvent melting point and sufficient duration of vacuum-drying to remove the remaining solvent, a porous scaffold is obtained. The porous media produced by phase separation technique are capable of attaining 95% porosity but the pore size is relatively small (13-35 μm) and is often irregular. In addition, the reliance on cytotoxic solvents for these techniques means that special care must be taken to remove such hazardous chemicals to ensure cell viability.

The process of solvent casting and particulate leaching allows the preparation of regularly porous structures. First, the polymer is dissolved into a suitable organic solvent and then the solution is cast into a mold filled with porogen particles (i.e. inorganic salt, sugars, gelatin or paraffin spheres). The size of the

porogen particles dictates scaffold pore size while the polymer to porogen ratio correlates to the amount of porosity of the final structure. After the polymer solution has been cast and the solvent has fully evaporated, the mixture is immersed in a bath of a suitable solvent to selectively dissolve the porogen. Similar to the phase separation techniques described above, remaining organic solvents from the manufacturing process must be fully removed to avoid any possible damage to cells or surrounding biological material. This technique can produce scaffolds with 87% porosity and pore sizes greater than 100 μm.

Gas foaming represents a processing technique which circumvents the use of cytotoxic organic solvents. First, a gas (i.e. CO_2 or N_2) or blowing agent is added to a melted polymer at high pressure to saturate the material. The high pressure blowing agent is then allowed to attain atmospheric pressure reducing its solubility in the melted polymer while causing the gas-polymer mixture to expand. The expanded volume is then quenched to set the final polymer scaffold structure. Manufacturing parameters effecting pore size include: temperature, degree of saturation, hydrostatic pressure, interfacial energy, and visco-elastic properties of gas/polymer mixture [24]. Mechanical properties of the resulting material can be altered by the temperature at which foaming occurs. This manufacturing process is capable of producing scaffolds possessing pore sizes of 200-300 μm and total porosities up to 93% [25]. Some drawbacks of this process include excessive heat during compression molding, which limits the incorporation of temperature labile materials, and difficulties obtaining well interconnected pore structures.

From a mechanical design perspective, using these techniques must strike a balance between porosity and mechanical integrity to withstand the dynamic environment of a load bearing tissue. Increased porosity necessitates a reduction of scaffold material per unit volume, reducing mechanical strength. Furthermore, tailoring mechanical behavior to emulate tissue mechanical behavior such as anisotropy is difficult and requires an ability to controllably induce structural anisotropy. Recently, it has been shown that the thermal induced phase separation technique can be modified by applying thermal gradients during the separation process to induce oriented pore formation resulting in an anisotropic material response with the preferred material direction being approximately 6 fold stiffer [26-27] (Figure 3 A, B, D). Moreover, the oriented pores could potentially induce a cell micro-patterning effect to orient the cells and the matrix which they produce.

1.3.2 - FIBROUS SYNTHETIC SCAFFOLDS

Synthetic scaffolds comprised of a fibrous micro or nano-architectures present many advantages for tissue engineering applications. Namely, long continuous structures with diameters on the order of native ECM (50 – 500 nm) approximate the local cellular environment well (Figure 3 C, E).

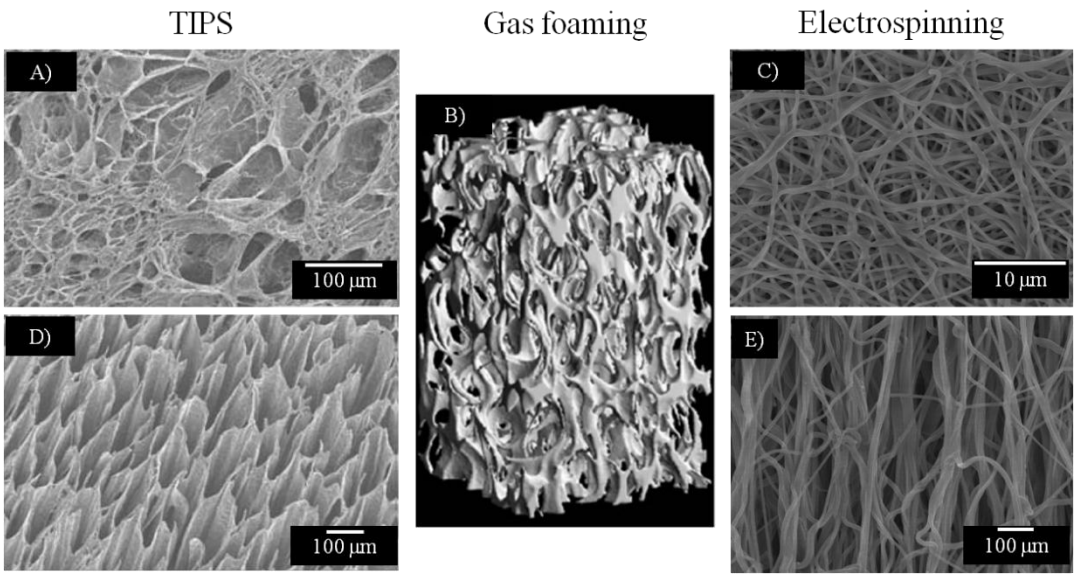

Figure 3. Attaining structural anisotropy through material processing. (a and d) Polyurethane scaffolds produced by thermally induced phase separation with oriented pores produced by imposing a heat transfer gradient during cooling. (b) Supercritical gas foaming used to create open cell composite foams for bone tissue engineering which exhibit morphological and mechanical anisotropy with pores oriented in the foaming direction. (c and e) Electrospinning allows controlled fiber deposition by the use of a rotating collection surface and has the potential to produce scaffolds which exhibit gross, anisotropic soft-tissue-like mechanical behaviors.

A population of fibrous structures makes them appropriate for handling tensile loads while maintaining relatively low bending rigidities. Control of the distribution of fibers during manufacturing enables the production of scaffolds exhibiting a wide array of mechanical behaviors. Furthermore, scaffolds comprised predominantly of fibrous structures provide high surface area to volume ratios and high porosity. These characteristics encourage cell contact and transport of nutrients or removal of waste products respectively.

1.3.2.1 - SELF ASSEMBLY

Self assembling materials are a relatively new technology capable of producing nano-scale fibers (Figure 4 A). The process of self-assembly can be observed throughout the natural world (nucleic acid synthesis, protein synthesis, etc.) and is mediated by weak non-covalent bonds, ionic bonds, hydrophobic interactions, van der Waals interactions, and water mediated hydrogen bonds [28-29]. Individually, these forces are relatively insignificant but when combined, they govern the structural conformation of all biological macromolecules and influence interactions with other molecules [29]. Self assembly or self-organization is a spontaneous event where individual components combine to form an ordered structure with preprogrammed non-covalent bonds within and between molecules. This process can be used to produce a range of structures which can themselves self-organize into superstructures [28]. Currently, this technology is in early stages of its development and would have limited use in a mechanically demanding application.

Controlling the self assembly process is a very complex procedure and fabricating 3D scaffolds with reproducible microstructure and satisfactory mechanical properties poses significant challenges for scientists and engineers. Self assembled materials typically exhibit morphological characteristics on a sub-cellular scale. For instance, fiber dimensions are on the order of 5-10 nm in diameter and 1μm in length with pore diameters ranging from 5-200 nm. As a result of the small characteristic length of these materials, production yields are significantly limited. This in turn means that that this technology is currently

better suited as surface coatings or to be used to develop diagnostic materials comprised of marker proteins. As the production methods mature, this technology may prove viable in the production of organized structures up to the macroscopic scale comparable to native tissues. Furthermore, understanding mechanisms of the self assembly process may prove valuable in attempts to guide in-vivo or in-vitro self assembly to regenerate viable tissues or organs.

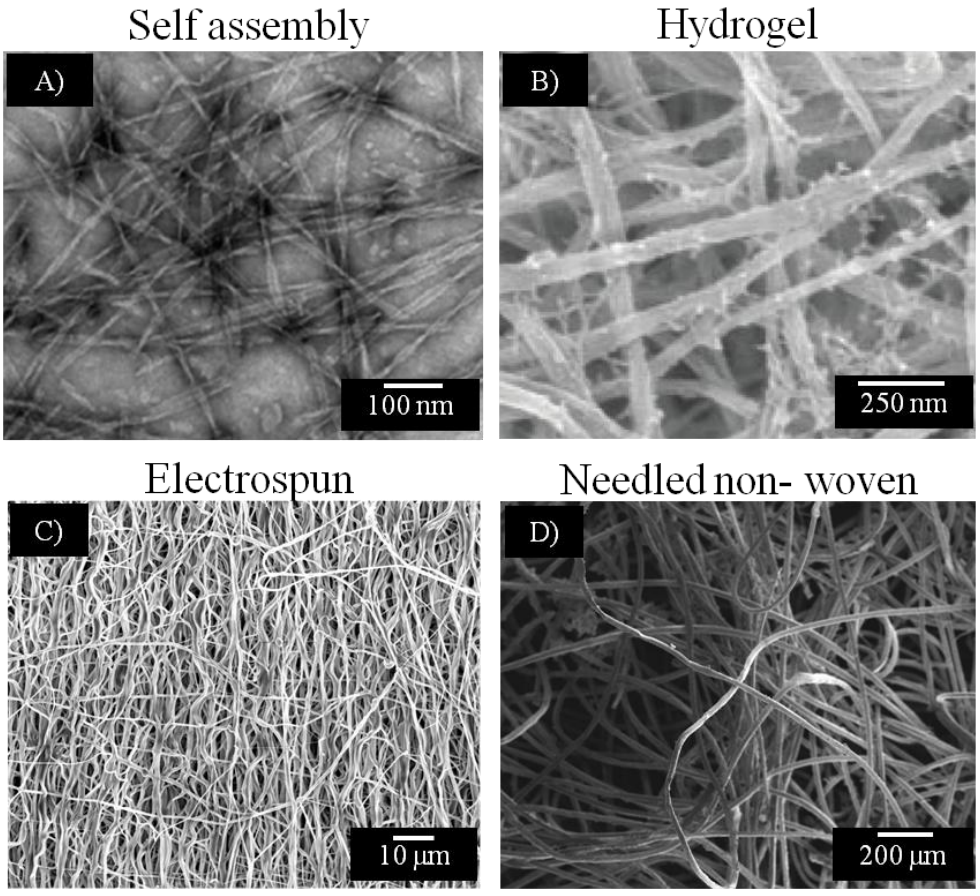

Figure 4. Scanning electron micrographs of methods commonly employed to create 3D scaffolds exhibiting fibrous structures with diameters on the order of native ECM. (a) Nano-scale self-assembled alkylated peptide amphiphiles forming tristed ribbon morphologies. (b) Biopolymer gels like those made of fibrin readily support cellular viability and can be used to investigate cellular

behavior in a reasonably well controlled environment. (b) Electrospinning scaffolds for tissue engineered applications have seen widespread use owing to the inherent ability to produce a synthetic matrix with tunable fiber architectures (diameter, orientation, etc.). (c) Needled non-woven fabrics made of biocompatible polymers have shown promise in tissue engineered efforts to produce de novo ECM in a well-defined microstructure.

1.3.2.2 - HYDROGELS

Hydrogels (Figure 4 B) are comprised of cross-linked, hydrophilic polymers allowing them to maintain large fractions of interstitial fluid (water) [30]. The properties of these materials can be designed for various applications by specifying material characteristics such as biocompatibility, permeability, mechanical and chemical stability, as well as easily controlling gross scaffold geometry and cellular distribution. Cellularized hydrogels constructs can be produced via three main methods; adhesion, matrix entrapment, and mico-encapsilation [31-32]. Hydrogels can be rather weak, exhibiting only marginal mechanical strength. However, their mechanical properties can often be tailored to a reasonable degree via controlled alterations of their microstructure. The density of cross-links (chemical bonds, ionic interactions, hydrogen bonds, physical bonds) directly influences mechanical strength both in shear and tension [33]. Polymerization conditions can also dramatically alter the material behavior of the hydrogel produced. Alterations in reaction time, temperature, and the

amount of solvent used can influence polymer structure, number of cross-links, and type of cross-linking bonds formed.

In tension, hydrogels typically do not exhibit material strengths observed in dense collagenous tissues. Instead, these less robust materials exhibit ultimate stress levels on the order of tens to several hundred kPa. Due to the ability of hydrogels to hold significant amounts of fluid (incompressible) they can be produced to exhibit a high compressive modulus comparable with native articular cartilage. As a result, hydrogel materials are often investigated in orthopedia applications for the treatment or repair of articular cartilage displaying loss of function due to arthritis or acquired via severe trauma to the joint. The only caveat of these materials is their ability to maintain adequate levels of their aqueous component during loading. Fluid loss often results in reduced mechanical integrity, visco-elastic like material behavior, and can even produce anisotropic material behaviors. These phenomena, though interesting and important to global mechanical function of these materials, can obscure true polymer material behaviors under loading. Since polymer chain architecture will dictate gross mechanical behavior it is important to understand how the polymer chains behave individually and how they interact with one another during loading. Currently, the understanding of hydrogel lattice microstructure during deformation is limited and requires further investigation [34-35].

A subclass of hydrogels utilizes protein monomers of biological origin (collagen, glycosaminoglycans, fibrin, etc.) to create artificial scaffolds for tissue engineered applications [36-41]. Processed and purified matrix proteins are

commercially available in their respective monomer forms. Gels can be easily created from these materials by pH- and temperature-dependent polymerization. For example, the group of Tranquillo et al. has extensively investigated the use of fibrin gels as a base scaffold for seeding cells in attempt to develop engineered constructs for cardiovascular applications such as heart valve tissues and vascular grafts and the study of Schwann cell mechanobiology [37, 39-43]. Collagen gels have seen similar use to characterize mechanical behavior as well as to study a variety of cell-matrix interactions and their resulting phenomena such as the effects of matrix stiffness on cellular contraction or the ability of cells to remodel the collagen matrix [44].

Biopolymer gels inherently exhibit good cytocompatibility, are easily formed into physiologically relevant geometries, and can be manipulated to produce constructs which exhibit a relatively large range of mechanical behaviors. Currently, it is not clear whether these scaffolds exhibit a structure-function relationship comparable to native collagenous tissues nor has there been substantial evidence demonstrating the characteristic length of these polymerized biopolymers. In a recent study by Thomopoulous et al.[45], the authors attempted to apply a structural continuum model [46-48] with unsatisfactory results. This may be indicative that these gels do not functionally behave as long fiber composite materials. This may also lend insight to the shortcoming of these biopolymer gels of lacking mechanical integrity, rendering them unable to adequately mimic native tissues in a functional manner. Native valvular tissues

on the other hand exhibit collagen fibers or fiber bundles spanning the leaflet which measure on the order of tens of mm.

1.3.2.3 - NEEDLED NON – WOVENS

Polymer processing techniques originating in the textile industry have proven valuable in producing synthetic fiber meshes which are capable of stimulating isolated cells to regenerate tissue. For instance, carded polymer fibers consolidated into roughly intertwined fiber webs with barbed needles have been shown to support tissue formation [49-50]. Needled non-woven scaffolds (Figure 4 D) can be manufactured quickly, at relatively low cost, and withstand sterilization processes necessary for in-vivo use. Isolated cells of a desired lineage can then be seeded and cultured in static or dynamic conditions. Since these highly porous scaffolds exhibit an open pore structure, the seeded cells can quickly and easily infiltrate the scaffold producing a construct populated with cells throughout.

Flat sheets of PGA/PLLA non-woven textile have recently been employed to recapitulate the geometry of the native pulmonary valve and trunk by Sutherland et al [51]. After seeding and culturing ovine endothelial progenitor cells on the non-woven scaffold, the constructs were implanted into the pulmonary valve position of a juvenile ovine model. After 16- to 20-weeks, the engineered valve constructs were explanted for histological evaluation. Interestingly, histological preparations of the ECM architecture resembled that of native valves remarkably well with a tri-layered structure of organized tissue. To investigate more

fundamental mechanisms of tissue evolution in response to mechanical cues and their resulting effects on mechanical behavior, Engelmayr et al. developed a modeling framework for the flexural properties of these needled non-woven scaffolds [50, 52]. In short, the structural model accounts for unique fiber morphologies which arise from the fabric manufacturing process and the production of new ECM to predict the scaffold's effective stiffness during bending which was in agreement with experimental values. Furthermore, it was determined that matrix deposition during culture resulted in increased scaffold stiffness and can be attributed to an increased number of fiber-fiber bonds.

1.3.2.4 - ELECTROSPINNING

The final fibrous scaffold production technique covered in this work is electrospinning (Figure 4 C). Scaffolds fabricated by electrospinning natural polymers, synthetic polymers or polymer blends have received widespread attention. Beyond its relative affordability and simplicity, this popularity is largely a result of a versatile manufacturing process where slight alterations during fabrication enable the production of scaffolds with a wide array of fiber morphologies (i.e. fiber diameter, porosity, packing density, orientation, etc) directly influencing bulk mechanical properties [53-54]. One commonly employed method for controlled mechanical anisotropy is attained by using a rotating collection surface which induces a preferred fiber direction as the rotational speed of the collector increases [55-57]. This ability is extremely beneficial in mimicking native tissue architecture and has even been shown to

approximate the highly nonlinear biaxial mechanical response of collagenous soft tissues, such as the native porcine pulmonary valve leaflet [55]. Similarly, the electrospinning process has shown the ability to produce biocompatible polymer constructs which exhibit tissue-like mechanical behaviors comparable with nucleus fibrosis tissues or bone [55, 58-59]. Electrospinning produces continuous fiber scaffolds exhibiting a wide range of mechanical properties, while also providing suitable surfaces for cell proliferation and growth [58-63]. A substantial amount of work can be found in recent literature concerning the mechanical and structural characterization of electrospun scaffolds [55, 58, 64-67]. Initial attempts to produce electrospun scaffolds for tissue engineering were concerned with the production and characterization of the materials including uniaxial tensile properties, measurements of porosity, and fiber diameter. Courtney et al. were the first to characterize multiaxial mechanical behavior of electrospun fabrics via planar biaxial testing. The production of a continuous fiber has the added benefit of creating multiple interrelated functional length scales, a characteristic observed in biological materials.

While electrospinning can fabricate scaffolds that possess ECM-like structures, this morphology also results in pore sizes that are generally smaller (<5 μm) and more irregular than those produced by some of the non-fibrous production methods introduced above [68-69]. While it may be possible that cells seeded on the surfaces of electrospun matrices can migrate into the interior by displacing or enzymatically degrading individual fibers, an extended culture period and appropriate signals for cell migration into thick construct interiors

might also be required. Thus, while electrospinning permits fabrication of biodegradable matrices that resemble the scale, architecture, and mechanical behavior of the native ECM [70], achieving high cellular density and infiltration remains challenging. To overcome this limitation typical of electrospinning, we employ a technique which involves the concurrent deposition of electrospun polymer and electrosprayed cells suspended in culture media[60]. This process integrates a cell population throughout the construct and is the subject of active work in our lab

1.4 - ENGINEERED TISSUES AS MODELS SYSTEMS

Although the feasibility of many of the technologies for the production of cell-based engineered tissue constructs mentioned above have been demonstrated, the long-term function, safety, and efficacy of these tissue replacements as well as their capacity to grow and adapt remain largely unknown. The challenge with designing engineered tissues for load bearing applications requires more than matching a single mechanical parameter as many tissues exhibit complex viscoelastic, anisotropic, and highly nonlinear behaviors. Long-term function requires the ability to withstand in-vivo stresses of significant magnitude and temporal loading regimes from the time of implantation. Deficient mechanical integrity at the time of implantation will likely lead to catastrophic implant failure or adverse integration of the construct with surrounding tissue. As such, the ability to control scaffold composition, 3D geometry and structure at multiple scales provides a unique opportunity to

recapitulate critical mechanical behaviors of the tissue while elucidating fundamental biological phenomenon (Figure 5).

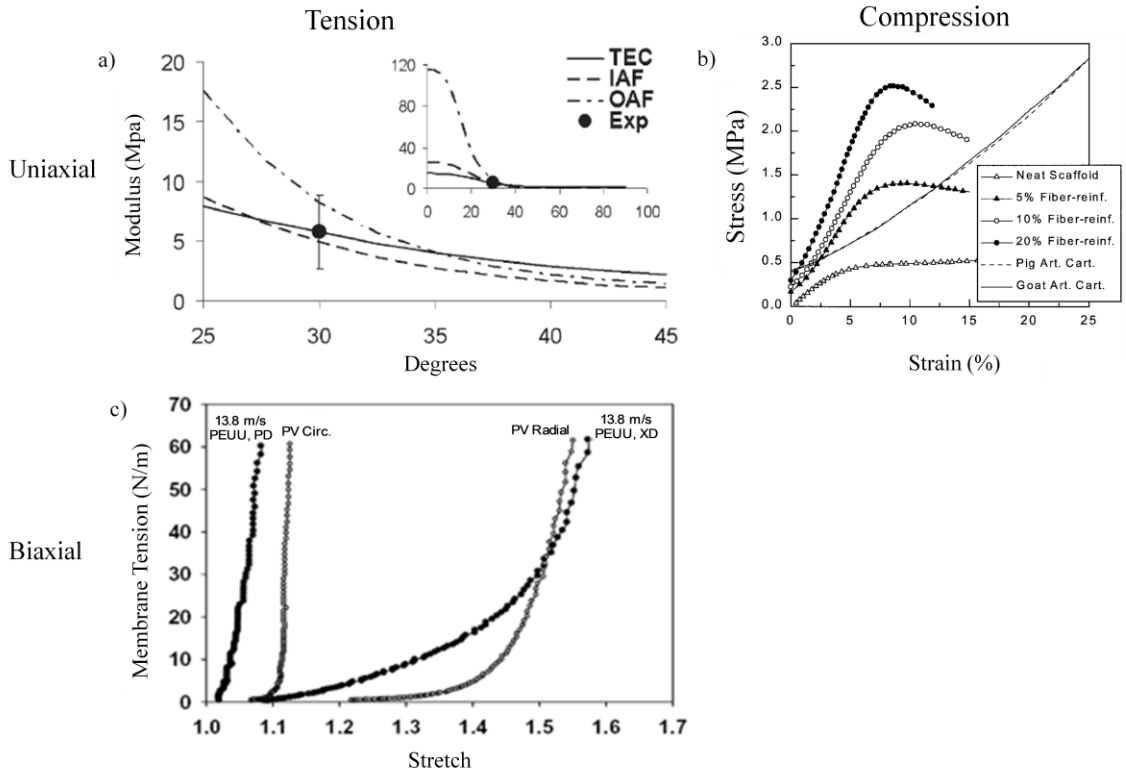

Figure 5. Synthetic scaffold production to recapitulate native tissue mechanical behavior. (a) Homogenation model prediction of material modulus for native inner and outer annulus fibrosus lamella (IAF and OAF, respectively) and annulus fibrosus cell seeded electrospun scaffolds. (b) Representative stress vs. strain curves from compression testing of porous poly(lactide-co-glycolide) scaffolds and articular cartilage sourced from porcine and ovine models. (c) Planar biaxial response comparison of native porcine pulmonary valve and highly aligned electrospun poly(ester urethane) urea construct.

Laboratory tissue culture devices such as bioreactors enable the development of engineered tissues in a controlled mechanically active environment. Mechanical stimuli ranging from cyclic tension, compression, or bending to altered hydrodynamic conditions [71] can be investigated independently or in conjunction to create complex deformation modes approximating those encountered in-vivo. It is well accepted that mechanical stimulation has a profound impact on cellular processes [72-73]. For example, endothelial cell orientation and morphology is known to be influenced by mechanical stimulation such as strain or shear stress [74]. Cellular differentiation of mesenchymal cells can be guided through the application of compressive forces [75-76]. Song et al. have shown that cyclic strain promotes proliferation of rat bone marrow mesenchymal stem cells [77]. Similarly, both static and cyclic modes of mechanical stimulation have been shown to alter protein synthesis and the amount and integrity of ECM proteins [2, 78-81]. Engelmayer et al. demonstrated that cell under simple cyclic flexural deformation; seeded needled nonwoven scaffolds produced new ECM which was deposited in the polymer fiber interstitium. As a result, the effective nonwoven scaffold-ECM composite stiffness was measurably increased [52].

Bioreactors aid in the maturity of cell-based constructs into engineered tissues to improve construct mass, composition of ECM constituents [82-84], and cell proliferation prior to implantation (Figure 6).

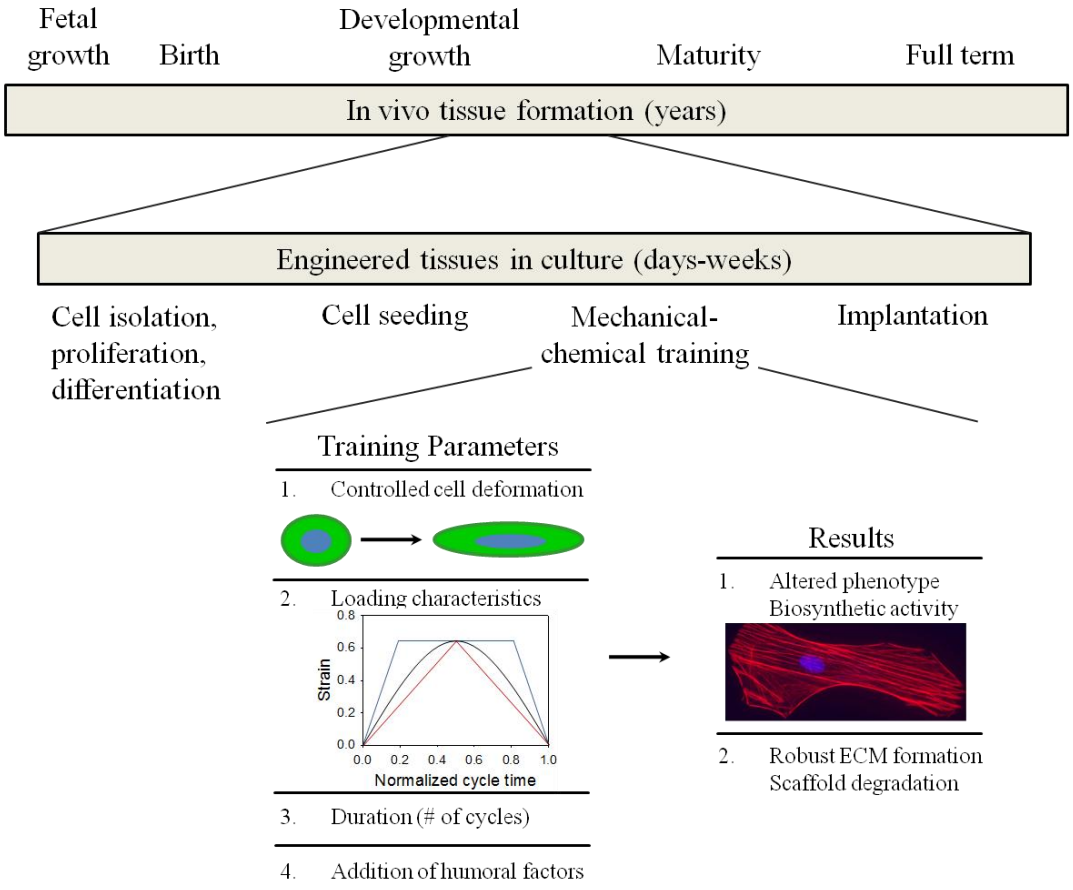

Figure 6. Expediting natural processes for the production of engineered tissue technologies. The time span required to produce practical options for tissue engineering is constrained significantly compared to native tissue development. As such, it is necessary to tailor culture parameters including mechanical and chemical cues in an effort to improve construct mass, composition of ECM constituents, and cell proliferation prior to implantation.

Moreover, such devices are ideal for the investigation of the more fundamental biological phenomenon revealing mechanisms of cell function. For instance, the controlled environment can be used to guide stem cell

differentiation or other cellular processes by physical stimuli [85-88] both in the presence and without humoral or pharmacological factors [2, 89-92]. Despite our growing understanding of the cause-effect relationship of mechanical stimulation on cellular processes, the specific mechanisms responsible for these phenomenona continue to be poorly understood. Not only are these phenomena unclear for native tissues, it is particularly true of cells embedded within three dimensional synthetic scaffolds [34, 64].

Development of engineered tissue or organ replacements must rest on a strong fundamental knowledge of cellular interactions with the local environment and how these interactions span multiple length scales to contribute to the overall function [93-95]. Cells interact with local matrix proteins via focal adhesions, a transmembrane complex of integrins and other proteins such as focal adhesion kinase, talin, and vinculin. Focal adhesions connect cytoskeletal actin fibers with ECM allowing the cell to sense or communicate with its environment [96] which then dictates cell shape, motility, and adhesion characteristics [97-99]. It is known that cell morphology profoundly affects a range of cellular functions, and that changes in the cell cytoskeleton lead to altered stress levels imparted on the nucleus, ultimately affecting cell function. For example, Thomas et al. [100] showed that gene expression and protein synthesis of primary osteogenic cells were altered by changing nuclear shape. Specifically, collagen Type I synthesis correlated directly with nuclear shape, where certain values promoted maximum synthesis, supporting the concept of gene expression and protein synthesis based on optimal distortion of the nucleus. Guilak et al. investigated chondrocyte

nuclear deformations under compressive loads in articular cartilage in an attempt to explore how cell deformation may be a stimulus to cell metabolic activity [93]. They observed a reduction in chondrocyte volume with compressive loading, linked to mechanical transduction and signaling through mechano-sensitive channels [5]. Under the hypothesis that cellular deformations which occur during tissue or organ level motion or deformation, we have recently conducted cell morphology studies on native valvular tissue and engineered constructs for valve replacement during deformation (Figure 7-8) [101].

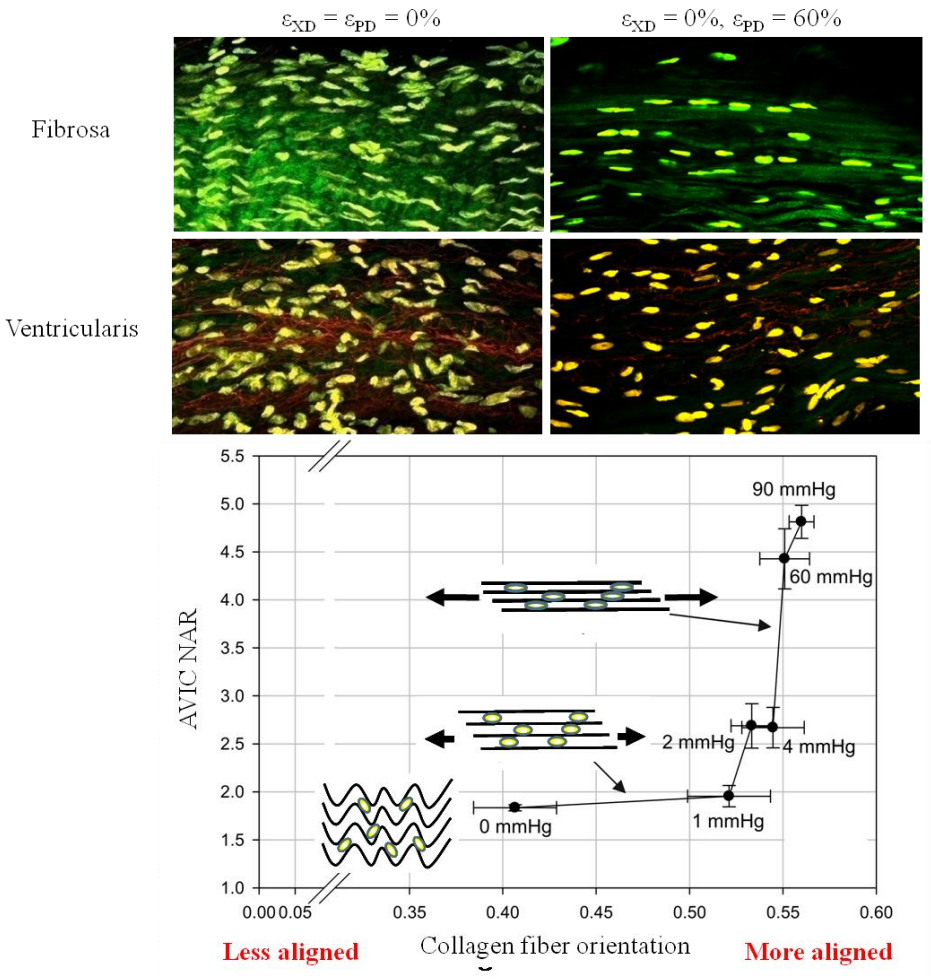

Figure 7. Native aortic valve cell–matrix interaction and cell deformation response to increasing transvalvular pressure. Cell deformation in native porcine aortic valve leaflets,

as quantified by changes in nuclear aspect ratio, was highly dependent on local collagen fiber kinematics. Generally speaking, increasing transvalvular pressure resulted in increased cell nuclear aspect ratios but unique layer-dependent responses were observed. Furthermore, a bimodal trend was observed between cell deformation and increased diastolic loads.

Aortic valve interstitial cell (AVIC) deformation behavior in response to increasing transvalvular pressure has been shown to be mediated by local fiber kinematics. Specifically, a bimodal response was observed where little AVIC deformation occurs with the large amount of fiber straightening for pressures below ~1 mmHg, followed by substantial increases in AVIC nuclear aspect ratio from 4 to 90 mmHg (Figure 7) [101]. Taken as a whole, AVIC responses to tissue level stresses are modulated through complex micromechanical and fiber-compaction effects that occur under physiological stress levels.

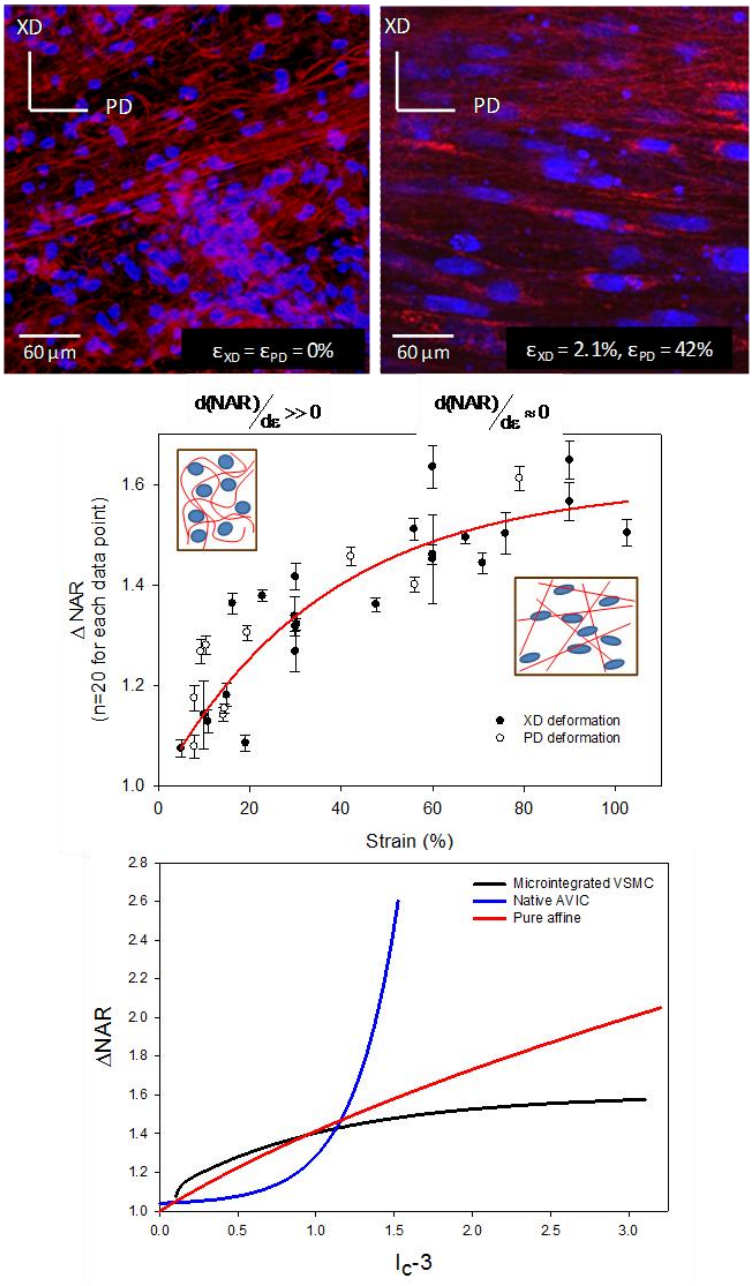

Figure. 8. Strain-induced changes in electrospun polyester micro-architecture and resulting nuclei deformation. When exposed to biaxial modes of deformation, electrospun fibers were observed to transition from a tortuous configuration in the unstrained state to an interconnected web-like architecture at high strains. A composite of all NAR measurements (mean ± s.e.m.) demonstrated a rapid increase to 60% strain, after which a plateau was observed

with further strain increases, indicating that nuclei deformations are dominated by local fiber straightening. A composite cell–scaffold deformation response (bottom) is provided for native porcine aortic valve leaflet, cell integrated electrospun PEUU, and a theoretical purely affine cell deformation response to macroscopic strain.

In a similar manner, cell micro-integrated electrospun scaffolds exhibited micro-fiber morphologies and kinematics that were shown to directly influence local cellular deformations (Figure 8). For instance, in the unstrained configuration the polymer fibers exhibited a tortuous architecture which transitioned to a web-like network of straight, interconnected fibers at high levels of strain (Figure 8). The deformations of the micro-integrated cells were found to be primarily mediated by this phenomenon. The cell integrated electrospun constructs underwent fully recoverable large deformations akin to many native tissues. Moreover, while a non-linear relation between the tissue strain and NAR was observed for both the aortic valve and cell integrated electrospun PEUU, the underlying micro-mechanical mechanisms were clearly different. Initially, the integrated vascular smooth muscle cells exhibited a rapid increase in NAR as fibers straightened and tortuosity was reduced. Once the PEUU fibers became straightened and the architecture transitioned to an interconnected web like structure, changes in NAR were observed to plateau.

Due to the particular methods employed to investigate the deformation responses of cells in the native porcine aortic valve and cell integrated

electrospun constructs it can be difficult to interpret their respective behaviors. For comparison purposes, the measured deformation responses were cast into invariant forms of their right Cauchy-Green strain tensor (**C**) where $\mathbf{C} = \mathbf{F}^T\mathbf{F}$. The first invariant of the Cauchy-Green tensor was chosen to compare the cellular deformation response to macroscopic matrix deformation for three cases (Figure 8). Namely, the experimentally measured deformation responses of the AVIC in response to increasing transvalvular pressure, the response of cells integrated in electrospun scaffolds under strip biaxial planar deformations, and the simulated response of a cell which convects in a pure affine manner with global scaffold planar deformation. In the simulated case a planar strip biaxial deformation, characterized by deforming the specimen on one axis to a desired level while holding its orthogonal component fixed, was imposed as it was similar to imposed deformations of the cell integrated electrospun constructs. The unique cellular deformation responses observed in the native aortic valve and cell integrated electrospun constructs were preserved in this invariant form with the AVIC exhibiting large changes in nuclear aspect ratio (Figure 8). The simulated affine response was seen to fall between the integrated VSMC and AVIC behaviors at high strain.

From the examples cited above, it is clear that force induced changes in nuclear shape strongly correlate with cell function and phenotype. Empirical results indicate a wide array of potential cellular adaptations in response to exogenous forces ranging from cell morphology and metabolic activity to cell motility or proliferation. As incremental improvements in the understanding of

mechanical and chemical cues on cell function are obtained, it is then possible to exploit these mechanisms to accelerate ECM protein production and formation to prepare engineered tissues prior to implantation. Currently, much is known about the arrangement and connectivity of load bearing elements from the ECM to cell cytoskeletal structure and further to nuclear structure. Despite this reasonably detailed knowledge, the field currently lacks a detailed mechanistic understanding of the role of force on nuclear mechanotranstuction and requires further investigation.

1.5 - EMULATING NATIVE TISSUE MECHANICAL BEHAVIOUR

All of the scaffold manufacturing methods cited above can be employed to produce a 3D scaffold appropriate for cell culture and will likely foster the production of new tissue. However, the fields of tissue engineering and regenerative medicine continue to struggle to adequately match the mechanical behavior of native tissues. Emulating a single measure of mechanical behavior is often insufficient. For instance, a material's tensile or compressive modulus is often chosen as a design requirement so as to prevent catastrophic implant failure and little concern is given to how the material responds during the loading and unloading regimes (Figure 9).

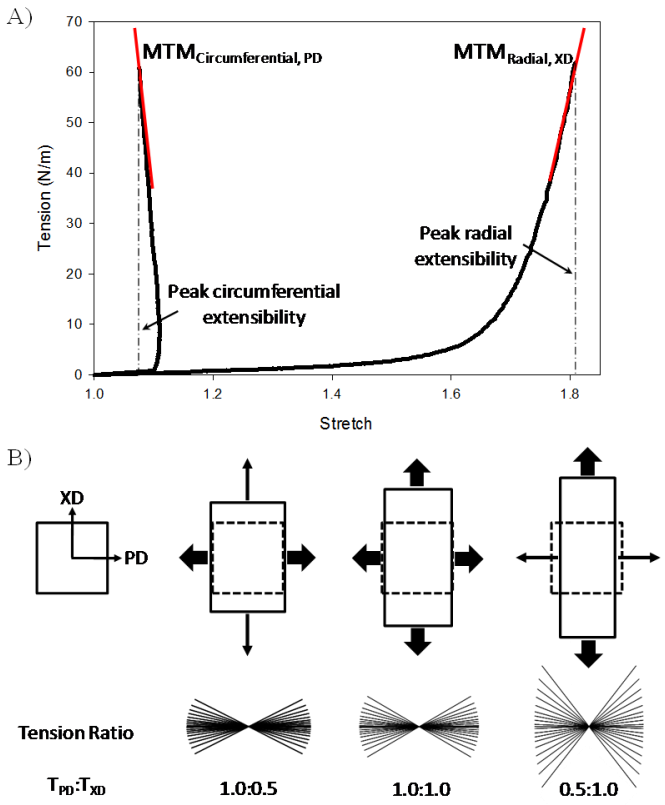

Figure 9. Application of traditional engineering metrics of material behavior are often inadequate descriptors of the complex behaviors observed in biological materials. (a) Load bearing matrix rich biological materials, like native aortic valve leaflets, typically exhibit highly non-linear and highly anisotropic tensile behaviors. (b) The transition from low to high stiffness is attributed to a coupled fiber recruitment process capable of exhibiting lateral contraction at high stress levels (circumferential shortening). As a consequence, the determination of traditional engineering indices of material behavior, such as tensile modulus has little meaning (i.e. negative modulus). Adequate characterization of biological materials can necessitate more sophisticated methods to characterize their true mechanical behaviors.

Disparities in gross mechanical behavior will likely lead to insufficient implant performance, undesirable tissue formation, or no production of new viable tissue. Electrospun materials have been shown to exhibit many complex, soft-tissue like material behaviors such as material anisotropy, significant non-linearity, and the ability to recoverably undergo large strains. Courtney et al. were the first to systematically investigate the planar biaxial mechanical behaviors of an elastomeric electrospun scaffold and model this response. In short, several scaffolds were manufactured with varying degrees of structural anisotropy or fiber alignment which was controlled by altering the rotational velocity of the collection mandrel. Increasing rotational velocity results in an increase in fiber alignment and was experimentally measured from SEM micrographs via an image analysis algorithm. The controlled structural anisotropy produced during manufacturing directly led to more non-linear, anisotropic material responses. These structural properties form a complex 3D scaffold with tunable tissue-level mechanical behavior that can be remarkably similar to the gross biaxial mechanical response of the native pulmonary valve leaflet [55]. Furthermore, modeling attempts originally developed for dense collagenous planar tissues proved quite successful in capturing the mechanical behavior of electrospun scaffolds under biaxial modes of deformation[55]. It was noted that the model predicted a higher degree of fiber orientation than measured experimentally. This is attributed to structural characteristics of the polymer fibers measured in the image analysis algorithm but are not accounted for in the model formulation.

In order to improve upon these initial modeling efforts and gain a better appreciation of how these materials function across multiple length scales, our lab has conducted additional studies to quantify additional structural characteristics. The electrospun scaffolds investigated in this study exhibited complex, hierarchical architectures spanning multiple length scales (Figure 10).

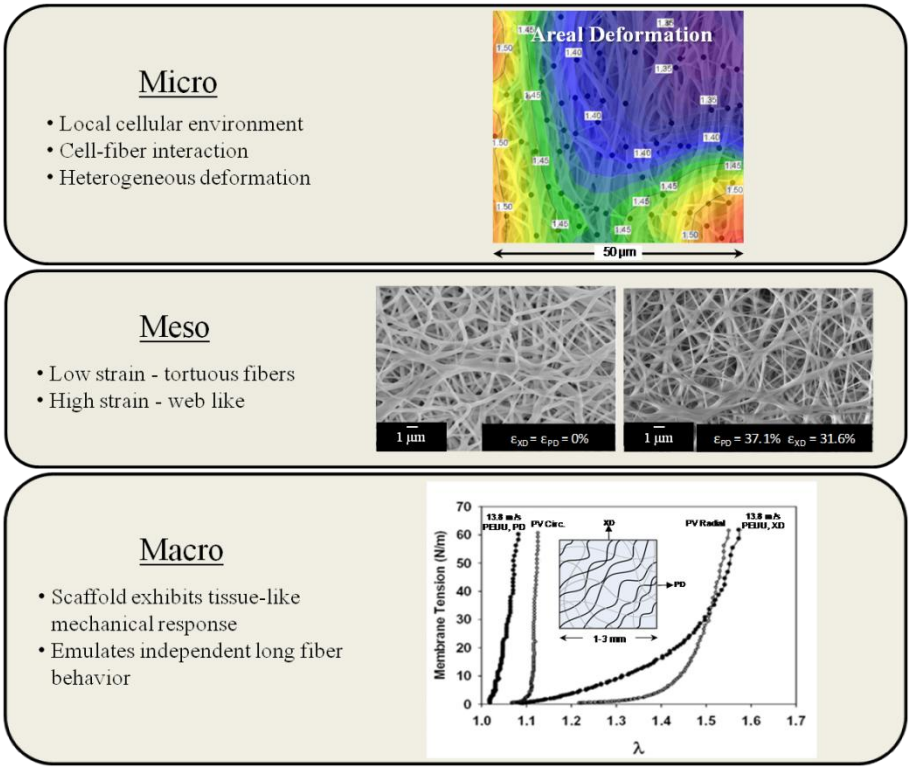

Figure 10. Cell integrated electrospun scaffold hierarchical structure and function. Despite exhibiting a tissue-like mechanical response at the macro scale, the scaffold exhibits vastly different micro and meso mechanical behaviors. For instance, at the micro-scale a heterogeneous deformation response is observed. In addition, fibers in the unstrained configuration exhibit an undulated or tortuous morphology which transitions to a highly interconnected web-like

architecture at finite strains. At the macro scale, we observe a complex 3D scaffold with tunable tissue-level mechanical behavior that can be remarkably similar to the biaxial mechanical response of the native porcine pulmonary leaflet.

As a result, understanding the mechanisms by which these materials deform and behave under various loading conditions is not an elementary task. Fiber tortuosity, diameter, and fiber orientation distributions were all quantified as the electrospun specimen underwent planar biaxial modes of deformation. In addition, the deformation behavior of the scaffold was investigated across multiple scales by defining three characteristic lengths (micro (1-2 μm), meso (40-50 μm), and macro (1-3 mm)). Fiber tortuosity, a measure of how much a fiber deviated from being straight, in the unloaded scaffold, was observed to be dependent on both mandrel velocity during production and orientation. As the scaffold underwent planar biaxial modes of deformation, fiber tortuosity is extinguished and substantial fiber rotational kinematics was observed contributing to an intricate fiber recruitment process [64]. Electrospun constructs were observed to follow gross affine fiber transformations and can be described in a manner similar to collagenous scaffolds [19]. Interestingly, some fibers were observed to rotate or change their direction of orientation during deformation while as a population, no net change was measured. This is likely an additional manifestation of the local heterogeneity which exists at the micro scale. With increased specimen deformation, a monotonic decrease in PEUU fiber diameter

was measured for all specimens. Furthermore, it was observed that neighboring the fibers were well attached where they overlapped or intersected impeding translation of fibers with respect to one another but does not appear to inhibit rotational fiber kinamatics about these points of intersection. Johnson et al. presented a similar hypothesis for reduced fiber mobility in electrospun polymers exhibiting "point bonding" [102]. In short, polymer sintering was utilized on electrospun poly (ε-caprolatone) scaffold to invoke definite point bonds between fibers. Scale-dependent variations in deformation were observed and were attributed to the complex, spatially variant structure which results from the electrospinning process. These heterogeneous variations occur at the micro-scale which a cell might experience. The overall strain behavior tends to become increasingly more homogenous as the scale of interest approaches the tissue level.

We also recently investigated the unique coupled matrix-cell deformation response of electrospun scaffold integrated with cells (Figure 8) [64]. The scaffolds exhibited micro-fiber morphologies and kinematics that were shown to directly influence local cellular deformations. For instance, in the unstrained configuration the electrospun fibers exhibited a tortuous architecture which transitioned to a web-like network of straight, interconnected fibers at high levels of strain. The deformations of the micro-integrated cells were found to be primarily mediated by this phenomenon. The cell integrated constructs underwent fully recoverable large deformations akin to many native tissues. Initially, the integrated cells exhibited a rapid increase in NAR as fibers

straightened and tortuosity was reduced. Once the PEUU fibers became straightened and the architecture transitioned to an interconnected web like structure, changes in NAR were observed to remain constant. Microintegrated cell deformation was mediated by the local reduction of tortuosity or straightening of the PEUU fibers. Thus, cell-scaffold interactions can be subtle and can bring about unique deformation behaviors.

These results indicate that it may be possible to successfully emulate gross native tissue behavior without exactly replicating their highly complex micro-architectures (Figure 10). Attempting to delineate the individual contributions of structural fiber characteristics such as tortuosity and fiber kinematics to the constructs mechanical behavior would be quite cumbersome and better lends itself to the development of a numerical framework to explore this unique, interrelated phenomenon.

1.6 - MECHANICAL MODELING

Tissue engineering and regenerative medicine have evolved from fields which strongly rely on empirical findings to solve practical problems or drive our understanding of complex biological phenomena. Despite the significant advancements made in recent years, there continues to be a multitude of fundamental questions which remain unanswered inhibiting the production of truly functional tissue surrogates. Answers of many fundamental biological or structural questions cannot be ascertained through experimental testing alone. By combining well posed theoretical frameworks with experimentally derived

observations it is possible to elucidate the complex interrelated mechanisms presented to us by nature. Not only can modeling approaches serve as valuable tools to simplify and test our understanding of complex biological systems but they can be used to guiding future hypothesis based investigations.

The development of constitutive relations for soft tissues has been a rich field of study for several decades beginning with the seminal work by Fung [103]. Early constitutive models for soft tissues, though successful at capturing characteristic tissue behaviors for a spectrum of applications, were phenomenologically based. This inherently limits their ability to probe underlying mechanisms governing tissue behavior. In response, structural approaches aimed at characterizing material response in terms of the underlying tissue constituents has gained favor [47, 104-106]. These approaches extend into current trends in the biomechanics community which have focused on computational implementations of established soft tissue models or new approaches specifically aimed to investigate underlying tissue structure-function mechanisms across multiple functional scales.

Correspondingly, methods to restore, maintain, or improve tissue or whole organ function must incorporate a thorough understanding of the intricate multi-scale hierarchical arrangements typically found in nature. Engineering sustainable solutions concerned only with tissue or organ level function belies the multifaceted, coordinated function of these tissue structures and their constituents which are in turn a result of cellular or subcellular processes that reach down to the molecular scale of protein interactions and gene transcription. Surely, one

model cannot incorporate this large range in scales with our current understanding of biological processes. Instead, a hierarchy of models and approaches is necessary to connect the established continuum level relationships (i.e. phenomenological [103, 107-110], structural [47, 104-105, 111-115]) with underpinning cell and subcellular events. From a modeling point of view, a vital aspect of these models consists of the difficult task of seamlessly coupling various length scales.

For materials with regular, repeating structures, unit cell based modeling approaches can successfully relate microstructural responses to global material behavior. Stylianopoulos et al. have developed such a model for the mechanical behavior of collagen fiber networks [35, 116-117]. Briefly, the unit cell, the representative unit of the continuum that encompasses the periodicity of the microstructural parameters, contains an idealized fiber mesh generated in-silico. A group of unit cells are then perturbed in some defined manner (i.e. uniaxial tension) and a force balance within each unit cell results in a volume-averaged macroscopic response. In related work, Zahalak et al [118] and Marquez et al. [119-120] have developed constitutive relations to relate individual cellular contributions to macroscopic material response in an effort to elucidate active and passive cell deformation responses and material properties. The fundamental unit of this model is comprised of cells, idealized as contractile rods, within a compliant matrix. Both constituents were parametrically assigned linear elastic, isotropic material properties. Due to these assumptions, the predictive capabilities are limited since biologic materials typically exhibit nonlinear

viscoelastic behaviors. However, they did show that the strain experience by the cell can be related to macroscopic strain via a scalar valued strain factor and that reasonable approximations of cell stiffness can be determined from measured tissue properties. One shortcoming of the unit cell approach is that it neglects the structural heterogeneity seen in biological tissues and the incorporation of increased structural complexity is penalized by significant computational demands. Moreover, native dense collagenous tissues are long fiber composites with fiber lengths up to the mm scale while the characteristic length scale of this model is much shorter as it is defined by unit cell dimensions.

Additionally, insufficient efforts have been spent on assessing appropriate material specific representative volume element (RVE) size. Morphology descriptors produced through image analysis such as material porosity, fiber density, fiber alignment distribution, fiber connectivity distribution, and fiber diameter strongly depend on the material architecture at micro-meso level. This can be demonstrated studying the evolution of these parameters over regions of interest of increasing sizes and/or repeating the image analysis over analogous regions differing in location. Morphology feature fluctuations and location dependency gradually cease as the analyzed region of interest approaches an appropriate RVE size. A direct implication is that an image analysis technique remains incomplete if it does not identify an appropriate RVE for the variable of interest. RVE size in random composites can be derived statistically, numerically or empirically studying the stabilization of the analyzed variable over RVE's of increasing sizes. Thus, the importance of identifying the appropriate RVE size is

twofold. Firstly, in terms of material characterization, it contributes to develop reliable tools to assess scaffold manufacturing process repeatability and secondly, in terms of mechanical modeling, the analysis performed at the RVE can provide physically meaningful data. In particular, structural deterministic models dramatically depend on rigorously defined material structural descriptions. This need has been extensively highlighted in recent literature where the model capability relies prevalently on the accuracy of the network topology [35, 116-117, 121].

Furthermore, not only is micro-architectural data extraction accuracy crucial but it is fundamental for stochastic representations of engineered scaffolds such that the error between the real and simulated structure is minimized. The potential of a structural deterministic approach in elucidating the inherently multi-scale nature of native and engineered soft tissues response seems to justify its apparent complexity [35, 116-117, 121]. An alternative path of the deterministic modeling is to reproduce the entire scaffold area or volume without duplicating the RVE. The main benefits of this alternative solution are that the implicit error introduced by the multi-scale approach is removed and the information at the meso level is preserved. For instance, tortuosity measurements of a collagen fiber in a heart valve leaflet under loaded and unloaded configurations can be performed which would otherwise be neglected in the multi-scale approach where the fibers cannot cross element boundaries. However, these expected benefits are counterbalanced by a significant increase of computational cost.

The modeling strategies outlined above also have potential use for the rational design of future scaffold morphologies at the macro (tissue engineered construct size, shape etc.) and micro level (fiber connectivity, fiber density, fiber alignment etc.). This represents a profound advancement in the fields of biological science and tissue engineering whereby empirically driven experimentation can be augmented or replaced with more rational design approaches. Relating macroscopic kinematic events to the cell environment and understanding the cellular responses to these cues is critical in the production of engineered tissue surrogates. It has been shown that mechanical cues modulate many cellular processes and the ability to understand and predict the events leading to healthy tissue accretion or adaptive repair/growth can guide mechanical training regimes to produce robust tissue formation. Efficacious repair or replacement of abnormal or lost tissue relies on our ability to reproducibly control cellular responses to exogenous cues.

1.7 - AIMS OF THIS WORK

The development of efficacious therapies for tissue repair, replacement, or regeneration rests in large part on our ability to employ new materials, manufacturing and processing techniques, and manage the events of cellular mechanobiology. Understanding factors responsible for tissue function and dysfunction necessitates a strong fundamental knowledge of native biological material structure and function across multiple scales. Cells continually assess their local environment through various mechanosensors (focal adhesion

complexes, transmembrane proteins, stretch activated ion channels, etc.) and react accordingly by activation of signaling cascades to produce physiologic responses. Furthermore, the intricate interactions between cells and their environment dictates mechanotransduction of proteins critical for cell function and maintaining a mechanically sound, organized matrix. From a tissue engineering perspective, managing cellular processes through controlled exogenous cues has immense and widespread implications. As our understanding of the structure-function relationships governing cell mechanics expands, gaining a mechanistic understanding of how mechanical stimuli translates to protein mechanotranstuction will become feasible. This detailed understanding will then enable rational design of efficacious clinical approaches for improving health along with sound hypothesis driven examinations of new problems and even provide analytical tools to evaluate engineered implant performance. The study illustrated in this dissertation try to address these issues utilizing the electrospinning technique and poly (ester urethane) ureas scaffolds as a learning platform. This work aim to connect the material fabrication parameters with the material micro-meso structure and, in turn the structure with the scaffold mechanical response at different scale levels. Following the similar approach for a given mechanical response a different scale levels it is possible to associate an ideal scaffold structure to a specific clinical application. The targeted clinical application is the pediatric heart valves. This manuscript is organized as follows: the second chapter describes the image analysis technique developed to quantify scaffold architecture, the third chapter summarizes the mechanical

testing results on Electrospun PEUU scaffold and focuses on the relationship between the fabrication parameters and the material micro architecture. Chapter number four describes the mechanical formulation introduced, chapter five reports on in vivo preliminarily data obtained implanting the studied ES-PEUU scaffold on an ovine model. Finally, chapter 6 concludes this work suggesting future directions.

CHAPTER 2

CHARACTERIZATION OF THE COMPLETE FIBER NETWORK TOPOLOGY OF PLANAR FIBROUS TISSUES AND SCAFFOLDS

2.1 – CHARACTERIZATION OF PLANAR FIBROUS TISSUES AND SCAFFOLDS TOPOLOGY: THE STATE OF THE ART

Investigating how scaffold architecture affects cell morphology [122-124], predicts mechanical behaviors [116, 125-133], and serves as a basis for improved fabrication and design strategies [56, 134-135], have all become crucial goals in the development of engineered tissue scaffolds. A mechanistic understanding of the connection between fabrication methods, micro-architecture, and micro- and macro-mechanical behavior could lead to a generation of better performing tissue engineered constructs [136]. Objective and automated extraction of fiber architectural features such as diameter, intersections, network connectivity, and orientation distribution provide the ability to assess the influence of specific manufacturing parameters on the material geometry and to correlate scaffold structural parameters with cell morphology, metabolism and phenotypic expression [122-124]. Similarly, structural based approaches that are either statistical [47, 50, 55] or deterministic [116, 125-132] need to be supplied with accurate descriptions of the fibrous scaffold architecture. The benefit of comparing microstructure with mechanical properties of healthy and pathological

tissues using objective and automatic algorithms in place of manual procedures has been extensively demonstrated [137-138].

Current imaging techniques aiming to quantify fiber architecture of native and engineered tissues include light scattering [47, 139], scanning electron, and multi-photon microscopy and optical coherence tomography [140]. In order to process this information different image analysis paradigms such as Hough transforms, intensity gradient based texture analysis algorithms [55, 138, 141], direct tracking methods [142-144], and fast Fourier transform (FFT) based image analysis [145-147] have been successfully implemented. However, regardless of the imaging technique adopted as a data source, the totality of the mentioned image analysis methodologies share the limit of not offering a complete description of the fiber network topology. Fiber intersections and fiber connectivity, both relevant parameters for scaffold mechanical modeling [116, 125-132] and structural characterization, are not quantified by currently available techniques. In addition, objectivity and speed of analysis are important characteristics associated with an automated algorithm that are difficult or impossible to achieve with human operator-based analysis.

In the present study an image-based analysis approach that provides an automatic tool to characterize engineered tissue fiber network topology was developed. A complete set of fiber network descriptors were collected from standard SEM images including: fiber angle distribution, connectivity, intersection spatial density, and diameter. In order to demonstrate the potential of this approach a synthetic polymer system was initially evaluated: electrospun

poly(ester urethane)urea (ES-PEUU) scaffolds. Electrospun scaffolds were chosen for their recognized capability to recapitulate native soft tissue mechanical behavior [55, 148-149]. In addition, analyses on rabbit mesenchymal stem cell (MSC) seeded collagen gel scaffolds and decellularized rat carotid arteries provided further evidence of the applicability and flexibility of the presented methodology.

2.2 – A NOVEL METHODOLOGY, THE ALGORITHM DESCRIPTION

2.2 .1 – MATERIAL FABRICATION AND IMAGING

Detailed descriptions of the polymer synthesis, scaffold fabrication and imaging have been previously presented [55, 69, 150-151]. Briefly, ES-PEUU scaffolds were fabricated and imaged using the procedure described previously by Courtney et al. [55]. Briefly, PEUU was synthesized from polycaprolactone diol and 1,4-diisocyanatobutane with subsequent chain extension by putrescine. Scaffolds were fabricated by dissolving PEUU in hexafluoroisopropanol at 12% (w/v) followed by electrospinning onto a cylindrical stainless steel mandrel with a diameter of 11.43 cm (solution flow 1.0 mL/h, ΔV 19kV, injector-target distance 13 cm) using three different mandrel tangential velocity velocities: 1.5, 4.5, 9.0 m/s. Next, (10-mm x 10 mm) ES-PEUU specimens were excised from intact ES-PEUU sheets with the known circumferential axis of the collection mandrel parallel to the y-axis of the device. For imaging, each ES-PEUU

specimen was sputter coated with Pd/Au and imaged (grayscale, 8-bit) using a standard SEM (JEOL JSM6330F). The specimens were imaged at 3500x and 2500x magnification for the 1.5, 4.5 and 9.0 m/s scaffolds respectively. Nine images were selected from random locations of each sample. Rabbit MSC seeded collagen gel scaffolds were produced adopting a protocol described in [152] and the same imaging protocol was used for these scaffolds. Decellularized rat carotid artery SEM images were provided by Liao et al. [153].

2.2 .2 – IMAGE PRE-PROCESSING

A custom image analysis algorithm was developed with MATLAB (The MathWorks, Natick, MA). The algorithm scripts are presented in Appendix A). The full code was copyrighted and it is now partially property of the University of Pittsburgh. Image histogram equalization followed by 3 by 3 median filtering was operated on the starting image (Fig. 1-a) to increase the contrast, reduce noise and preserve structure edges (Fig. 1-b) [154-155]. Local thresholding was performed to separate the outer fiber network from the background [144, 155-156]. The original image was divided into sub-images each with image length equal to 10 times the representative fiber diameter (RFD), which was manually identified by the operator selecting the fiber diameter borders on the original SEM image. The local thresholding approach overcame local intensity unevenness of the SEM images, which can compromise the quality of global thresholding segmentation. Approximate RFD detection was required to identify the characteristic scale length of the system and to conduct the original image

subdivision. Grey level unevenness in the SEM images involved areas bigger than the sub-image size identified with the described procedure. The selection of appropriate thresholds for each sub-image was automatically determined using the Otsu method [157]. The latter assumes that the image is composed of two classes of pixels (e.g. foreground and background) then calculates the optimal thresholding value to divide those two classes so that their combined intra-class variance is minimal (Fig. 1-c) [155-157].

2.2 .3 – FIBER NETWORK DETECTION

Thinning, smoothing, and removal of isolated pixel areas not associated with scaffold fibers was performed on the thresholded image using the MATLAB function (MF) *bwmorph* (*thin, majority, clean*). A sequence of morphological operations consisting of erosion (through MFs *imerode,* disk element, size RFD/6), elimination of pixel areas smaller than 200xRFD (MF function *bwareopen*), dilation (MF *imdilate,* disk element, size RFD/3), and an additional erosion (MF *imerode* disk element, size 1/6xRFD) operation served to refine the image, highlighting fiber edges and eliminating isolated pixel areas [155]. The sizes of the eroding/dilating elements (erode 1/6 RFD, dilate 1/3xRFD, erode 1/6xRFD) were chosen in order to not alter the size of the fibers (Fig. 1-d). The described morphological procedures were performed to improve the precision of the skeletonization (MFs *bwmorph, skel*) [158], which is the final step carried out to identify the main direction and shape of the fibers [144, 156, 159-161]. This cascade of image processing steps (Fig.1-a-f) produced two final outcomes: (1) a

binary image of the skeletonized fiber network where every fiber has a thickness of one pixel (Fig. 1-e), (2) a binary filter (Fig. 1-f) obtained from a further operation of dilating (MF *imdilate* disk element, size 1/4xRFD) on the skeletonized fiber network.

In the next processing step, fiber intersections were detected within the skeletonized fiber network. For each pixel in the skeletonized image the pixel values within a circular corona (d_{max} 10 pixels, d_{min} 6 pixels) centered on a given pixel were collected at angle steps of 10°. Grey intensity values were radially summed and plotted vs. angle. The number of peaks in the polar plot corresponded to the number of fibers intersecting in proximity to the analyzed pixel. Pixels reporting more than 2 peaks were considered to belong to zones of fiber intersections. A new binary image composed of the totality of pixels associated with zones of fiber intersections was generated. Isolated pixels not representing fiber intersection zones were removed from the new image using MF *bwmorph, clean,* and *bwareopen.* Similarly, to merge fiber intersection areas that were in close proximity to each other, a further dilation was performed using MF *imdilate* disk element, size 2/5xRFD. The center of mass of the detected intersection areas was identified as the actual fibers intersections. An additional set of helping points (HP) were automatically detected using the Boolean intersection of two binary images: (1) the fiber network skeletonization and (2) a regular grid (grid step 3/2xRFD). Delaunay triangulation was utilized (MF *delanuay*) using the fiber intersections and the HPs as data points. Note that HPs are not considered as fiber intersections, they have been adopted as extra points in

the Delaunay triangulation to enhance the algorithm capacity in capturing fibers shapes. An additional binary filter (Fig. 1-g) was created assigning a value of 1 to pixels within a circle of radius RFD centered in all fiber intersections and HPs. The Delanuay triangulation was consequently modified using the two binary filters described. All segments in the Delanuay network were subjected to the following condition: if the segment connecting two points (either fiber intersections or HPs) had 80% of its points within the first filter (Fig. 1-f) and if the connecting segment did not hit more than 2 non zeros circles in the second filter (Fig. 1-g), the segment was considered a fiber segment otherwise it was deleted from the Delaunay network. The two described conditions reduced the over-connection of the Delaunay network preserving only the fiber segments that corresponded to real fibers (Fig. 1-h).

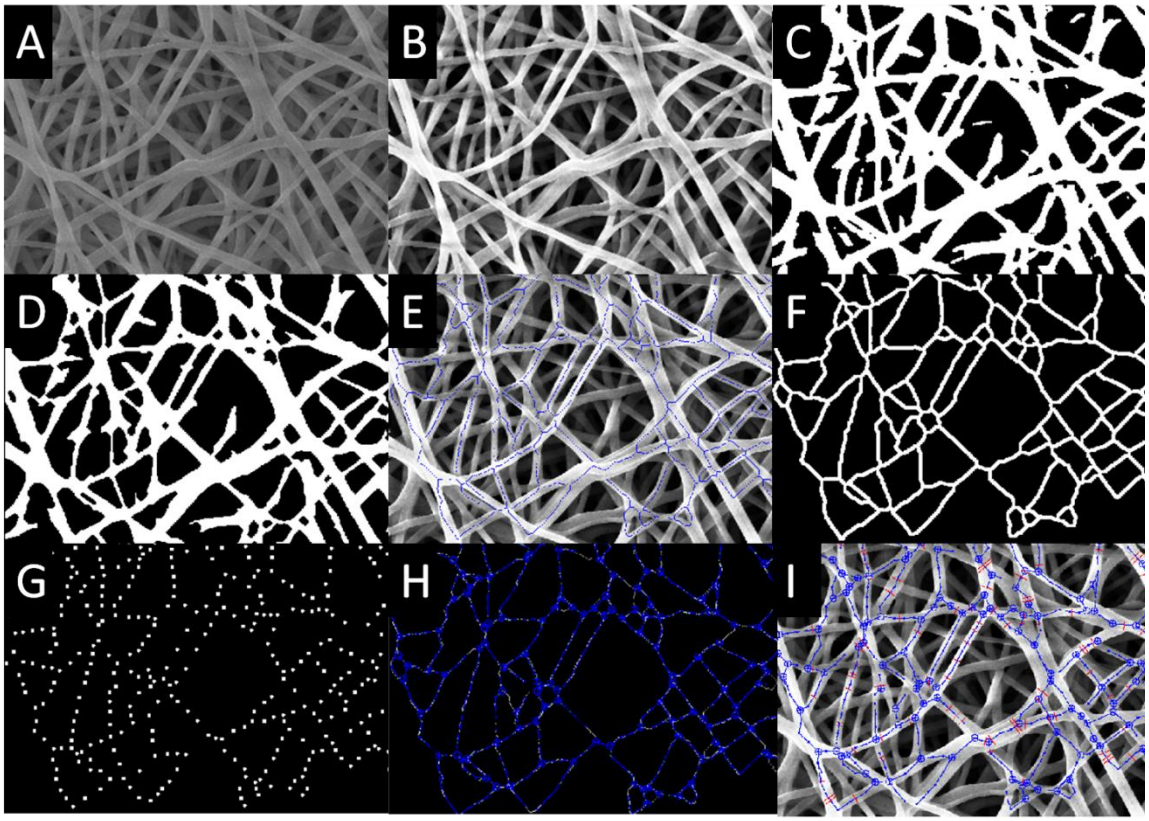

Figure 1. A) Starting SEM image. B) Image histogram equalization followed by 3 by 3 median filtering. C) Local thresholding through Otsu method. D) Thinning, smoothing and removal of isolated pixel areas through a cascade of different morphological operators. E) Skeletonization. F-G) binary filters for Delaunay network refinement. H) Modified Delaunay network associated to the real fiber network. I) Final network and fiber diameters detected.

2.2 .4 – FIBER DIAMETER DETECTION

Fiber diameter was evaluated for each network segment along a direction perpendicular to the segment direction in correspondence with three points: the intermediate point, plus two additional points shifted of ± 5 pixels along the

segment direction. In order to find the diameter edges, grey intensity levels of pixels were collected in an array along the described directions. A representative maximum fiber diameter was manually detected by the operator to limit the length of the analyzed segments along the direction perpendicular to the fiber. The positions of five points were identified from the detected grey levels array corresponding to: the maximum of grayscale gradient (Fig. 2, point A), the maximum of the sum of the grayscale array and its gradient (Fig. 2 point A), the maximum grayscale value (Fig. 2, point C), the minimum of the sum of the grayscale array and its gradient (Fig. 2 point D), and the minimum of the grayscale gradient (Fig. 2 point E). Points (A, E) corresponded to the edges of the fiber. The points (B, D) either coincided with point (A, E) in a single fiber or corresponded to the first/second edge of the most external fiber inside a fiber bundle.

Point (C) represented the brightest point within the analyzed searching line. These positions were used to automatically select fibers where the diameter detection can be conducted without ambiguity and to identify a single fiber diameter within a fiber bundle. This condition was verified when points A-E stayed in the described order. Only the fibers respecting this condition were considered to determine the fiber diameter distribution. The maximum length value among the detected segments A-B, B-C, C-D, D-E was considered to be the fiber diameter.

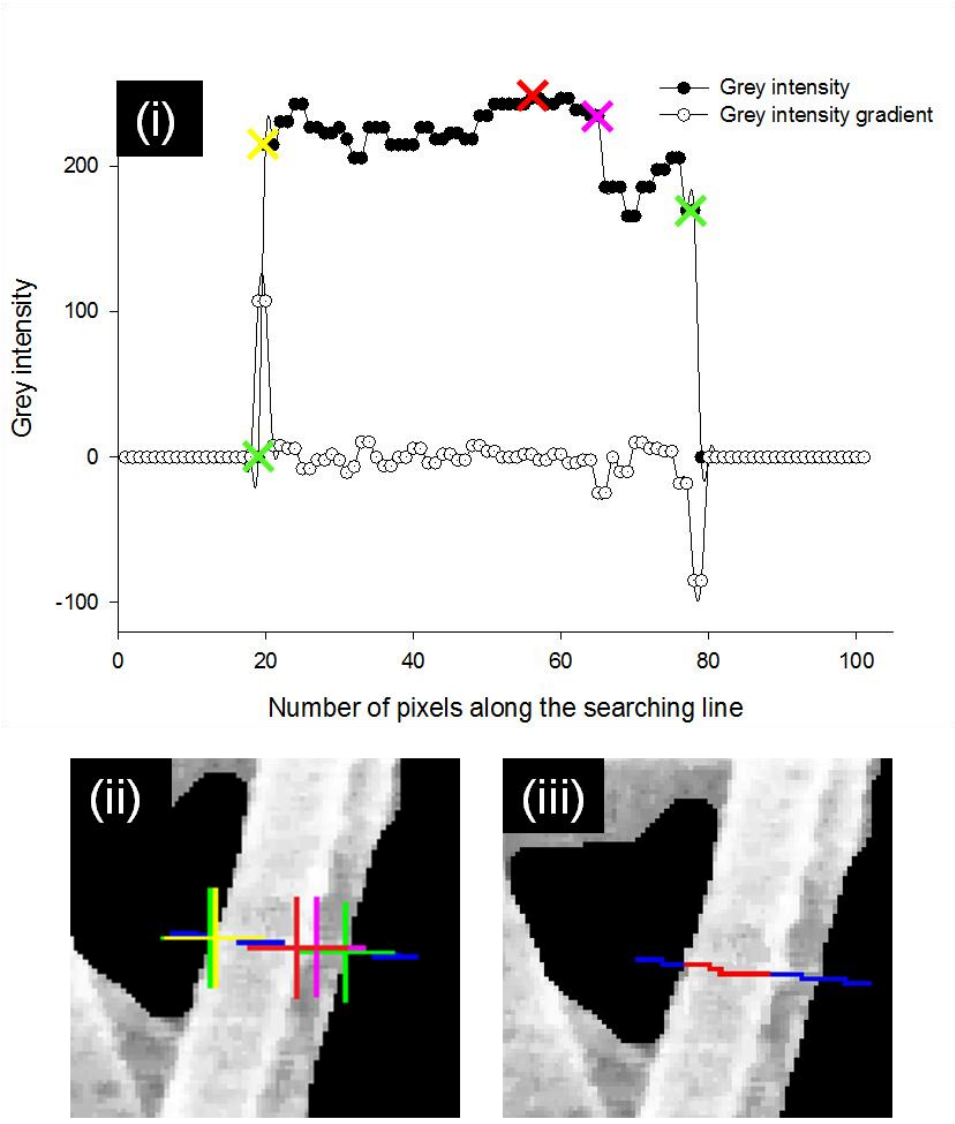

Figure 2. Fiber diameter detection. Five points are detected from detected grey levels and its gradient (i) along the searching line (blue segment in (ii)): Point A (ii) maximum of the grey levels array gradient. Point B (ii) maximum of the sum of the grey levels array and its gradient. Point C (ii) the maximum grey levels value. Point D (ii) minimum of the sum of the grey levels array and its gradient. Point E (ii) minimum of the grey levels array gradient. Searching line ((ii)-(iii) blue segment) and detected diameter ((iii) red segment).

2.2 .5 – EXTRACTION OF ARCHITECTURAL DESCRIPTORS AND VALIDATION

To summarize, the described algorithm extracts the following data from the generated network (Fig. 1-i):

(1) the number, spatial position, and density of the fiber intersections, defined as two or more overlaying fibers;

(2) the connectivity distribution, defined as the percentage of fiber intersections vs. number of fibers crossing a fiber intersection;

(3) fiber orientation angle distribution;

(4) fiber diameter distribution.

This data enables the scaffold fiber network to be uniquely and fully described. In addition, comparisons were made using the two most adopted fiber orientation indices: the average over all fiber segments of $cos^2(\theta)$ (COS OI), where θ represents the angle between a fiber segment and the direction of supposed alignment) [116, 162] and the normalized angle (NOI) that contains one-half of the total area under fiber count-angle distribution, representing 50% of the total number of fibers (OI) [163-164]. NOI it is obtained from OI using the relation (Eq. 1):

$$NOI = \frac{90-OI}{90} \times 100\% \qquad \text{(Eq. 1)}$$

so that NOI $\in$ [0,100%].

The image analysis software presented in this study was evaluated on the following fibrous structures:

1. ES-PEUU scaffolds that were fabricated at three mandrel tangential velocities (1.5, 4.5, 9.0 m/s) providing increasing levels of structural anisotropy [55, 165];
2. Rabbit MSC seeded collagen gel [152];
3. Decellularized rat carotid arteries [153].

Algorithm performance was first tested using phantom images and actual sample images of ES-PEUU scaffolds [137]. Phantom images were obtained by merging together subsets of original SEM images and adding random noise (Fig. 3). In addition to the adoption of validation phantoms, scaffold architectural features were manually detected (n=5 individuals blinded from the others results).

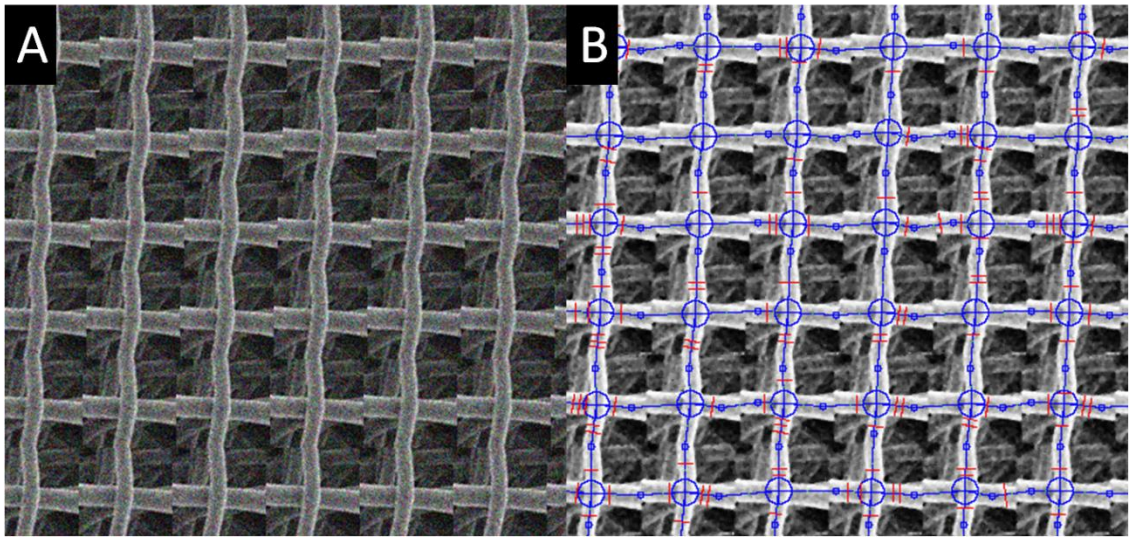

Figure 3. A) Phantom image obtained merging together subsets of original SEM images and adding random noise. B) Network and fiber diameters detected by the algorithm.

A custom made interactive procedure was developed for validation. Operators manually selected fiber intersections, HPs, and fiber diameters. Fiber segments were automatically generated with their orientation dictated by the fiber intersections and HP positions. The automated algorithm results and manually generated results were qualitatively and quantitatively compared. A coefficient of determination (R^2 value) was calculated for fiber angle and fiber connectivity distribution. Comparison of the manually detected and algorithm detected results were presented as mean ± standard deviation. The combination of the full set of variables: (1) density of the fibers intersections, (2) connectivity distribution, (3) fiber orientation angle distribution, and (4) fiber diameter distribution gives an unambiguous representation of a straight fiber network. For this reason, the validation study was conducted over the full set of architectural descriptors.

2.3 – IMAGE ANALYSIS RESULTS

The analyses of phantom images using the presented algorithm demonstrated that fiber orientation, intersections, connectivity and diameter in the image were correctly evaluated (Fig. 3). In addition to the use of phantom images, our custom software was validated using ES-PEUU scaffold SEM images (2 images were randomly selected from the 1.5 m/s and the 9.0 m/s processed groups). The validation results (Fig. 4) demonstrated that manual and automated methods were

comparable (Fig. 5-7). Coefficient of variation for the 9.0 m/s scaffold fiber angle distribution and for the connectivity distribution (Fig. 6-a-b) was 0.86 and 0.93 respectively.

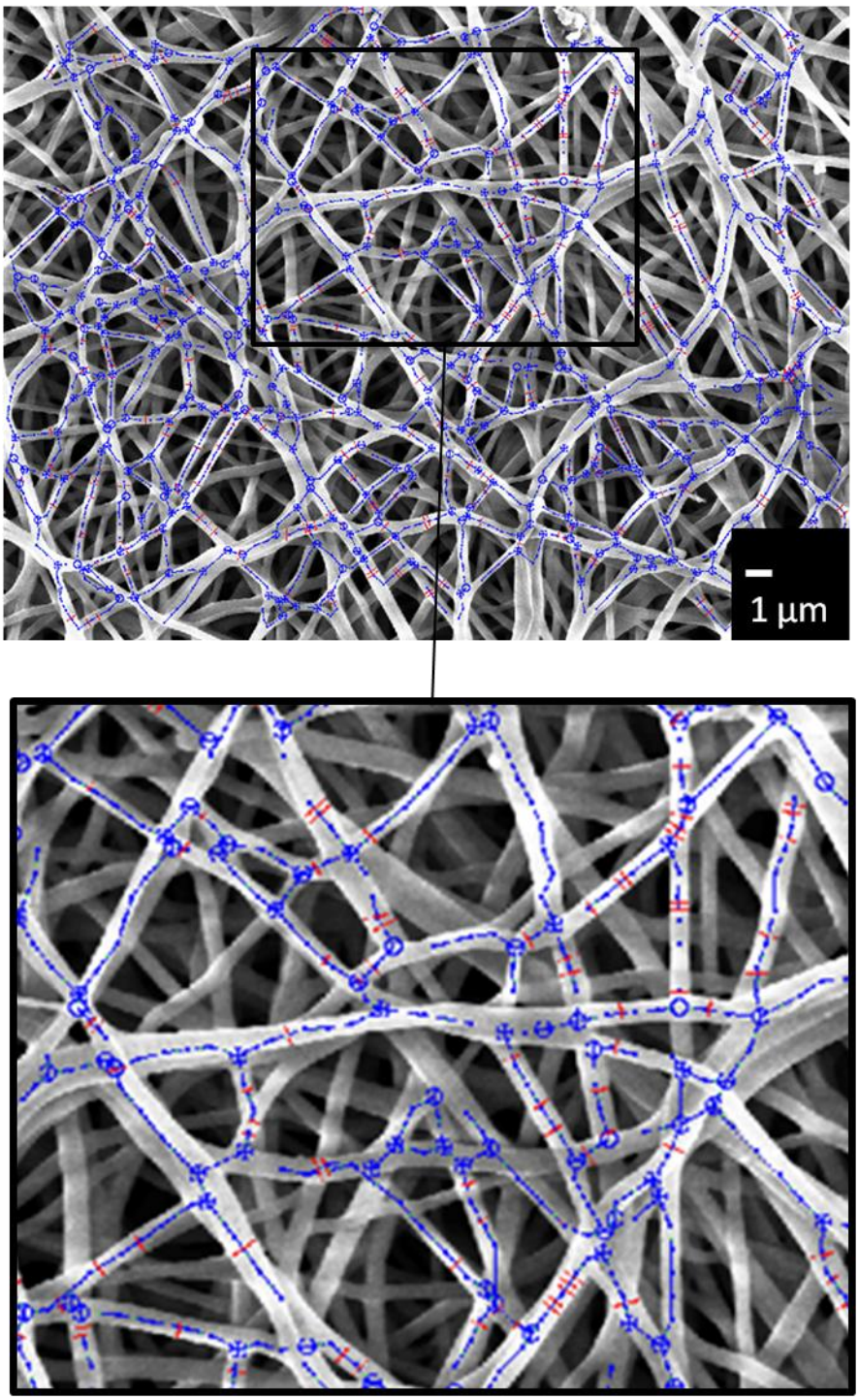

Figure 4. Example of analysis on 1.5 [m/s] scaffold. Circles indicate fiber intersections, blue dots the "helping points", blue segments represent the scaffold fibers and red segments show the fiber diameters.

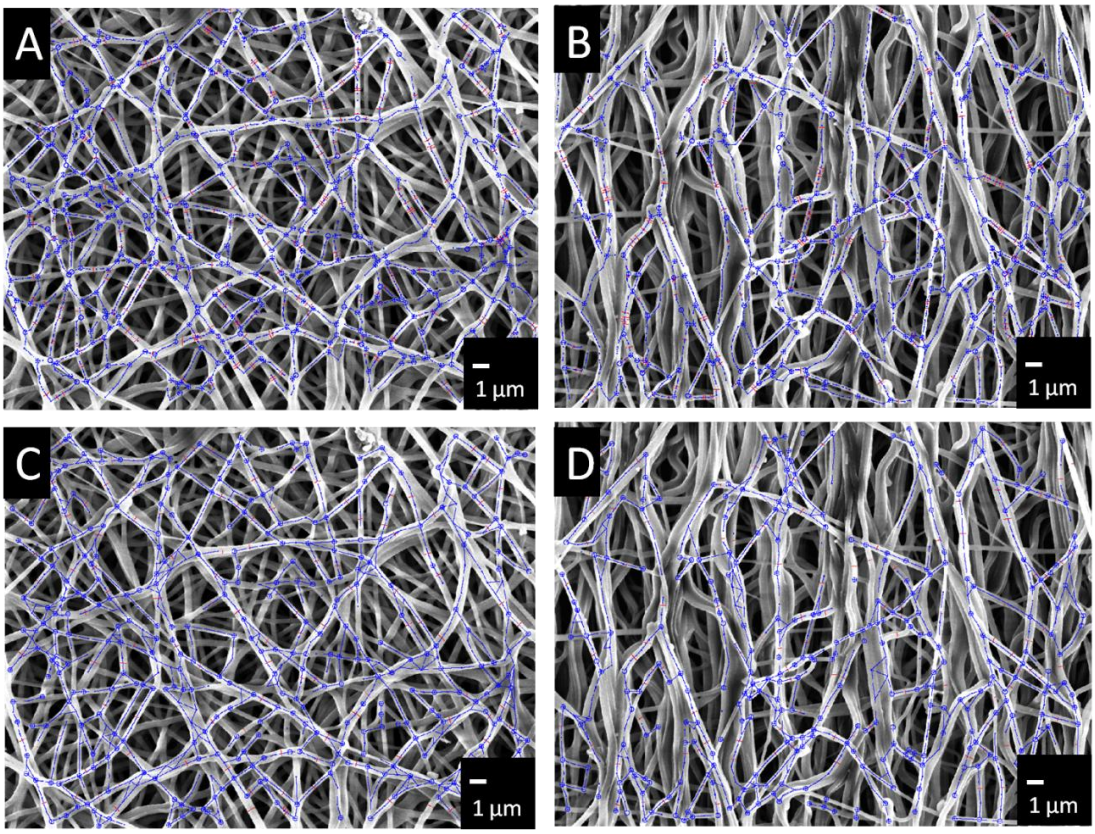

Figure 5. Fiber network and diameters detected by the algorithm, A) 1.5 m/s scaffold, B) 9.0 m/s scaffold. Fiber network and diameters manually detected, C) 1.5 m/s scaffold, D) 9.0 m/s scaffold.

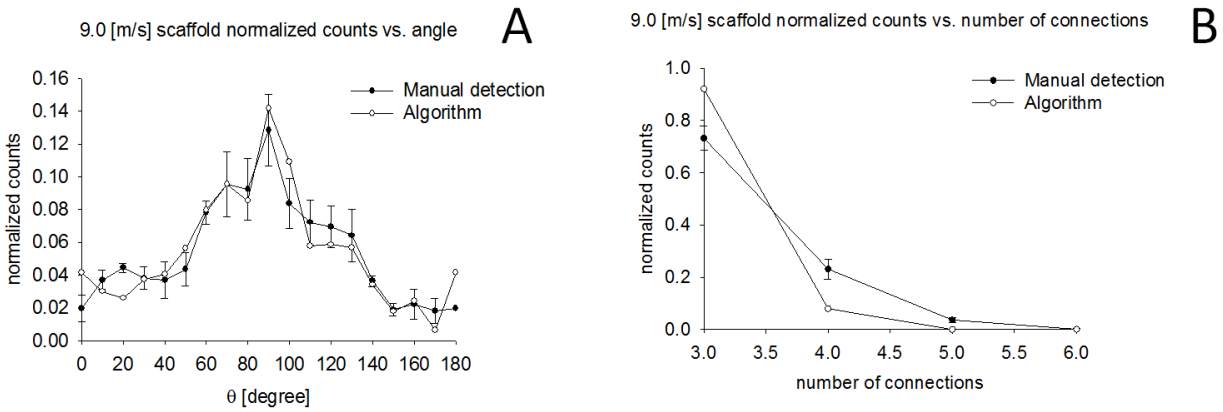

Figure 6. A) Fiber angle distribution measurements comparison 9.0 [m/s] scaffold. B) Fiber connectivity measurements comparison 9.0 [m/s] scaffold.

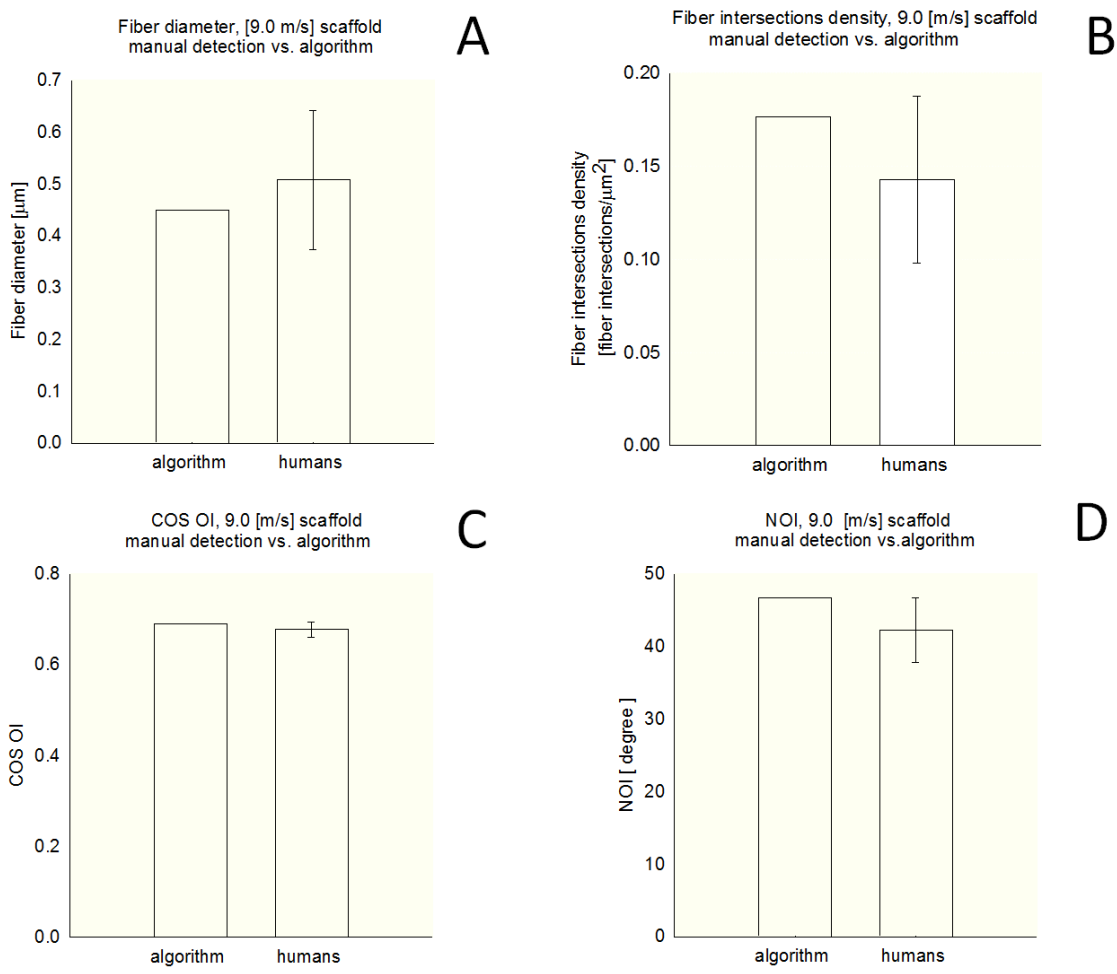

Figure 7. 9.0 Manual detection vs. algorithm measurements comparison for the 9.0 [m/s] scaffold. A) Fiber diameter measurements comparison. B) Fiber intersections density measurements comparison, C) COS OI measurements comparison, D) NOI measurements comparison.

2.3.1 – APPLICATION FOR ES-PEUU SCAFFOLDS

Nine randomly chosen images were selected for each of the three scaffold groups. The number of fiber intersections decreased as the mandrel velocity was raised, showing consistency with the known relationship between the mandrel velocity and the fiber alignment (Fig. 9-a) [55, 145-146]. Connectivity distribution was not affected by the mandrel velocity (Fig. 8-b). COS OI and fiber alignment along the circumferential direction increased as the mandrel velocity was raised (Fig. 9-c, Fig. 8-a). Consistent with the COS OI trend, the NOI decreased as the mandrel velocity was raised (Fig. 9-c-d). The fiber diameter followed a similar trend (Fig.9-b).

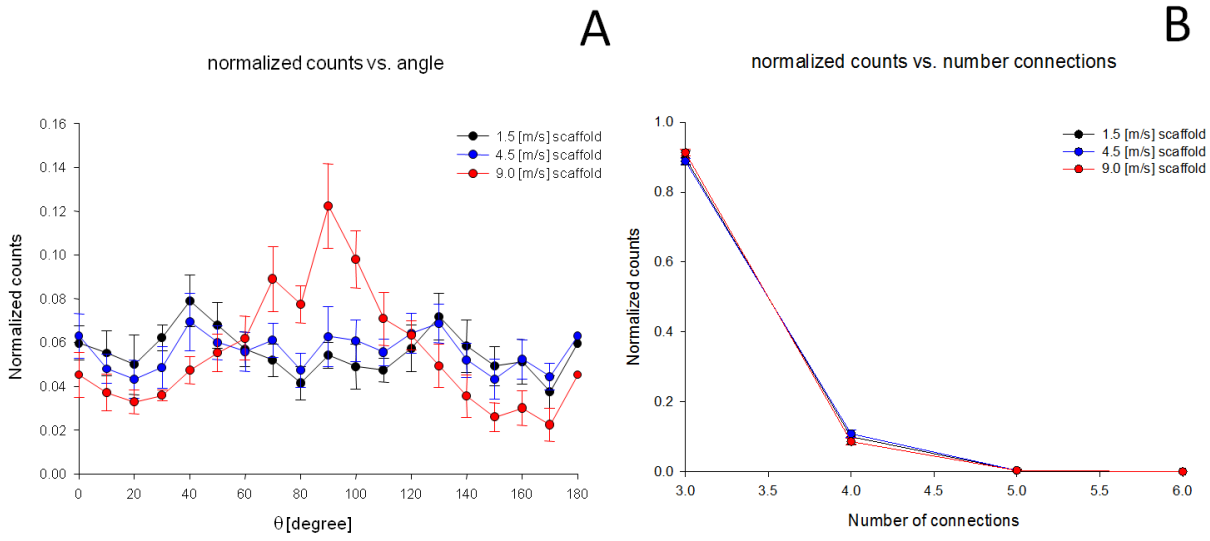

Figure 8. A) Fiber angle distribution for the 1.5, 4.5, 9.0 [m/s] scaffolds. B) Fiber connectivity distribution for the 1.5, 4.5, 9.0 [m/s] scaffolds.

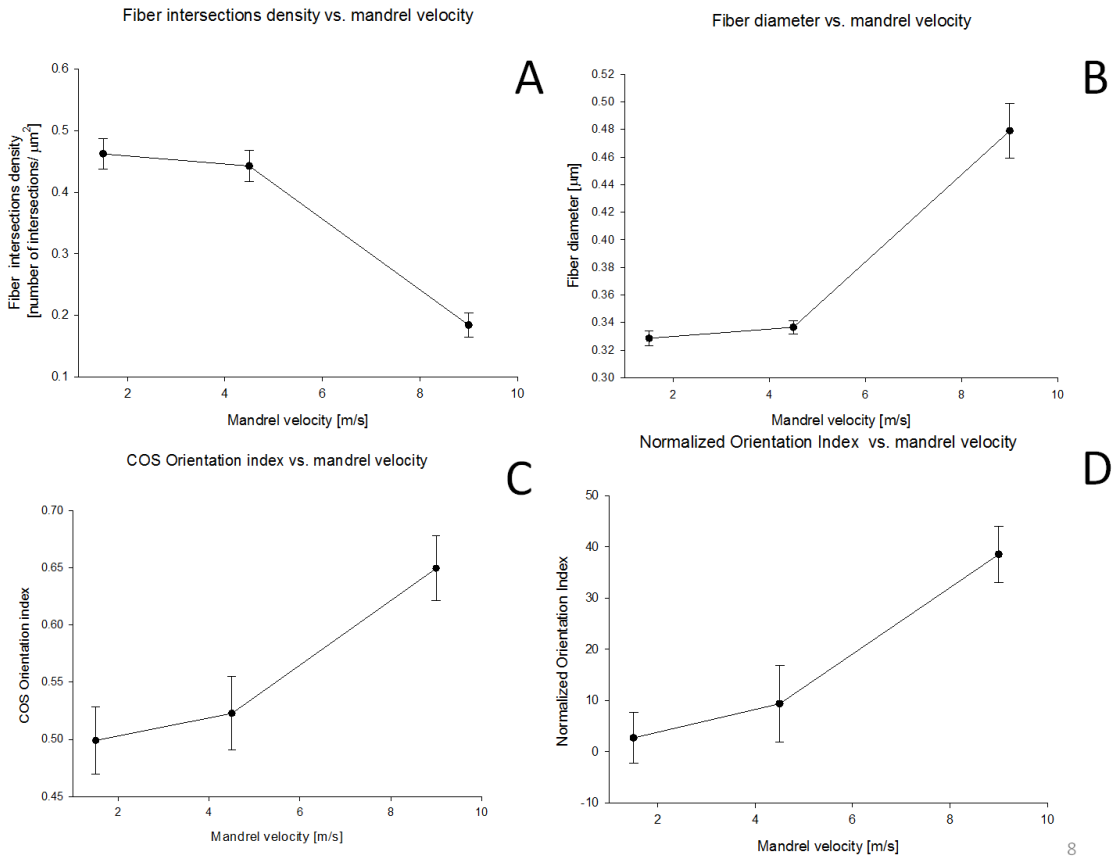

Figure 9. A) Fiber intersections density vs. mandrel velocity. B) diameter vs. mandrel velocity. C) COS OI 1 vs. mandrel velocity. D) NOI vs. mandrel velocity.

2.3.2 – APPLICATION TO COLLAGEN GEL AND DECELLULARIZED TISSUES

Algorithm accuracy in capturing the fiber network structure in rabbit MSC seeded collagen gel scaffolds and decellularized rat carotid arteries was demonstrated by the visual comparison between the actual fiber networks with the detected ones (Fig. 10-a-b). The presented methodology revealed key differences in network morphology between the different scaffolds typology. As expected, collagen gel and decellularized tissues showed a more tortuous fiber topology when compared with electrospun constructs (Fig. 11). The adopted collagen gel and decellularized tissues samples, both isotropic in nature, reported fiber orientation angle distribution characterized by COS OI values of 0.49 and 0.44 respectively. This result was consistent with the COS OI value found for the 1.5 m/s ES-PEUU scaffold which is a well characterized mechanically and structurally isotropic scaffold [55]. The density of the fiber intersections in the collagen gel and the decellularized tissue samples were 17.69 intersections/μm^2 and 7.45 intersections/μm^2 respectively, these values were greater than the correspondent values detected for the ES-PEEU scaffolds (Fig. 9-a). Differences in the determined average fiber diameter of decellularized tissue (0.045 μm), collagen gel (0.043 μm), and ES-PEUU scaffolds (0.328 - 0.479 μm, Fig.9-b) provided further evidence of the different characteristic length of the fibrous system (Fig. 11).

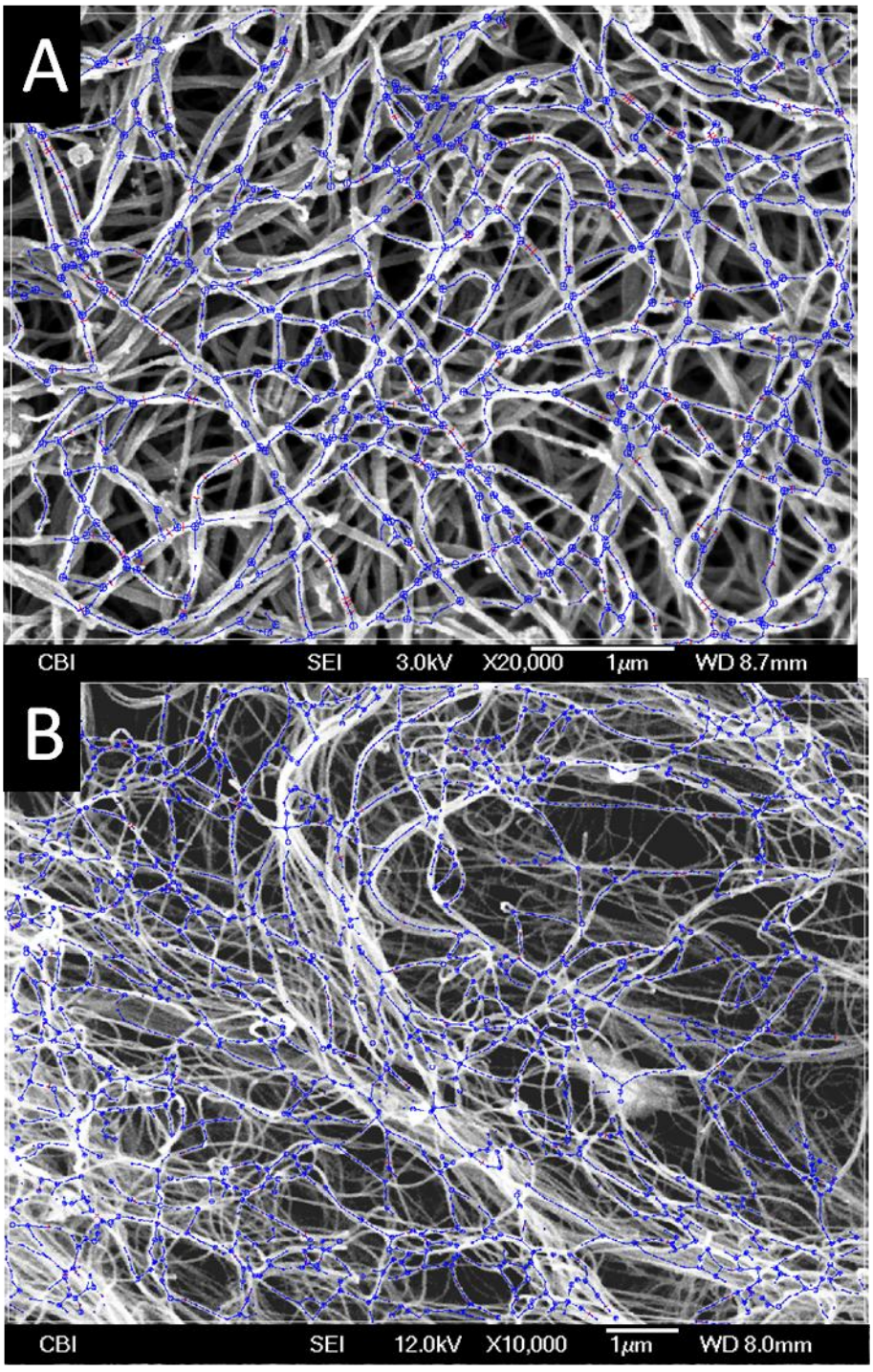

Figure 10. A) Rabbit MSC seeded collagen gel analysis, detected fiber network and fiber diameters. B) Decellularized rat carotid arteries analysis, detected fiber network and fiber diameters.

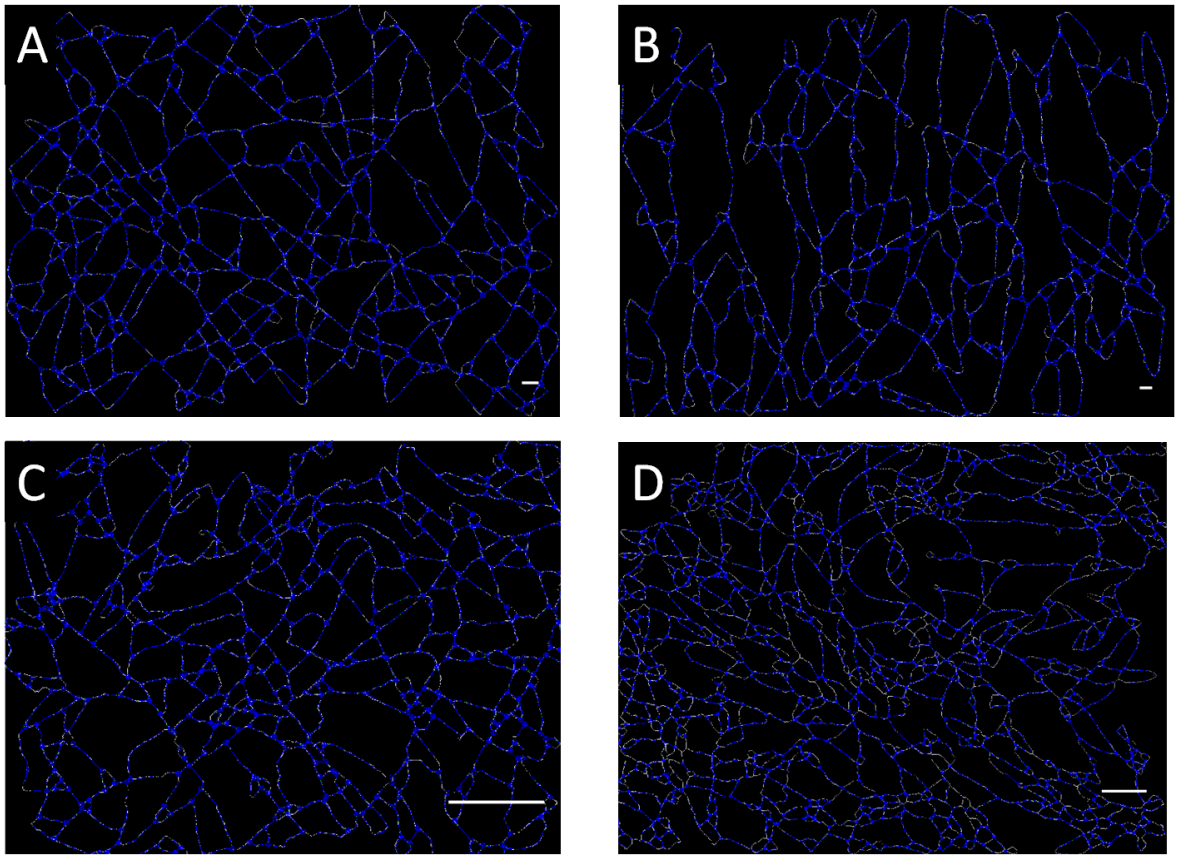

Figure 11. Detected fiber networks for A) PEUU 1.5 [m/s] scaffold, B) PEUU 9.0 [m/s] scaffold, C) Rabbit MSC seeded collagen gel, D) Decellularized rat carotid arteries.

2.4. – THE PRESENTED IMAGE ANALYSIS TECHNIQUE: ADVANTAGES AND LIMITATIONS

A methodology was presented to characterize the complete fiber network topology of planar fibrous scaffolds or tissues (both native and engineered). Validation results illustrated the capability of the algorithm to accurately identify

fiber network morphological parameters at a level similar to that of human operators (Fig. 5). Detected fiber diameter, fiber intersections density and orientation index values matched well (Fig. 7-a, b, c, d) with human operators. Fiber connectivity distribution and fiber angle distribution coefficients of variation for the 9.0 m/s electrospun, scaffold were 0.86 and 0.93 respectively. As expected a higher degree of fiber alignment produced a smaller number of fiber intersections (Fig. 9-a). The two different orientation indices showed consistency with the known fiber alignment – mandrel velocity relationship (Fig. 9-c-d). Specifically, higher values of mandrel velocity produced fiber angle distributions more aligned along the mandrel circumferential axis (90°, Fig. 5, Fig. 8-a) and increasing fiber diameter values (Fig.9-b). All of the described measurements required less than 180 seconds per image and a minimal user interaction. In contrast, the manual detection procedure was subjective and involved up to 2 hours per image. The algorithm capability to fully capture fiber network morphology regardless of the scaffold type (Fig. 10-a-b) and structure scale (Fig. 11) was shown for electrospun constructs, gels, and decellularized tissues.

Several imaging techniques have been previously adopted to collect scaffolds and native tissue micro structural data. Small angle light scattering (SALS) [47, 139] can accurately measure fiber orientation up to a tissue thickness of 500 μm to an angle of ~ 1° and a spatial resolution of ± 254 μm. Multi-photon microscopy and optical coherence tomography are emerging, nondestructive, real- time techniques with great potential for tissue biology and tissue

engineering [140]. The persisting caveat with these techniques is a lack of image resolution for micro-architecture quantification. Similarly, different image analysis methods have been developed to address the crucial issue of quantifying scaffolds and native tissue microstructure. Karlon and coauthors [138] used texture analysis algorithms based on the Hough transform [141] and intensity gradient techniques on histologic tissue sections from normal and transgenic mice. This approach was adopted [55] using scanning electron microscopy data of ES-PEUU scaffolds with different levels of fiber alignment imposed by varying the collecting mandrel tangential velocity. A different methodology based on direct tracking was successfully implemented and tested on simulated and real non-woven fabrics [142-144].

Ayres and coauthors were the first to introduce the use of the FFT to measure fiber alignment in electrospun constructs [145-147]. The method is applicable to SEM, light microscopy and confocal laser scanning microscopy. The latter imaging technique offers the clear benefit of capturing data from specific optical planes with depths of 30-40 μm in the z-direction normally achieved [147]. For example, this versatile approach was employed to assess the influence of a poly(glycerol sebacate) (PGS) scaffold microstructure on heart cell F-actin filament orientation from confocal micrographs taken at a depth of ~30 μm [166].

The summarized methodologies only quantify fiber angle distribution and do not offer a complete description of the tissue topology [137-147, 154-157, 160-161, 167]. In contrast, the relevance of fiber intersection and connectivity information for scaffold mechanical modeling has been extensively highlighted in

[116, 125-132]. Fiber diameter distribution is generally produced by subjective and time-consuming procedures where 50-100 fiber diameters are detected manually on SEM images [123-124]. Similarly, the importance of automatic and objective fiber diameter detection has been discussed [156, 160]. Although robust and easy to implement, the FTT based methods are limited by the lack of a physically meaningful relationship between the fiber alignment index and the actual fiber angle distribution. For this reason, these measurements are usually coupled with mechanical testing data [145-147].

Reliance on SEM images, while offering high-resolution, provides limited 3D dimensional information. Employing alternative imaging techniques such as confocal or fluorescence microscopy could be adopted to investigate structural properties throughout the material thickness. Although adapting the implementation to this data source seems feasible, it should be noted that confocal microscopy based data do not provide sufficient resolution for fiber diameter determination. The representative area element (RAE) size has not been identified, however, the error produced by analyzing a portion of the sample smaller than the RAE can be quantified by the magnitude of the standard deviation. Similarly, a representative volume element (RVE) that is the three dimensional equivalent of the RAE can be defined as: a model of the material to be used to determine the corresponding effective properties for the homogenized macroscopic model. The RVE should be large enough to contain sufficient information about the microstructure in order to be representative, however it should be much smaller than the macroscopic body [168]. An analysis performed

on an area size equivalent to the RAE produces a standard deviation of approximately zero, larger standard deviation values demonstrated that the analyzed area is smaller than the RAE for the specific material and variables studied [168-170]. The latter issue represents an additional limiting factor for the state of the art [137-147, 154-157, 160-161, 167].

The image analysis algorithm developed and validated in this study overcomes the state of the art limits in: (1) extracting physically meaningful topological information, (2) providing additional levels of detail to describe fibrous networks, and (3) circumventing reliance on human operators enabling objective, automated measurements. Critical information relevant to biomechanical modeling, scaffold fabrication optimization and the study of native/engineered construct architecture-cell interaction is provided. Fiber diameter, fiber intersection density, fiber angle distribution and connectivity all play a fundamental role in the macro - meso mechanical behavior and in the fiber network – cell interactions. Mechanical response prediction strongly relies on the capability of the model to reproduce the real material characteristics and topology. The algorithm capability to automatically identify and quantify the complete fiber network morphology regardless of the scaffold topology and of the structural scale of the system was proven by analyzing three different types of scaffolds. The presented validation showed strong consistency between the human operators and the algorithm analysis results. Moreover, the automatic procedure guarantees objectivity and a significant reduction in analysis time. This work represents an attempt to fully characterized fiber network topology, to

detect fiber intersections and to quantify fiber connectivity in native and engineered tissue constructs. This method to study engineered construct morphology extends well beyond the analysis contained herein, potentially enabling the investigation of scaffold architecture effects on cell populations and resulting aspects of tissue growth and remodeling in vitro and in vivo. This methodology was utilized to characterized all the scaffolds tested and modeled in this doctoral studies.

CHAPTER 3

ELASTOMERIC ELECTROSPUN POLYEURTHANE SCAFFOLDS: THE INTERRELATIONSHIP BETWEEN FABRICATION CONDITIONS, FIBER MORPHOLOGY, AND MECHANICAL PROPERTIES

3.1 – ELECTROSPINNING OF ELASTOMERIC POLYEURTHANE SCAFFOLDS: THE STATE OF THE ART

Electrospinning has been gaining increasing popularity in the fabrication of engineered tissue scaffolds due to its ability to produce nano to micro scale fibrous mats. During the electrospinning process, electrostatic forces draw a polymer solution from a conductive capillary towards a grounded target, and if the solution is sufficiently concentrated, the polymer chains entangle and produce continuous fibers [171]. Electrostatic repulsions between different fiber sections create a series of instabilities that cause the fiber to whip violently, stretching it further until it reaches the target [172]. The resulting process, in its simplest form, yields a fiber network with complex, unpredictable topologies. Many investigators have attempted to apply various degrees of control to this process in order to produce fiber meshes with more predictable patterns. These attempts have largely been limited to controlling fiber alignment and have fallen into two categories: physical manipulation of the fibers by pulling them into

alignment using a rapidly spinning mandrel [55, 173-175] or manipulation of the electric field during fabrication [176-178].

To more fully quantify the structure of native and engineered tissue and scaffold fibrous networks, a novel image analysis technique has been recently introduced [179]. This technique, illustrated in the previous chapter is capable of transforming a scanning electron microscopy (SEM) micrograph into a modified Delauney network that matches the outer layer of electrospun fibers. From this network, it is possible to extract (Figure 1): (1) fiber overlap number, position and density, (2) connectivity distribution, and (3) fiber angle distribution. This novel technique has allowed us to obtain a more complete picture of the structural complexities of planar electrospun scaffolds ([179]).

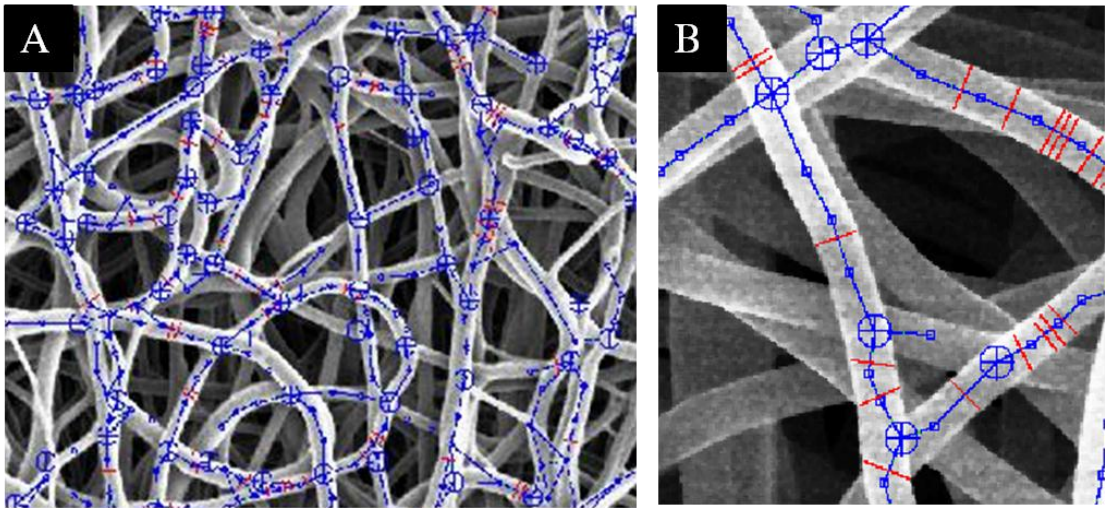

Figure 1. Examples of the analysis performed by the developed algorithm, a network is associated automatically to the actual fiber network. Fiber overlap density, fiber angle distribution, connectivity and fiber diameter were detected.

In the present work, we utilized this image analysis technique in conjunction with comprehensive mechanical characterization (Figure 2) to advance our understanding and control of the electrospinning process.

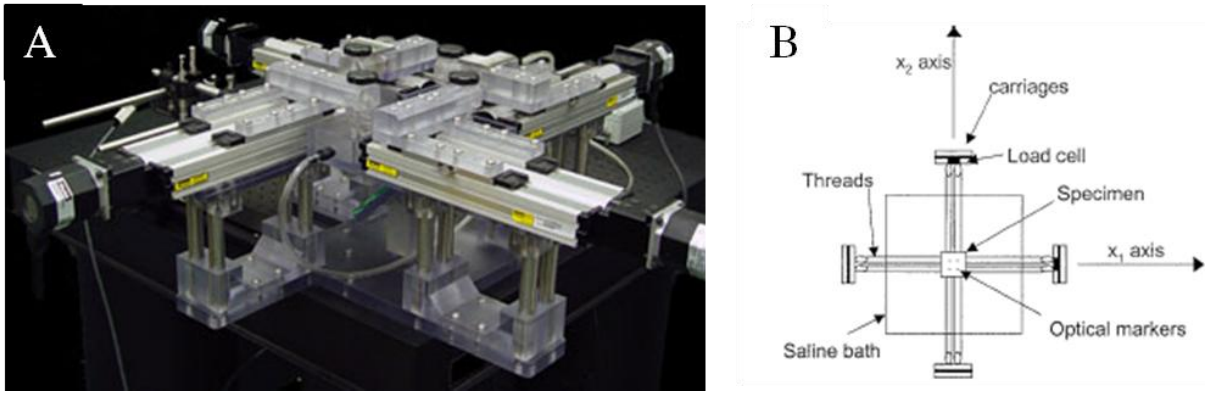

Figure 2. Schematic of the adopted biaxial testing device.

Two previously unevaluated fabrication modalities were investigated in order to more deeply explore how the fabrication process can alter the structural and mechanical response: (A) variation of mandrel translation velocity and (B) concurrent electrospraying of cell culture medium with or without cells or rigid particulates. These modalities were evaluated both independently and in conjunction with one another to gain deeper insight into how electrospun scaffold structure relates to bulk mechanical response, as well as determining any dependent or synergistic relationships (Figure 3). These fabrication parameters were chosen to introduce additional degrees of freedom and enrich control in the electrospinning process.

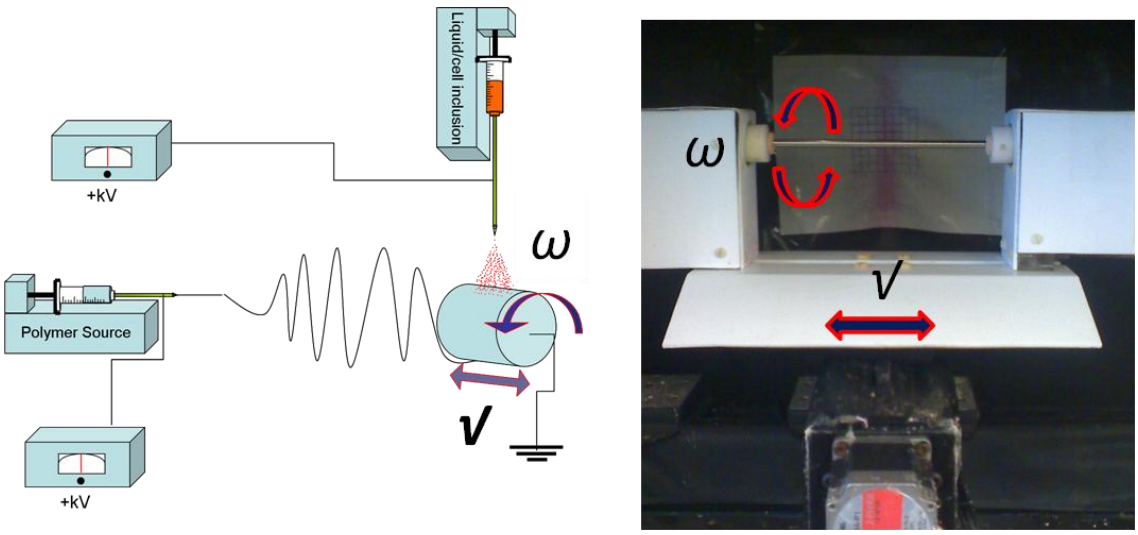

Figure 3. Schematic of the adopted two streams electrospinning device. Polymer deposition and cell spraying can be operated simultaneously. √ rastering velocity (linear velocity along the longitudinal axis), ω mandrel angular velocity (angular velocity around the circumferential axis).

3.2 – ELECTROSPINNING OF ELASTOMERIC POLYEURTHANE SCAFFOLDS: MATERIAL FABRICATION

Vascular smooth muscle cells isolated from Lewis rat aortas were expanded on tissue culture polystyrene culture flasks using Dulbecco's modified Eagle medium (DMEM) (Lonza) supplemented with 10% fetal bovine serum and 1% penicillin-streptomycin[180]. Poly(ester urethane) urea (PEUU) was synthesized as described previously [150] from polycaprolactone diol (Mn=2000, Sigma), 1,4-diisocyanatobutane (Sigma) and putrescine (Sigma). The current study focuses on evaluating the effects of several electrospinning process modifications and variables described previously [60, 181]. Briefly, a 12% (w/v) solution of PEUU in 1,1,1,3,3,3-

hexafluoroisopropanol (Oakwood Products, Inc) was fed at 1.5 mL/h through a stainless steel capillary (inner diameter: 1.2 mm) charged to 11 kV and located 17 cm horizontally from a stainless steel cylindrical mandrel. (Figure 1a) The mandrel was 6 mm in diameter, charged to -4 kV, and rotated at 266 rpm (~8 cm/s tangential velocity). The mandrel was rastered upon its rotational axis at 0.3, 1.5, 3 or 30 cm/s. PEUU was electrospun dry (with no further modifications to the process) or "wet" by concurrently electrospraying cell culture medium (fed at 15 mL/hr, charged to 8kV) onto the target from a perpendicular orientation to the polymer stream 4.5 cm above the mandrel. The effect of cell and particulate inclusion into the fiber scaffold matrix was studied by electrospraying the same medium containing a suspension of a known concentration of either smooth muscle cells or polystyrene microspheres with 10 micron diameter (Invitrogen) into the cell culture medium. Cells were electrosprayed at concentrations of 2 and 6 million/mL, and microspheres were electrosprayed at 7 million/mL. The mass fraction of polymer in each scaffold was determined by first rinsing sections of known dimensions in distilled water five times and allowing them to dry at room temperature in a desiccator over 48 h. Polymer mass fraction was computed by dividing the mass of the electrospun sample by a sample of cast PEUU of identical dimensions.

3.3 – ELECTROSPINNING OF ELASTOMERIC POLYEURTHANE SCAFFOLDS: MATERIAL CHARACTERIZATION

Prior to imaging, any scaffold sections that had come into contact with a salt solution were rinsed 5 times in distilled water to remove any remnant salt crystals. After drying over 24 h the scaffold sections were gold sputter coated and imaged with SEM (JEOL JSM6330F). Sample orientation with respect to the original mandrel axes was recorded. Sets of five images at ~1000x magnification were considered for each sample. This magnification value was adopted to achieve a resolution of approximately 20-30 pixels per fiber diameter, which was considered an optimal value for fiber architecture quantification.

Fiber structural features were quantified from these images using a method previously described [179]. Briefly, the outer layer of fibers was isolated using a combination of threshold segmentation followed by morphological procedures of eroding and dilating. Fiber overlaps were counted manually, and a modified Delaunay network was generated from these overlap coordinates. The following micro-architectural data were extracted from the generated network (Figure 1): (1) fiber overlap number, position and density, (2) connectivity distribution, and (3) fiber angle distribution. Fiber connectivity was normalized to fiber diameter for comparisons, as variations in fiber size produce a scaling effect. Fiber angle distribution was further described by calculating the fiber orientation index, which was defined as: the average over all fiber segments of $\cos^2(\theta)$, where θ represents the angle between a fiber segment and the direction of alignment [116, 182].

Following fabrication, constructs were allowed to sit in cell culture medium overnight at 37^{o}C. Samples were divided into 10 mm x 10 mm sections for testing. Slices of polypropylene suture (Ethicon) were affixed to each section to form four small markers of ~1 mm in diameter in the central region. Samples were then tested in a physiological saline solution at room temperature using a Lagrangian membrane tension (T) controlled protocol as previously described [183]. Equi-biaxial tension was imposed up to a maximum of 90 N/m to facilitate comparison with previous studies on valvular tissues [55, 184-185], a major application of our lab. Data post-processing was completed using a preconditioned free-float reference, and was converted to stresses using measured specimen dimensions. Unless mentioned otherwise, data are shown as mean +/- standard error. For each group studied, five independent specimens were fabricated separately to define n=5 for all statistical analyses. Significance was determined using one way ANOVA with $\alpha = 0.05$. Post-hoc testing was performed using the Holm-Bonferroni method.

3.4 - ELECTROSPINNING OF ELASTOMERIC POLYEURTHANE SCAFFOLDS: FABRICATION STUDY

3.4.1- EFFECT OF RASTERING ON DRY ePEUU

No variations in electrospun scaffold processing studied in this work produced significant change in mass fraction of polymer within the constructs (Table 1).

POLYMER MASS FRACTION IN EACH SCAFFOLD

	Dry ePEUU		*Wet ePEUU*	
Rastering Speed	0.3 cm/s	0.421+/- 0.011	0.3 cm/s	0.433 +/- 0.011
	1.5 cm/s	0.442 +/- 0.034	1.5 cm/s	0.451 +/- 0.014
	3.0 cm/s	0.383 +/- 0.023	3.0 cm/s	0.409 +/- 0.031
	30 cm/s	0.445 +/- 0.017	30 cm/s	0.379 +/- 0.005

Table 1. Mass fraction of polymer across the experimental groups. No statistically significant difference in this measure was observed across any group.

Increasing the rastering velocity above 0.3 cm/s while keeping all other variables constant appeared to produce a modest stiffening ($p<0.001$) of the circumferential axis, perpendicular to the axis of raster (Figure 4 a).

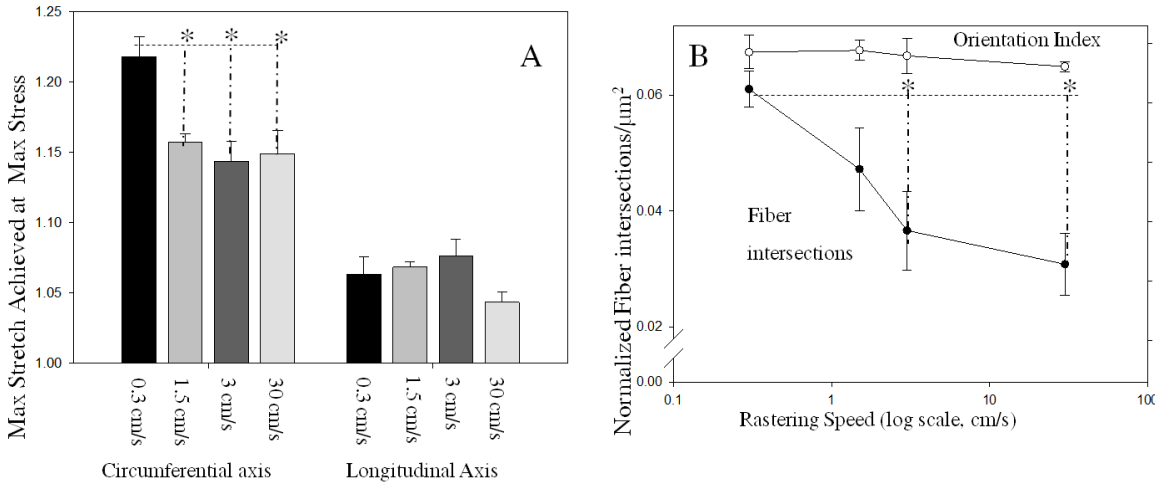

Figure 4. A) The effect of increased rastering speed on dry ePEUU biaxial mechanical properties. B) The effect of increasing rastering speed on fiber orientation and normalized fiber intersection density. ∗ Indicates statistically significant difference (p < . 05)

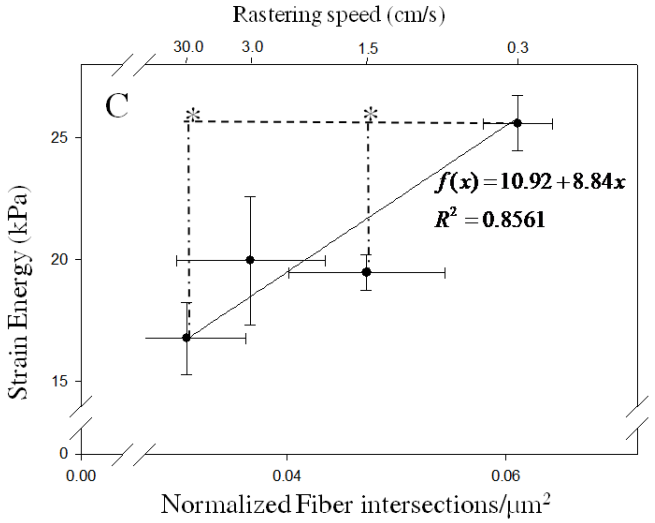

*Figure 5. The relationship between a fabrication parameter (rastering speed, top x-axis), a structural measure (fiber intersection density, bottom x-axis), and mechanical response (strain energy, y-axis). * indicates statistically significant difference (p < . 05)*

Image analysis revealed a pronounced decrease in fiber intersections at rastering velocities above 0.3 cm/s ($p < 0.01$) in a pattern reminiscent of the stiffening observed in the circumferential axis of the same constructs. This occurred without any significant change in the fiber orientation index (Figure 4b). Consistently, higher strain energy was observed in specimens processed under lower rastering speeds and, consequently, higher fiber intersection densities (Figure 5). None of the remaining architectural features (fiber angle distribution, connectivity, fiber diameter) identified by the image analysis algorithm demonstrated significant differences between groups.

3.4.2- EFFECT OF CONCURRENT MEDIUM SPRAYING

The inclusion of cell culture medium into the construct resulted in a dramatic change in scaffold microarchitecture (Figure 6 a, c). Wet ePEUU qualitatively appears to possess a greater degree of undulation, bundling, and looping that is less common in the 'dry' electrospun samples. Unlike dry electrospinning, structural analysis of these scaffolds demonstrated no significant difference in nodal density over the rastering speeds studied (Figure 5; $p=0.117$, 0.3 to 30 cm/s rastered groups). Fiber orientation did not differ significantly from that found in dry ePEUU. Faster rastering speeds during fabrication were associated with pronounced stiffening in the circumferential axis ($p<0.001$) (Figure 6b). Additionally, strain energies in this group did not differ significantly from those that characterized the dry ePEUU groups ($p=0.382$, data not shown). It can also be observed that introduction of culture medium alone only affected the mechanical response of ePEUU at the lowest rastering speed studied ($p<0.001$).

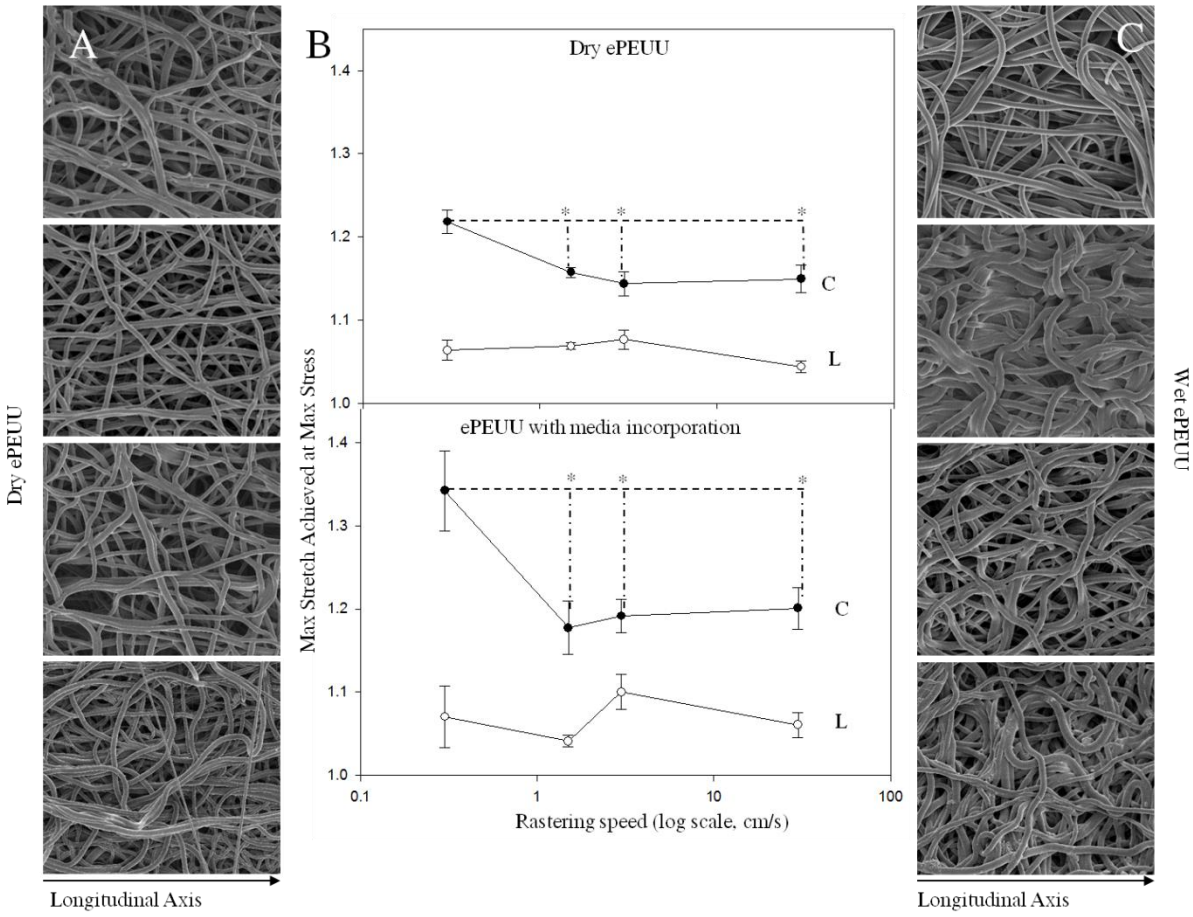

Figure 6. (a,c) Qualitative depictions of fiber microarchitecture of both dry and wet ePEUU scaffolds in descending order according to rastering speed, with 0.3 cm/s on the top and 30 cm/s on the bottom. b) Comparison of the mechanical response of dry and wet ePEUU across rastering speeds. (C indicates the circumferential axis, whereas L indicates longitudinal axis) ∗ indicates statistically significant difference.

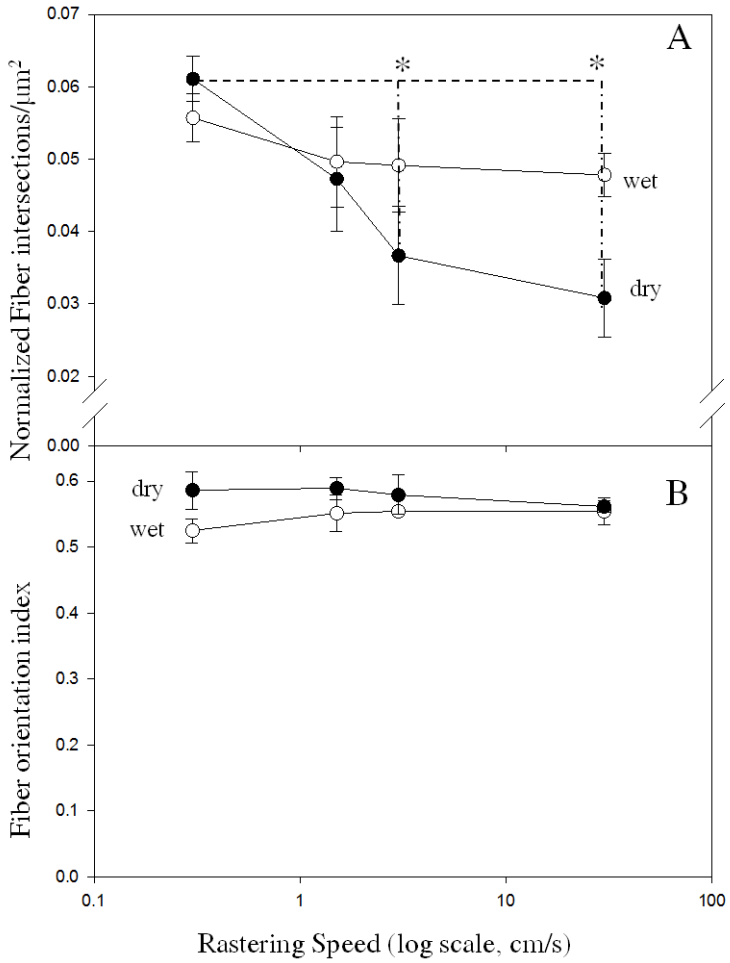

Figure 7. (a) Normalized fiber intersections vs. rastering speed for the wet and dry groups. (b) Fiber orientation index vs. rastering speed for the wet and dry groups.

3.4.3- EFFECT OF CELLULAR AND PARTICULATE INTEGRATION

Inspection of SEM micrographs (Figure 8) demonstrated that smooth muscle cells or microspheres concurrently electrosprayed into the ePEUU scaffolds become an integral part of the fibrous network, bridging multiple fibers.

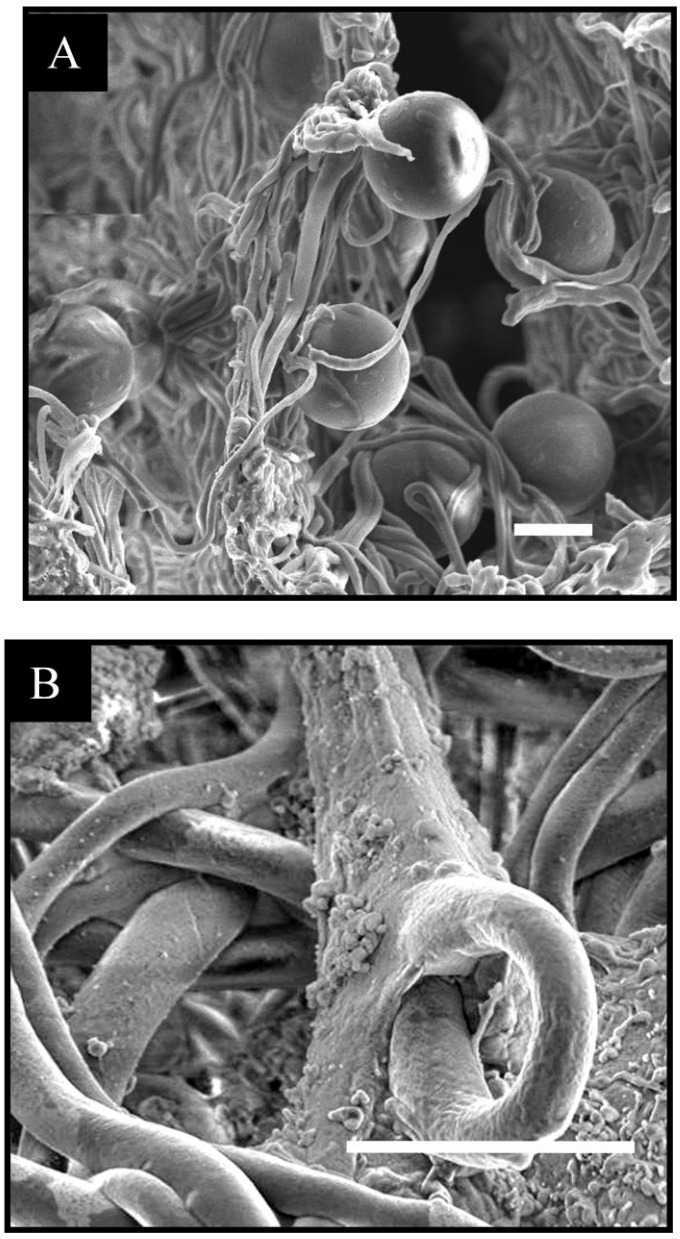

Figure 8 SEM micrographs depicting the immediate microenvironment surrounding integrated microspheres (a) and smooth muscle cells (b). Scale bar = 5μm

Interestingly, these particulates produced a significant increase in scaffold mechanical anisotropy beyond that found in either dry or wet ePEUU at the same rastering speed (Figure 9). This appears to be related to a stiffened longitudinal axis of

the construct. In contrast increasing the concentration of cells within the electrospray suspension did not affect the mechanical anisotropy, nor did replacing the cells with rigid polystyrene microspheres. No significant differences were observed between any micro-architectural parameters between microsphere integrated and wet ePEUU. More importantly, it was shown that when cells are integrated into an electrospun construct in conjunction with the adoption of slow rastering velocity (0.3 cm/s) the resultant construct possesses a mechanical response that resembles highly anisotropic soft tissues (Figure 10).

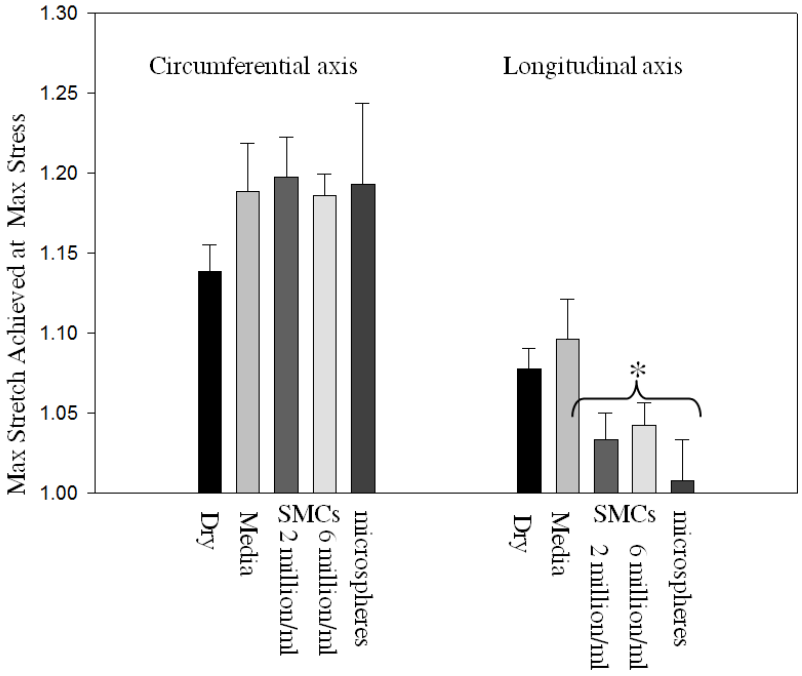

*Figure 9. The change in biaxial mechanical response resultant from integrated particulates within ePEUU fiber matrix. Rastering speed during fabrication was 3.0 cm/s, * indicates statistically signifi cant difference from both 'dry' and 'wet' groups.*

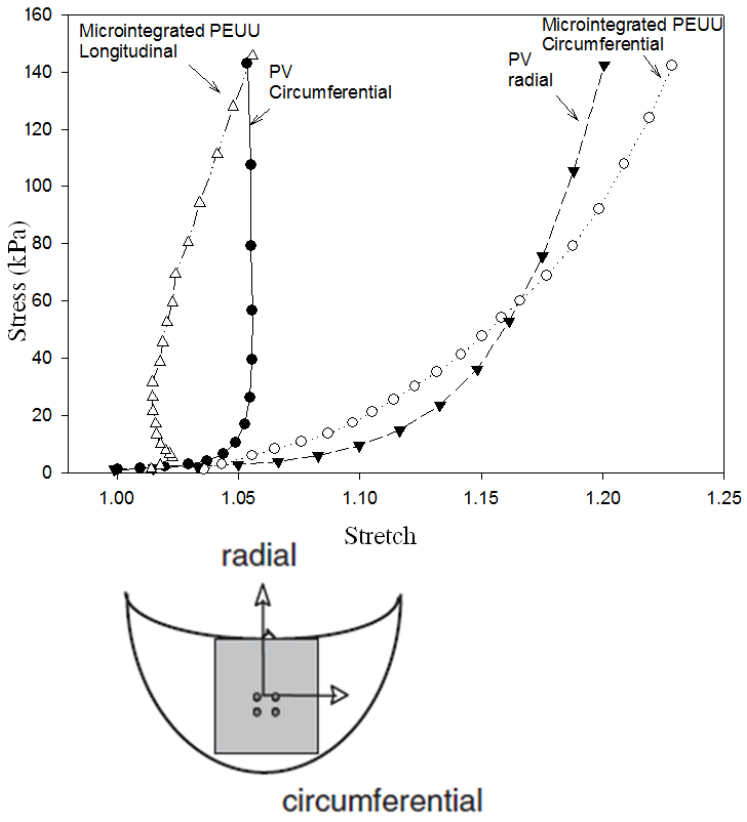

Figure 10. Combining slow rastering with concurrent cell electrospraying produces a construct with mechanical anisotropy reminiscent of the native porcine pulmonary valve. (plots not significantly different from each other).

3.5 - INTERRELATIONSHIP BETWEEN FABRICATION CONDITIONS, FIBER MORPHOLOGY, AND MECHANICAL PROPERTIES: RESULTS SUMMARY

The present study set out to modify electrospun fiber microarchitecture without altering fiber alignment and evaluate the effect of those modifications on bulk mechanical response. It was discovered that a higher degree of fiber intersections was associated with increased mechanical anisotropy and strain energy. This was

accomplished where not previously possible due to the introduction of a new method of microstructural analysis.

Current image processing techniques aiming to quantify fiber architecture of native tissues and engineered constructs rely prevalently on Hough transforms, intensity gradient based texture analysis algorithms [55, 138, 186], direct tracking methods [144, 187-188], and fast Fourier transform (FFT) based image analysis [145-146, 189]. Although robust and relatively easy to implement, these approaches can only quantify fiber angle distribution(s). In contrast, the importance of number and density of fiber intersections, network connectivity, and fiber diameter distributions on scaffold mechanical behavior has been previously discussed [116-117, 121, 159, 190-199]. Full scaffold network topology description provided by image analysis has the ability to more completely assess the influence of fabrication parameters on scaffold architecture [28, 56, 200]. Despite the intrinsic complex and chaotic nature of the electrospinning process, this approach in parallel with mechanical characterization was utilized to identify significant trends relating fabrication conditions, structural features, and bulk mechanical properties.

3.5.1 - RESULTS SUMMARY RASTERING

Modifying the rastering velocity is one of the simplest of the process modifications studied; however the mechanism behind its effect on overall scaffold mechanics is intricate and poorly understood. While it was expected that an increase in rastering speed would further orient the fibers, actual realized fiber alignment remained consistent through an order of magnitude increase in rastering speed despite a marked

change in mechanical properties (Figure 6 a, c). Previous results [55] indicated that an alignment due to increased rotational velocity will stiffen the rotational axis, however this was not significant until the mandrel was rotated at a tangential speed higher than 2 m/s, a faster rotational velocity than the one presented in this work. This finding suggests that fiber orientation alone does not adequately describe the mechanical response of these electrospun polyurethanes. An exponential decrease in fiber intersections was observed as rastering velocity was increased beyond 0.3 cm/s (Figure 2b). This pattern is reminiscent of the change in circumferential axis compliance observed. Furthermore, fiber intersections were shown to be correlated with strain energy (R^2 = 0.86) (Figure 5). As all other structural parameters remained comparable (fiber diameter, connectivity, orientation index) across the rastering speeds, it can be speculated that the fiber intersection density is potentially responsible for the observed increase in mechanical anisotropy and strain energy. Further research must be performed in order to fully understand this relationship and apply this knowledge to a structural deterministic model.

3.5.2 – CONCURRENT MEDIUM ELECTROSPRAYING

Concurrent electrospraying of cell culture medium onto the depositing scaffold during fabrication induced a distinct change in scaffold microstructure. This is likely due to an aqueous layer that adhered to the forming scaffold as it rotated, delaying but not preventing fiber bonding. It would follow that there would be more slack length, which would allow the additional, looping and undulation observed in SEM micrographs (Figure 6 a, c). At a rastering speed of 0.3 cm/s, this leads to a more

compliant mechanical response, however this appears to be overcome by increasing the rastering speed. It follows logically that the increase in tortuosity might create artifactual fiber intersections at locations where fibers are not actually securely bound together. This is admittedly a limitation with the current analysis, and has an implication in explaining the lack of definitive structural pattern with respect to rastering speed in wet ePEUU images while every mechanical trend observed in dry ePEUU remains consistent.

3.5.3 – EFFECTS OF PARTICULATE AND CELL INTEGRATION

When the electrosprayed medium is supplemented with a concentration of particulates or cells, a distinct change occurs in the mechanical response of the scaffold. The microsphere size (10μm) was chosen to be the same approximate physical size as the smooth muscle cells before they spread among the fibers. At this size, it seemed unlikely that they would cause substantial interference with fiber bending. However, a consistent pattern of a stiffened longitudinal axis was observed in the mechanical response of particulate integrated constructs (Figure 9). A potential interpretation of this phenomenon is that particulates could serve as bridges connecting fibers to an extent beyond that which would be found with fiber intersections only. Following this assumption, particulates and cells would act as additional fiber bonding increasing the effective fiber intersection density and consequently raising the level of mechanical anisotropy. Further studies are required to confirm this hypothesis.

3.6 – IMPLICATIONS FOR FUTURE STUDIES

Consistent trends were observed relating fabrication conditions to bulk mechanical response through key microstructural changes. Through these observations, it was shown that electrospun fiber alignment is inadequate as the sole descriptor of scaffold micro-scale structure. A more complete evaluation of fiber architecture which includes the influence of fiber interconnections is required to evaluate structure-function relationships. The complexity of electrospun scaffold morphology and the consequential difficulties in collecting quantitative structural information, particularly in wet processed ePEUU, imply the clear need for additional studies aiming to consolidate the aforementioned hypothesis. Specifically in this context structural deterministic modeling [28, 116-117, 121, 133, 190-195] represents an important approach in future studies. Micro-meso architecture based mechanical models might, for instance, be adopted to investigate the influence of fiber intersection density on the material mechanical response.

Wet processing and mandrel rastering were successfully implemented as tools to reliably modify scaffold microarchitecture without altering fiber alignment. Rastering was speculated to introduce macro-scale mechanical changes through modification of fiber intersection density. This method of scaffold fabrication was found to be nearly as effective in altering scaffold anisotropy as fiber alignment due to increasing tangential velocities. The results suggest the possibility that fiber alignment and intersection density can be controlled independently, granting an additional level of control on scaffold microstructure at the time of fabrication. This control can additionally be

applied in conjunction with cell electrospraying to create highly anisotropic cellularized constructs without utilizing high rotational velocities which are not amenable to such a technique[60, 201]. The mechanical and structural data described in this chapter have been used to feed the structural deterministic model adopted in this study and illustrated in the next chapter.

CHAPTER 4

STRUCTURAL DETERMINISTIC MECHANICAL MODELS FOR ELECTROSPUN SCAFFOLDS

4.1 – BIOMECHANICAL MODELING OF SOFT TISSUE ENGINEERING SCAFFODS

Interest in electrospun polymeric nano-microfibers for tissue engineering applications has rapidly grown during the last decade. In spite of this technique's flexibility and ability to form complex fiber assemblies, additional studies are required to elucidate how the fibrous microstructure translates into specific tissue (or meso-scale) level mechanical behavior. The first chapter of this thesis introduced the state of the art soft tissue engineered scaffold technologies, the second described the novel image analysis paradigm created and proposed in this study, the third chapter showed the influence of fabrication parameters on both scaffold micro-structure and mechanical response. Finally, this chapter illustrates the mechanical model proposed in this work, its results, and implications.

Modeling approaches for soft tissue engineering scaffolds follow into the three main categories:

- Phenomenological;
- Continuum;
- Structural [116-117, 121, 125, 128-129];
    a. Statistical [55];
    b. Deterministic [116-117, 121, 125, 128];

In phenomenological models generally, strain energy models with anisotropy are derived from the fiber architecture. They simply fit experimental variables to mathematical expressions. Although, correct mechanical response prediction can be produced, this class of models does not provide any physically meaningful connection between the manufacturing parameters, the material structural characteristic and the mechanical behavior.

The continuum is the most powerful approach however, a complete continuum theory for network-like scaffolds or tissues still needs to be developed. Structural models can be further classified as Statistical or Deterministic. In statistical approach the material bulk mechanical response depends on the sum of the response of its constituents. Micro-structural information can be effectively incorporated into the model, fiber to fiber interactions are not modeled. Hence, the totality of the network related structural information are ignored. Information at meso level cannot be provided, as a consequence electrospun scaffolds and native tissues multi-scale nature cannot be described nor predicted. As it has been highlighted in the second chapter

Representative Volume Element (RVE) size determination is generally ignored in the material characterization procedure.

Finally, according to the structural deterministic paradigm [116-117, 121, 125, 128] the microstructure is modeled in a representative way and coupled to the macroscopic equations through a multi-scale technique such as homogenization or volume averaging. In previous attempts to implement this modeling strategy the discrepancies between the real material and the simulated fiber network have never been quantified. Similarly, fiber overlaps and connectivity have never been reproduced in the simulated network in a controllable manner. Most importantly the limited size of the unit cells affected the capability of these models to elucidate the complex multi scale nature of native and engineered tissues [148]. Limitations in state of the art implementations of this methodology affect the result quality at different scale length, main caveat can be summarized as follows:

- Macro level (1 cm): multi-scale approach introduces an error by definition, the mechanical response is estimated but not predicted with accuracy [116-117, 121, 125, 128] ;
- Meso level (100 μm): the web-like structure formation under strain reported in [64] is not reproduced;
- Micro level (10 μm): Heterogeneous deformation at micro level, NAR-strain curves or equivalent descriptors of material-cells interactions cannot be predicted because of the limited size of the unit cell [148];

Deterministic structural models can quantify how key structures contribute to the mechanical response as a function of bulk deformation across multiple scales, as well as provide a better understanding of cellular mechanical response to local micro-structural deformations. Our ultimate aim is to utilize such models to assist tissue engineering scaffold design. In the current work, we presented a novel approach to automatically quantify key micro-architectural descriptors (fiber overlaps, connectivity, orientation, and diameter) from SEM images of electrospun poly (ester urethane) urea (PEUU) to recreate statistically equivalent scaffold mechanical models. Material models were then generated specifying: fiber overlap density, fiber orientation, connectivity and fiber diameter. Macro-meso mechanical response was predicted via FEM simulations in ANSYS environment, the mesh generator code was developed in Matlab and is available in appendix B - C, ANSYS input files are provided in appendix D.

In spite of a considerable harvest of studies aiming to characterize soft tissue engineering scaffolds morphology [55, 123, 145] and its influence on material mechanics and cell proliferation, insufficient efforts have been spent in assessing how the analyzed area or volume sizes can affect the structural data reliability. Morphology descriptors produced through image analysis such as material porosity, fiber density, fiber alignment distribution, fiber connectivity distribution and fiber diameter histogram strongly depend on the material architecture at micro-meso level.

The latter can be demonstrated studying the evolution of these parameters over area/volumes of increasing sizes and/or repeating the image analysis over area/volumes of different locations. Morphology features fluctuations and location dependency

gradually cease as the analyzed area/volume approaches appropriate Representative Volume Element size. A direct implication is that an image analysis technique remains incomplete if it does not identify an appropriate RVE for the variable object of the analysis. The RVE size in a random composites can be derived statistically, numerically [168] or empirically studying the stabilization of the analyzed variable over areas/volumes of increasing sizes. Thus, addressing the issue of identifying the appropriate RVE size has a twofold importance: firstly in terms of material characterization it contributes to develop reliable tools to assess scaffold manufacturing process repeatability, secondly in terms of mechanical modeling the analysis performed at the RVE only can provide physically meaningful data. In particular, structural deterministic models dramatically depend on how rigorous is the material structural description.

This need has been extensively highlighted by [116-117, 121, 125, 128] where the model capability relies prevalently on the accuracy of the network topology. To this extent, not only the micro architectural data extraction accuracy is crucial but also it is fundamental that the stochastic representation of the engineered scaffold is able to reproduce the extracted information minimizing the error between the real structure and the simulated one. The great potential of the structural deterministic approach in elucidating the inherently multi-scale nature of native and engineered soft tissues response seems to justify its apparent complexity [116-117, 121, 125, 128]. An alternative way within the promising path of the deterministic modeling is to reproduce the entire scaffold area or volume without duplicating the RVE as it has been done in [116-117, 121, 125, 128]. The main benefits of this alternative solution are that (1) the

implicit error introduced by the multi-scale approach is removed and (2) the information at meso level are preserved. For instance, tortuosity measurements of a collagen fiber in a heart valve leaflet under loaded and unloaded configurations can be performed whereas are neglected in the multi-scale approach where the fibers cannot cross element boundaries. However, these expected benefits are counterbalanced by a significant increase of computational costs. The latter makes this approach feasible when the nature of the problem is two dimensional like in the case of the elastomeric scaffolds produced by electrospinning [55]. Finally, an additional value of this modeling strategy consists on the possibility to simulate ideal design scenarios controlling design features at macro (tissue engineered construct size, shape etc.) and micro level (fiber connectivity, fiber density, fiber alignment etc.). This represents an essential milestone in the attempt of replacing a trial and error philosophy with a more rational design approach. The modeling strategy proposed in these doctoral studies overcomes the aforementioned limits, the main advantages are listed below:

- It is classified as a structural deterministic thus, it overcomes the limits of continuum and phenomenological approaches;
- Fiber to fiber interactions are modeled, two novel micro-architectural features are introduced: (1) overlaps density, (2) connectivity;
- Appropriate RVE is determined;
- It produces fiber networks stochastically equivalent to the real fiber networks detected in the original scaffold. Discrepancies between the real and simulated network are quantified and minimized.

- FEM simulation on a unique consistent network enables the mechanical response to be determined across the scales.

In summary, the purposes of this modeling efforts are threefold: (1) Guide tissue engineering scaffold design [116-117, 121, 123, 125, 128], (2) Provide a better understanding of cellular mechanical and metabolic response to local micro-structural deformations [64, 123, 148], (3) investigate structural changes as a function of deformation across multiple scales [148]. The presented approach is applicable to a broad spectrum of applications covering the most emerging biomaterials in soft tissue engineering such as electrospun constructs, hydrogels, fibrin gels, collagenous and network like native tissues, decellularized tissues.

4.2 – BIOMECHANICAL MODEL OF ES-PEUU: METHOD

The modeling steps can be summarized as follows: (1) extraction of micro-architectural scaffolds data from SEM images, (2) generation of stochastically equivalent fiber networks, minimization of the discrepancies between the real and the simulated fiber network, (3) equi-stress, displacement control biaxial testing simulation in ANSYS environment, the model outputs can be listed in relation to the characteristic length of the variable/phenomenon studied:

- Macro level (1 cm):
    - bulk mechanical response prediction, (membrane tension vs. stretch);

- elasticity moduli estimation (single fiber modulus);

- Meso level (100 μm):
  - network deformation; strain map showing different stain patterns due to anisotropy, fibers aligned toward the cross preferential direction tend to reach strain levels higher than those oriented toward the preferential one;
  - network activation; stress map showing long fiber behaviors and recruiting pattern;
- Micro level (10 μm):
  - Nuclear aspect ratio vs. stretch prediction (only on cells seeded scaffolds);

4.2.1 – EXPERIMENTALLY DERIVED MESH GENERATION

The micro-architectural data were extracted for three different electrospun PEUU scaffolds (1.5 m/s, 4.5 m/s, 9.0 m/s) using the procedure described in the second chapter. A custom made software has been developed to produce fiber networks stochastically equivalent to the ones detected by image analysis, the code implemented in Matlab is provided in appendix B.

Detected nodal spatial distribution showed that fibers overlaps/intersections are randomly distributed over the real material surface (Figure 1). Therefore, a random distribution of points, with the same real material overlaps/area ratio, can reproduce the original overlaps 2D spatial distribution.

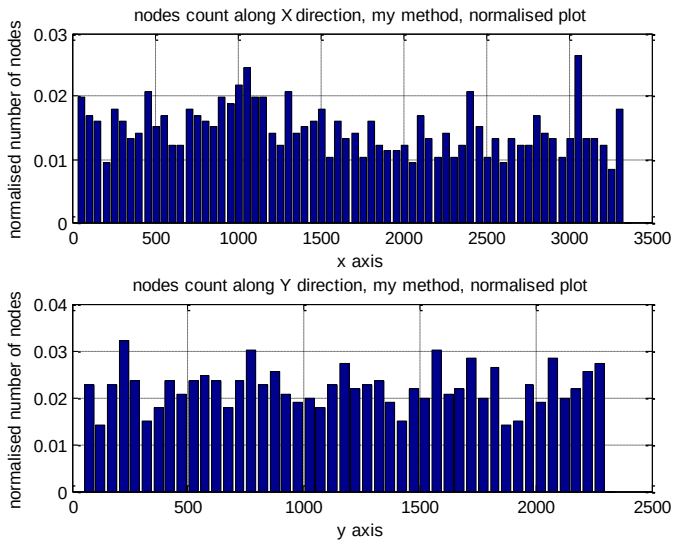

Figure 1. Nodes spatial distribution. ES-PEUU 9.0 m/s scaffold. X axis pixels, Y axis normalized number of nodes.

The Delaunay network definition was applied to the generated cloud of fibers overlaps, generating fiber network of known fiber intersection density, diameter, and random connectivity and fiber angle distribution (Figure 2).

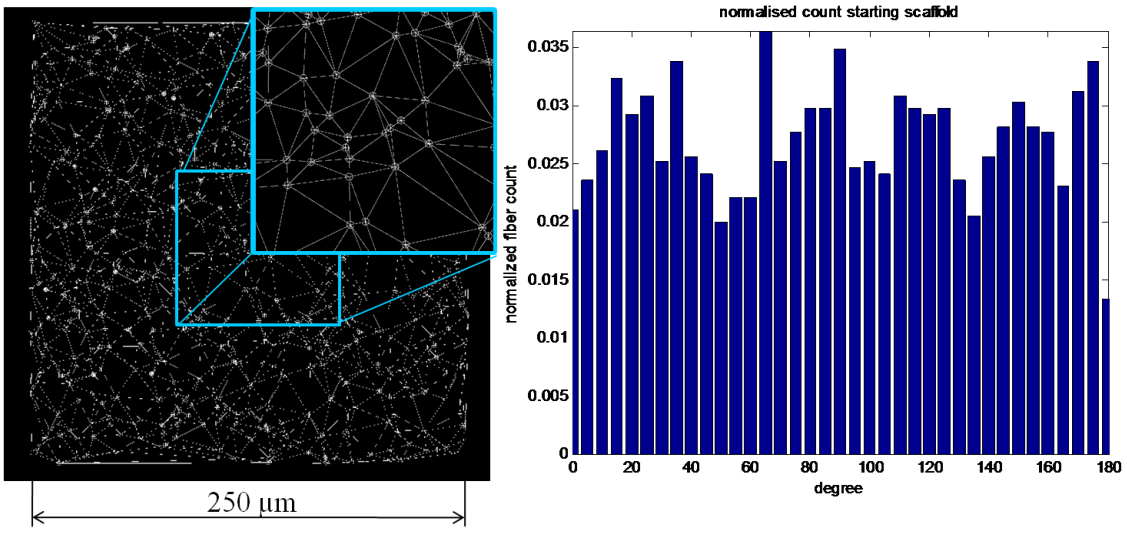

Figure 2. Generated fiber network with known fiber intersection density, diameter and random fiber angle distribution and connectivity.

The network area was divided in subsets along two directions perpendicular and parallel to the preferred fiber alignment direction respectively. In each subset, fiber overlap positions were modified by concentrating/extending fiber coordinates over the direction perpendicular/parallel to the preferred fiber alignment direction (Figure 3).

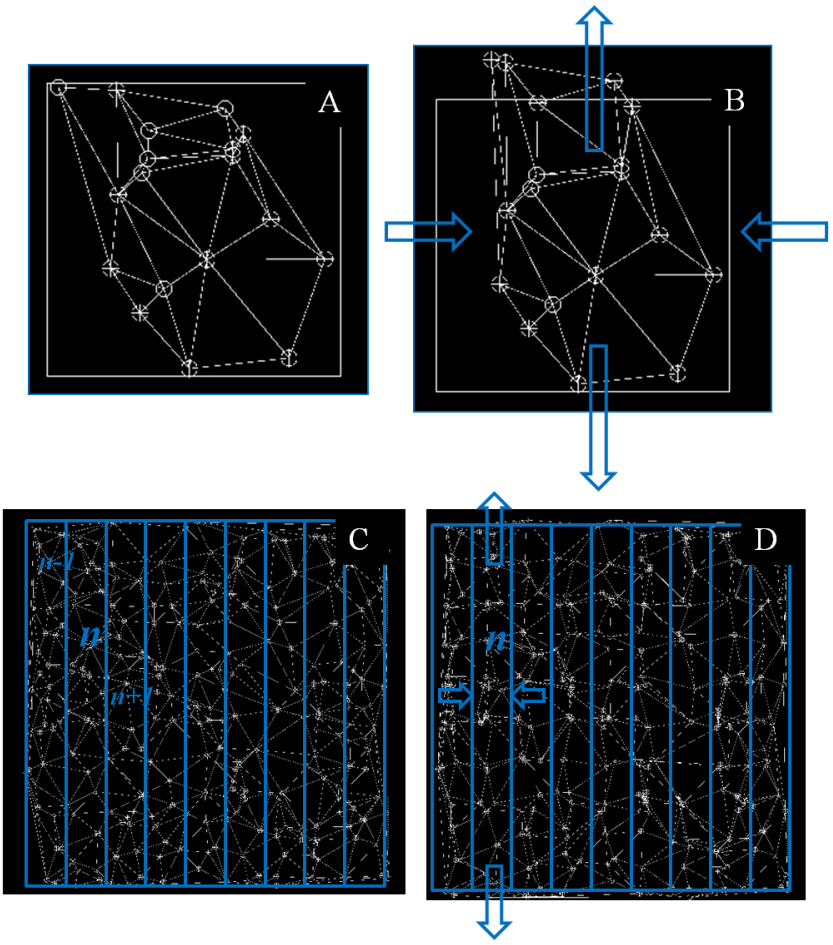

Figure 3. a) Starting network, b)Network deformed along the vertical direction, keeping the original connections and changing the nodes position, c) random network divided in subset, d) subsets of the original network similarly to b the nodes position is changed in order to induce alignment along the vertical direction.

Unwanted alignment at the desired angle of alignment +-90º was eliminated identifying and erasing fibers responsible for secondary alignment peaks (Figure 4). Full control over the fiber angle distribution was achieved combining the steps described (Figure 5).

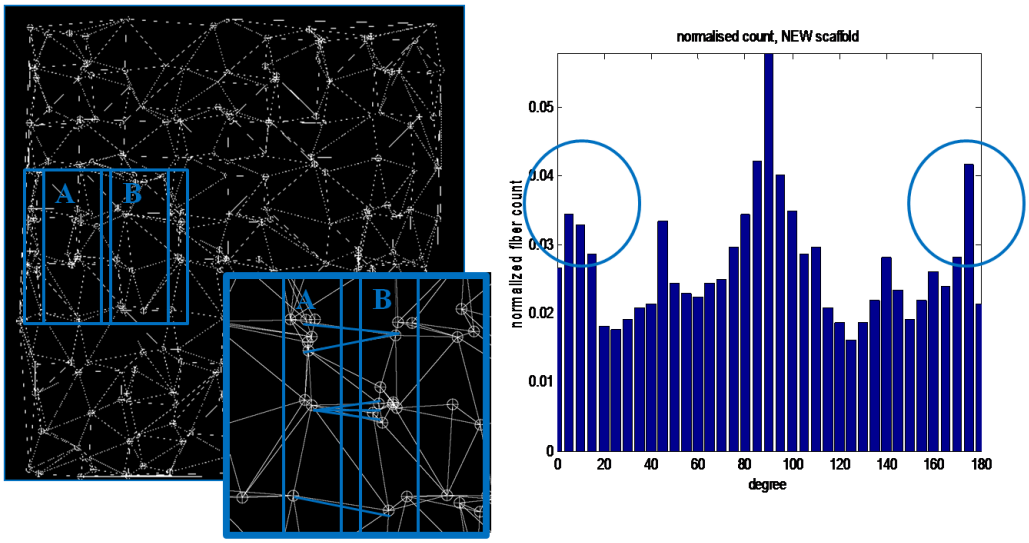

Figure 4. Deformed network, a and b subsets stretched to induce alignment at 90°, onset: fibers responsible for secondary alignment at 0 and 180 degrees.

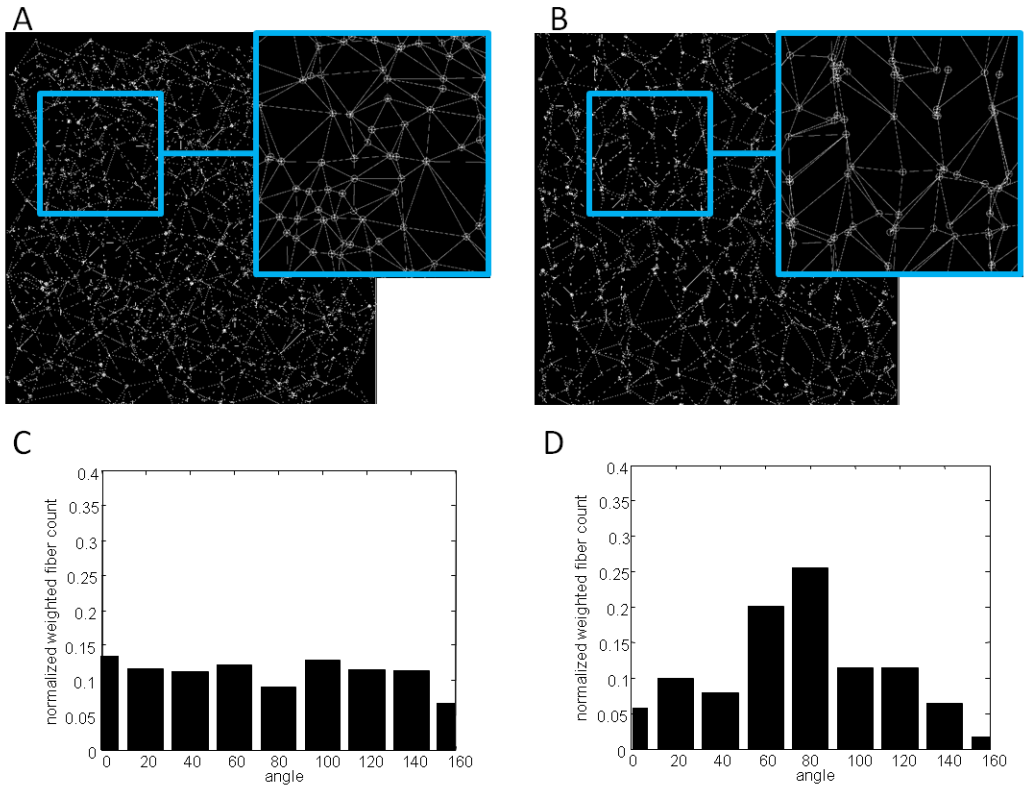

Figure 5. a) original network with random fiber distribution shown in c). Same network with imposed alignment at 90°, related fiber angle distribution presented in d). Comparison of the onsets in a) and b) highlights the increase in alignment over 90° .

The connectivity distribution was controlled eliminating a random fraction of fibers from the starting simulated networks, this step was necessary since the original Delaunay networks are over connected if compared with the real material networks (Figure 6).

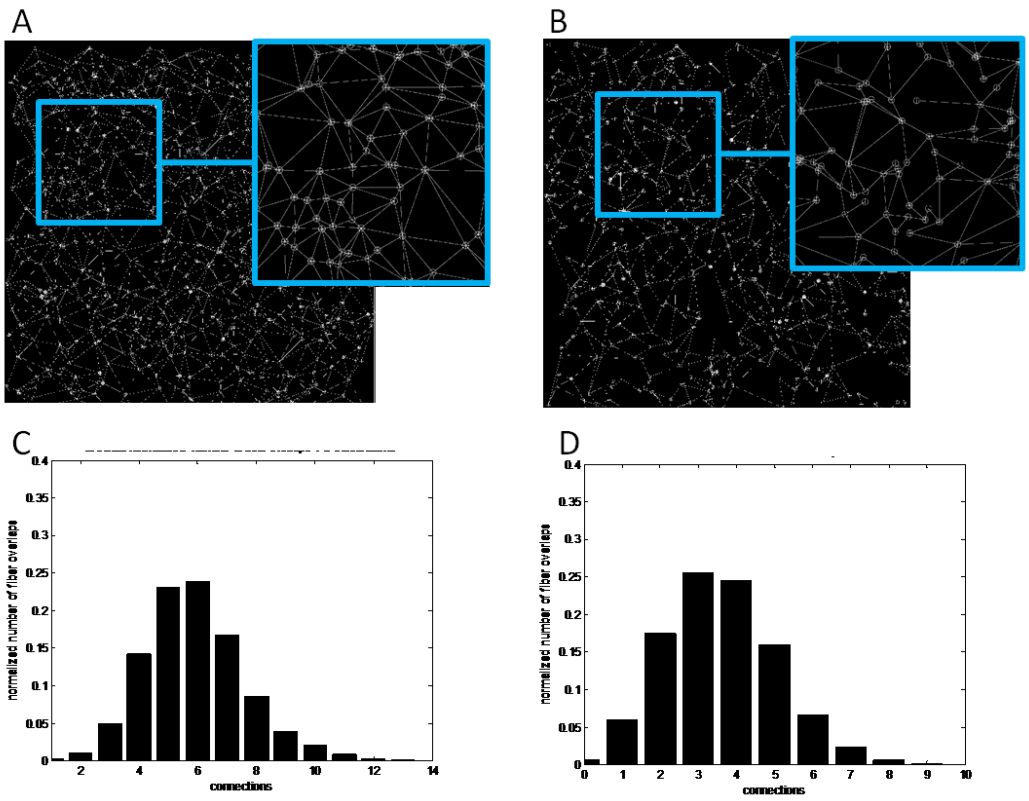

Figure 6. a) original network with connectivity plot maximum at 6 c). same network with reduced connectivity, maximum at 3 d). Comparison of the onsets in a) and b) highlights the reduction in connectivity, the network appears less dense.

The described coding cascade enable network models to be generated fully controlling the four variables previously described in chapter 2: (1) fiber angle distribution, (2) fiber connectivity plot, (3) fiber intersection density, (4) fiber diameter.

In order to study and simulate the mechanical behavior of soft tissue scaffolds up to the organ level (e.g 3 cm x 3 cm samples) a coding strategy has been implemented to

enable the generation of large and computationally expensive planar network models. The generation of such models requests the production of large quantity of data. In brief, instead of generating one unique network requesting one unique binary matrix defining the connections between the nodes the network was divided in sub areas. Each of these areas is separately generated, the related data are stored independently. This, reduces dramatically the amount of RAM memory required to . Additional subregions are produced to connect the different subareas (Figure 7).

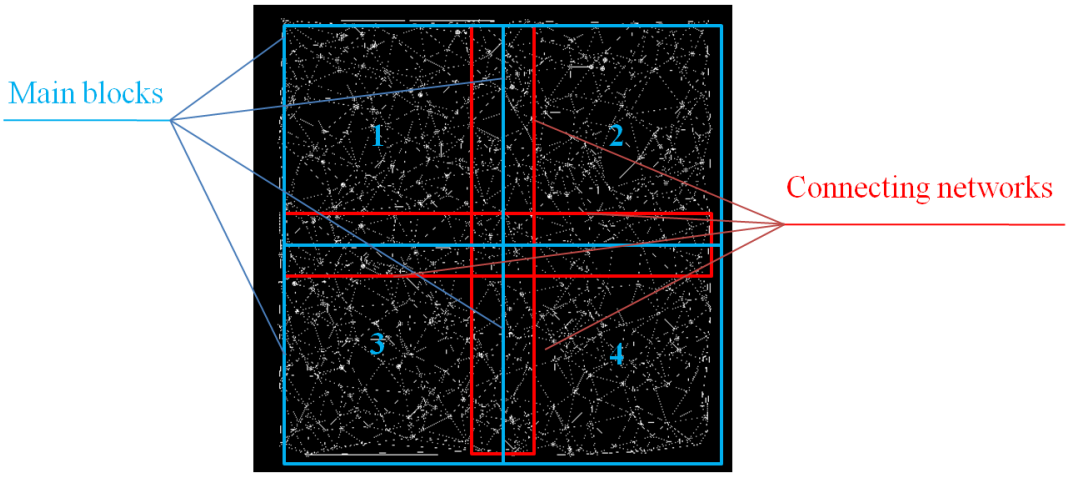

Figure 7. Strategy implemented to reduce the amount of memory required to generated the network models: step 1) main blocks are generated and their related data are stored, step 2)connecting network are generated covering main blocks edges, related data are stored.

In order to quantify and minimize the real – simulated networks discrepancies an error function was defined. The inputs of the minimizing function (Appendix C) were two parameters controlling the simulated network fiber alignment (DEF) and connectivity respectively (ConnIndex). Fiber network of 250 µm x 250 µm were

generated for DEF and ConnIndex values from 0 to1 with a step of 0.1. The error function output was defined as the sum of three terms:

1. Normalized overlaps density difference (real-simulated);
2. Normalized fiber angle histogram difference (real-simulated);
3. Normalized connectivity histogram difference (real-simulated);

Error maps were generated (Figure 8), the combinations of parameters producing the minimum error were identified for the three scaffolds (Table 1) (1.5 m/s, 4.5 m/s, 9.0 m/s) (Figure 9-12).

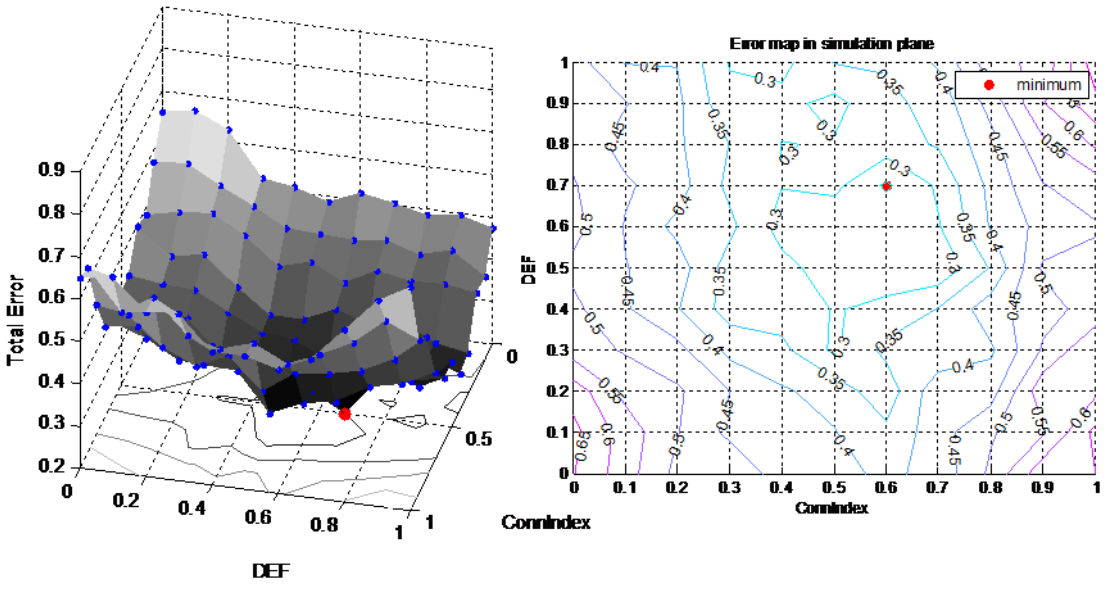

Figure 8. Error map example for the 9.0 [m/s] scaffold, minimum error detected for DEF = 0.7 and ConnIndex=0.6.

	1.5 m/s scaffold	4.5 m/s scaffold	9.0 m/s scaffold
DEF	0.2	0.4	0.7
ConnIndex	0.7	0.7	0.6

Table 1. DEF and ConnIndex values that minimize the discrepancies between the artificial and the real scaffolds.

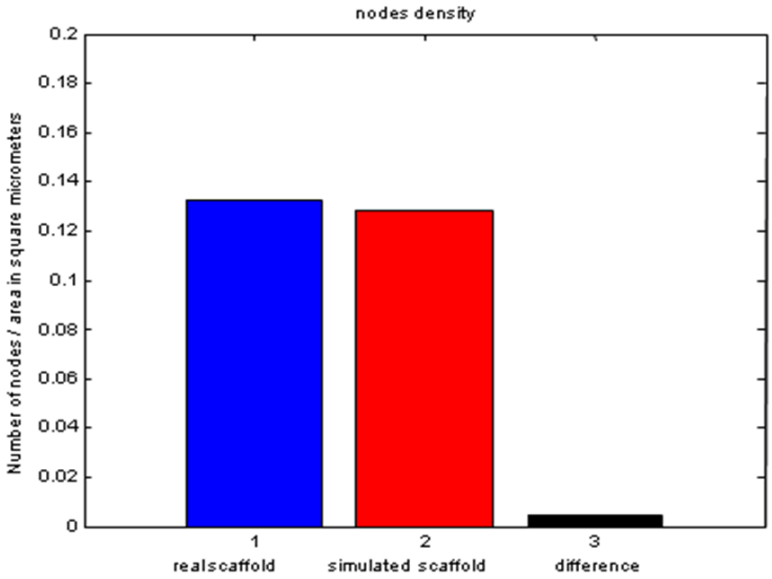

Figure 9. Example of the error minimization procedure for the 9.0 [m/s] scaffold. Fiber intersection density for real (blue), simulated scaffold (red), difference highlighted in black.

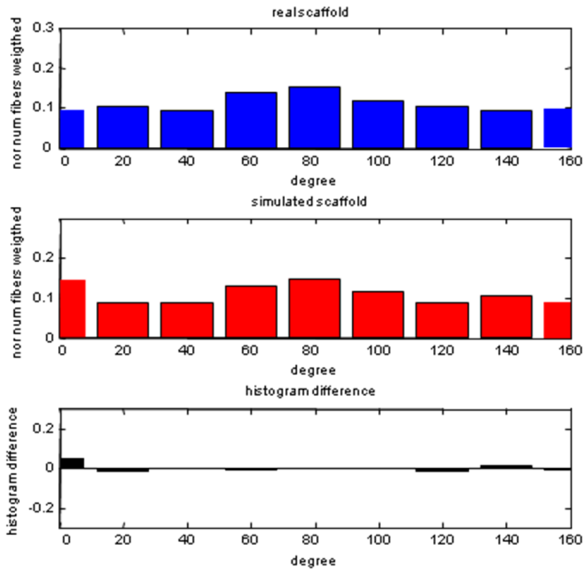

Figure 10. Example of the error minimization procedure for the 9.0 [m/s] scaffold. Fiber angle distribution for real (blue), simulated scaffold (red), difference highlighted in black.

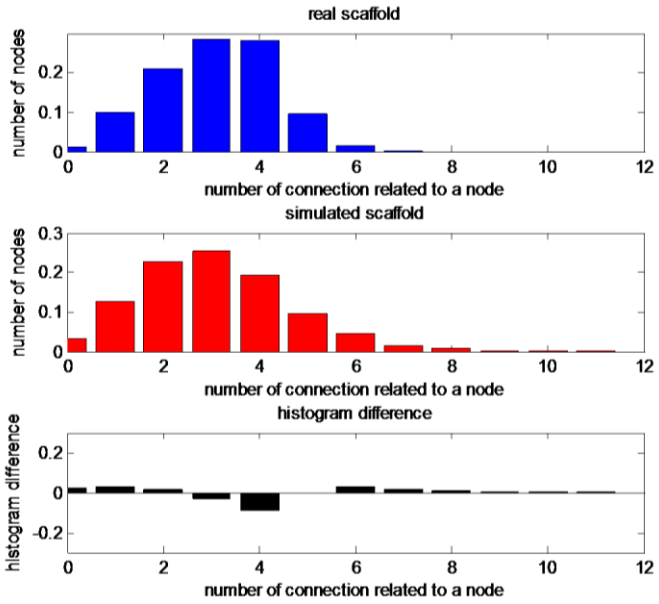

Figure 11. Example of the error minimization procedure for the 9.0 [m/s] scaffold. Fiber connectivity plot for real (blue), simulated scaffold (red), difference highlighted in black.

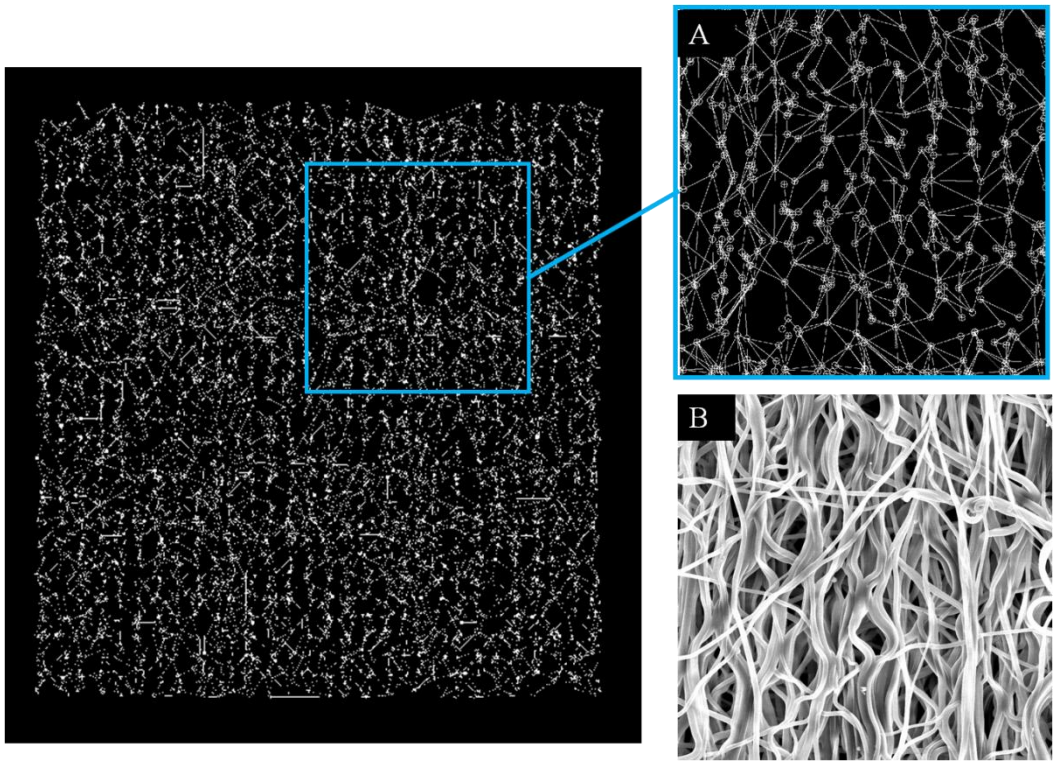

Figure 12. Example of final result of the mesh generation and error minimization, 9.0 [m/s] scaffold model, size=120 µm, OI = 0.65 fiber diameter= 0.5 µm, fiber intersection density = 0.28 [int/ µm^2] , connectivity plot maximum = 3, the comparison between the onset a) and the real scaffold b) confirms the high quality of the result.

4.2.2 – FINITE ELEMENT MECHANICAL MODELING

The simulated networks were then imported into ANSYS environment wherein the fibers were assumed to be linear elastic using link elements (ANSYS link10) with known axial and bending stiffnesses. In order to compare the model prediction with the experimental data from the biaxial testing the force produced by the mechanical model and the force experimentally derived from the equi biaxial stress test were converted in stress using the followings:

- $Force_{simulated}$= $Force_{simulated}$ / (sample size· number of fibers layers · fiber diameter);
- $Force_{experimental}$ = $Force_{experimental}$ / (sample thickness);

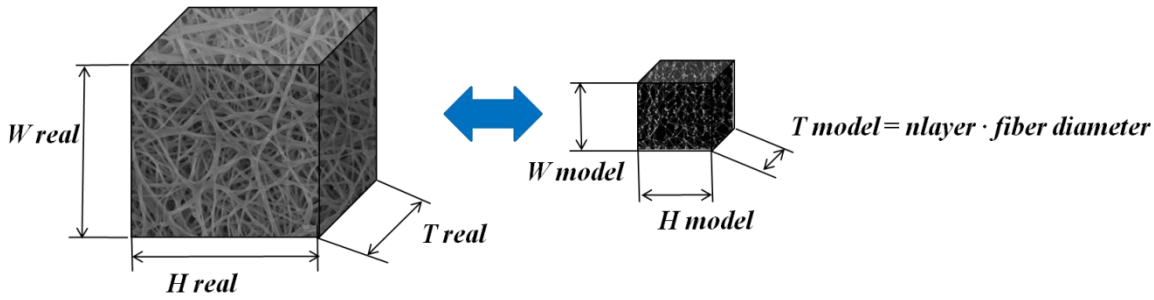

Figure 13. Mechanical response scaling factor evaluation to compensate the difference in volume between the real sample and the model, figure defines the real and simulated volume sizes.

Sample sizes (H model and W model) were imposed by the mesh generator (250 µm), fiber diameters were the average values from the fiber diameter distribution detected for each real scaffold (Chapter 2), sample thickness and sizes (W, H, T real) were measured with a caliber on the real samples (average of 9 measurement on each

sample). The number of fiber layers multiplied by the fiber diameter represents the thickness of the generated model (T model). Briefly, the extracted micro-architectural data are representative for a volume having a thickness equal to the number of fiber layers detected multiplied by their diameter. The number of layers can be derived studying the material bulk porosity. The material porosity can be derived extending the procedure developed and illustrated in the second chapter, the image processing cascade produced a binary filter that represents the area covered by fibers (Figure 14 d). The porosity can be estimated using the following relation:

$$Porosity = \frac{Volume_{voids}}{Volume_{total}} = 1 - \frac{Volume_{fibers}}{Volume_{total}} = 1 - \frac{Volume_{fibers}}{W_{real} \times H_{real} \times T_{real}} = 1 - \frac{Length_{total\ fibers} \times \frac{\pi(fiber\ diameter^2)}{4}}{W_{real} \times H_{real} \times nlayers \times fiber\ diameter} =$$

$$1 - \frac{\frac{\pi}{4 \times nlayers}}{W_{real} \times H_{real} / (Length_{total\ fibers} \times fiber\ diameter)} \qquad \text{(Eqs. 1)}$$

Hence, the porosity can be related to the number of layer detected by the image analysis procedure described in chapter 2 and to the ratio between the white pixels indicating areas occupied by the pixels and the totality of the pixels within the analyzed picture (Figure 14 d, Figure 15 and Eq. 2) the code producing the porosity estimation is provided in Appendix b) .

$$Porosity = 1 - \frac{\pi/(4 \times nlayers)}{(fiber\ pixels/background\ pixels)} \qquad \text{(Eq. 2)}$$

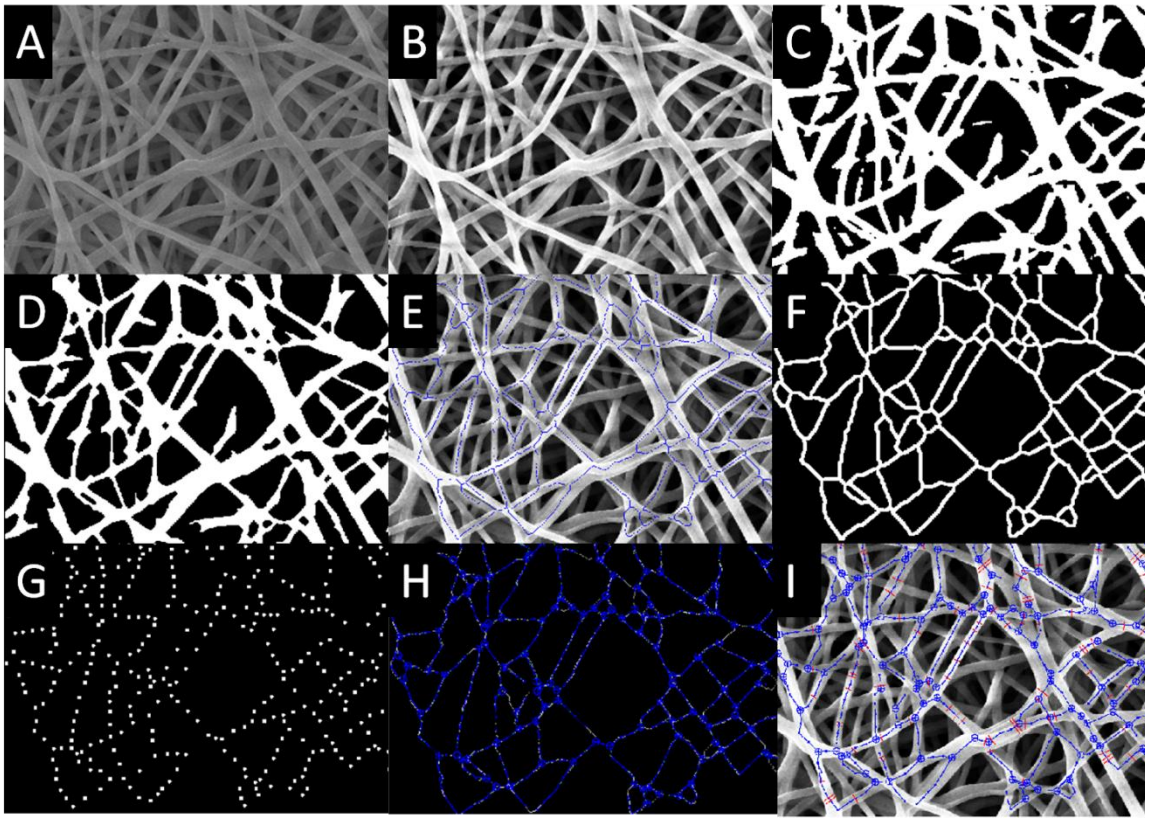

Figure 14. A) Starting SEM image. B) Image histogram equalization followed by 3 by 3 median filtering. C) Local thresholding through Otsu method. D) Thinning, smoothing and removal of isolated pixel areas through a cascade of different morphological operators. E) Skeletonization. F-G) binary filters for Delaunay network refinement. H) Modified Delaunay network associated to the real fiber network. I) Final network and fiber diameters detected.

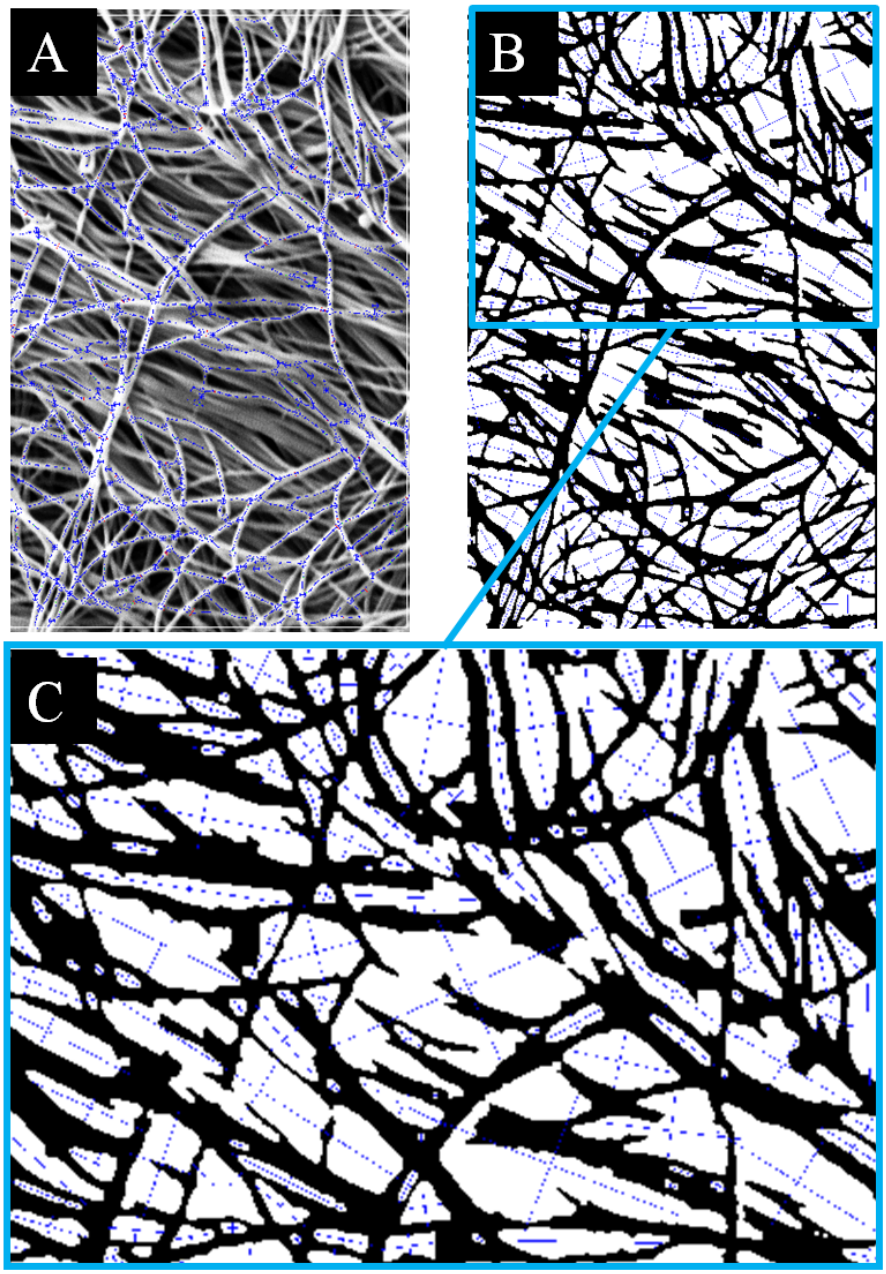

Figure 15. a) Example of analysis on decellularized rabbit aorta, b) image complementary to the fiber network, white areas represent the pores, blue segments the pores major axis, c) onset of b), porosity related measurements such as bulk porosity, pores size distribution, pores aspect ratio, pores orientation are based on these image.

The *nlayers* was considered to be an unknown and was determined imposing the minimum difference between experimental derived porosity values and the ones produced by the algorithm using guess values for *nlayers* . A data base of 35 ES-PEUU scaffolds analyzed by mercury intrusion porosimetry was adopted (0, 1.5, 3, 4.5, 9.0 [m/s] groups) (Figure 16), the optimal value for *nlayers* was 1.73 (Figure 17).

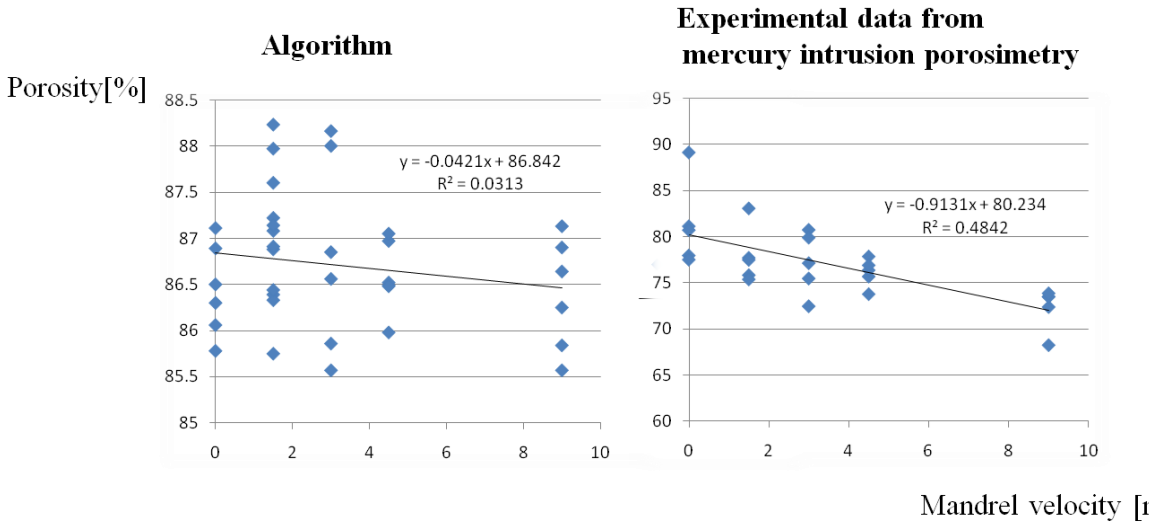

Figure 16. Porosity estimation from the developed algorithm and the mercury intrusion porosimetry technique, n=5 different scaffold batches for each scaffold family.

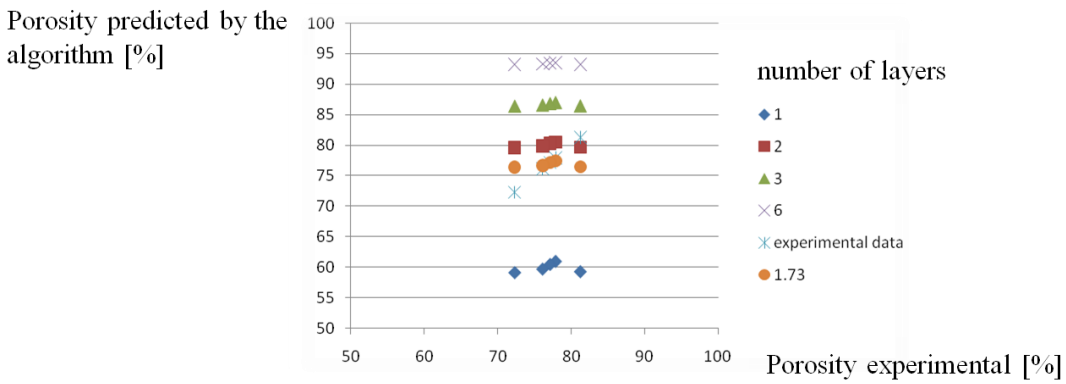

Figure 17. Number of layers analyzed by the image analysis technique presented. optimal value for the number of layers was 1.73.

Fiber compressive response was neglected under the assumption that PEUU fibers cannot produce significant responses under compressive load. Fiber intersection were modeled as simple hinge joints. Both the assumption are supported by the qualitative observation of the fiber network kinematic under increasing level of stretch [jbmr stella]. Equi-biaxial stress conditions [129] were reproduced with appropriate boundary conditions (Figure 18), the single fiber mechanical stiffness was the only parameter required to fit the equi-biaxial experimental data.

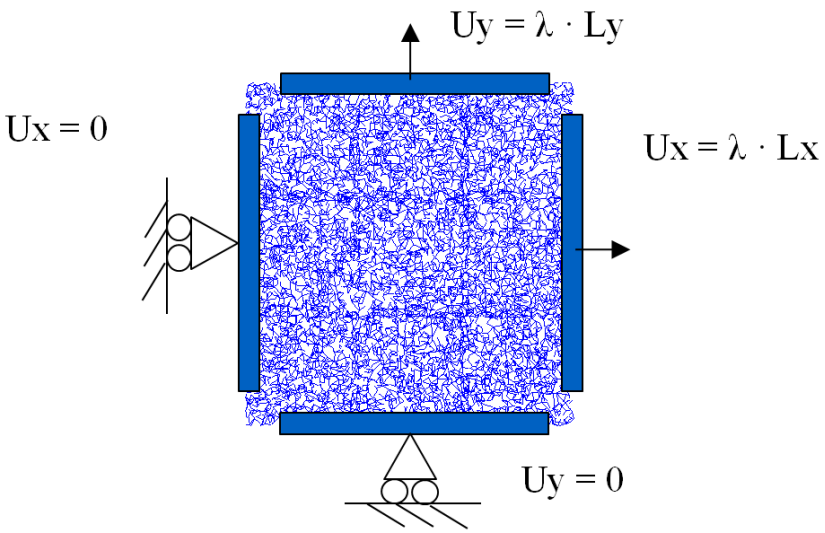

Figure 18. Schematic of the boundary conditions applied, Lx and Ly are the model sizes, λ is the stretch. The represented boundary conditions were applied to all the nodes following within the clamping areas in blue.

Large deformations were enabled, due to the several levels of non-linearity introduced by: (1) material topology, (2) material properties and (3) boundary conditions the Newton Rapson method was adopted in the solver. Although the constitutive equation adopted was simply linear the shift between compression and tension produced non-linearity in the material property. The FE simulations involved networks with 2000-3000 nodes and 10000-11000 truss elements. FEA results are presented in the next paragraph. Models sizes had an area of 62500 μm^2 larger than the maximum value detected for the 9.0 [m/s] scaffold Representative Area Element ~11000 μm^2 (RAE).

In addition to bulk mechanical response the structural deterministic approach provides a description of the network deformation. This information can be adopted to

estimate the Nuclear Aspect Ratio (NAR) of the cell population that can be incorporated into the scaffold following the procedure descried in chapter 3. Image processing on pictures produced by the FEM solver enables polygons complimentary to the space occupied by the fiber network to be detected (Figure 19). A polygons area range is selected to identify a subset of polygons representing areas occupied by cells embedded in the real micro integrated scaffold. An upper and lower area threshold are selected (< 1/10 average area cell > 10 average area cell). NAR delta – strain curve was obtained analyzing the NAR variation in subsequent strain configurations using the method presented in [64]. The prediction was compared with its corresponding experimental data [64].

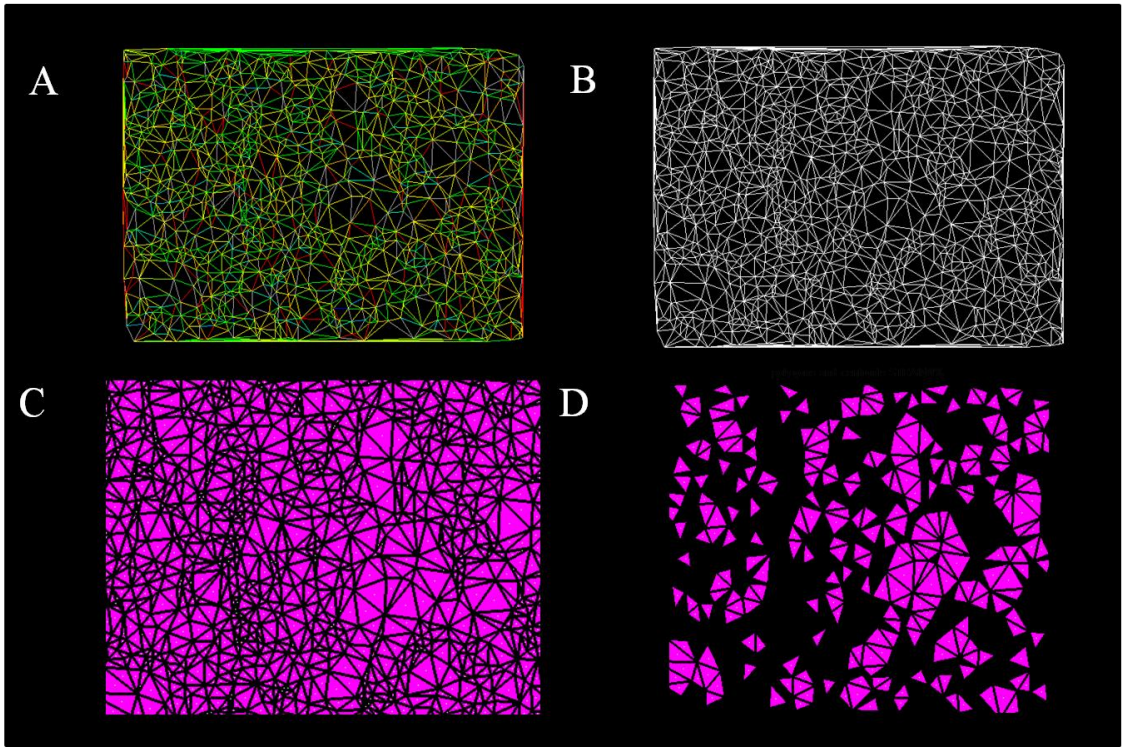

Figure 19. NAR-strain prediction method. (a) Original network representation produced by the FEM solver. (b-c) Polygons identified by the network topology a specific area range is selected to identify voids potentially able to host cells and to create mechanical coupling between the fibers network and the cells. (d) voids selected.

4.3 – BIOMECHANICAL MODEL OF ES-PEUU: RESULTS

4.3.1 – MACRO SCALE

The macro level mechanical response was produced for three scaffold family exhibiting increasing level of structural anisotropy (1.5 m/s, 4.5 m/s, 9.0 m/s). The model predictions (250 µm x 250 µm) for the engineering stress [Pa] vs. stretch were compared with the experimental data presented in chapter 3 (Figure 20).

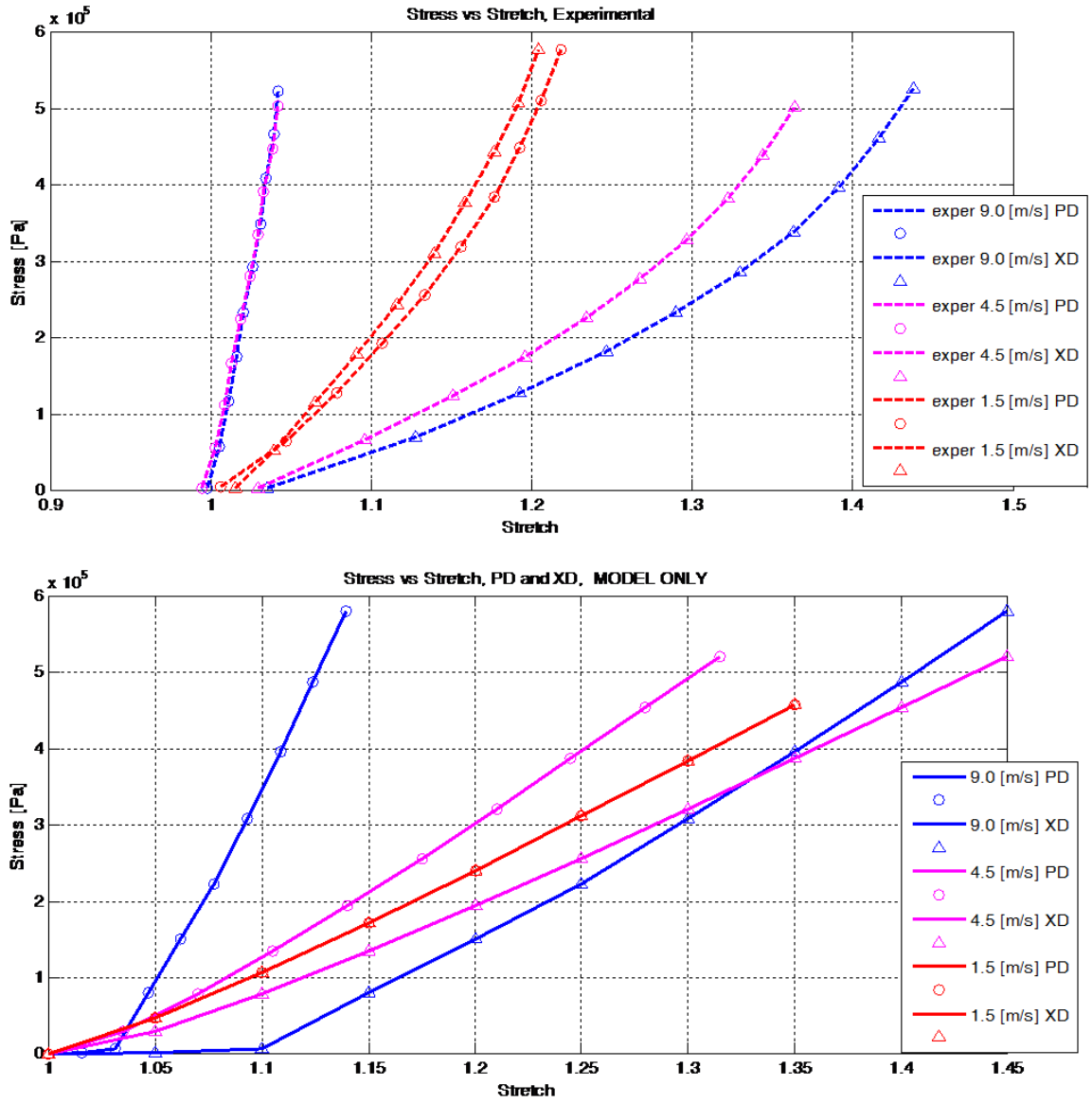

Figure 20. Engineering stress vs. stretch relationship for the three scaffold families (1.5 [m/s], 4.5[m/s], 9.0 [m/s]).

At macro level under applied deformation, the simulated fiber networks recapitulated the real scaffold anisotropy approximating the experimental biaxial data (Figure 20), since the model adopted a simple linear constitutive equation the structural anisotropy imposed by the material topology dictated the mechanical anisotropy shown

in the macro level mechanical response. The delay in the onset of the response, more evident in the 1.5 [m/s] scaffold is attributed to the gradual recruitment of the fibers.

Higher density in fiber intersection reported for the 4.5 and 1.5 [m/s] scaffolds mitigated this effect. Although the model is not able to fully capture the material non-linearity observed experimentally (Figure 20) the quality of the approximation is superior to comparable current attempts [116-117, 121, 125, 128] available in literature.

Material elasticity moduli were 11.25 MPa and 15.30 MPa, 131.85 MPa for the 1.5 [m/s], 4.5 [m/s] and 9.0 [m/s] ES-PEUU scaffolds respectively, the increase in the elasticity moduli is consistent with the increase in cristallinity observed for ES-PEUU. These values were compatible with single fibers elasticity moduli determined using Atomic Force Microscopy (AFM) on comparable materials.

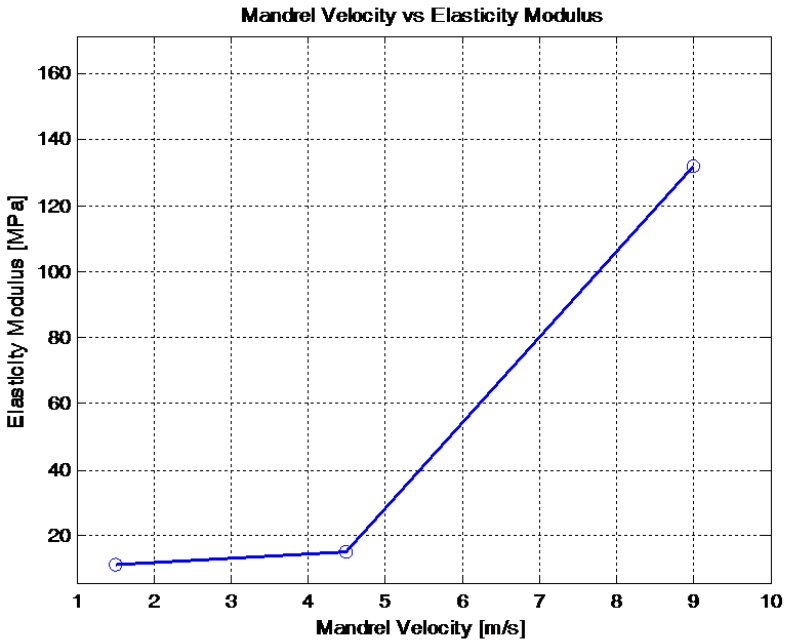

Figure 21. Elasticity moduli prediction for different mandrel velocity.

4.3.2 – MESO SCALE

At meso level under applied deformation, the simulated fiber networks showed long fibers behavior and loading pathways consistent with confocal microscopy experimental observation [64] (Figure 22-26). Contrary to the multi-scale approach the full solution adopted in this study enable to reproduce realistic fiber activation pattern spanning the full length of the model (Figure 22-24).

The stretch configurations emphasized the difference in material topology, the three networks have very similar connectivity plots characterized by a maximum at three, the 1.5 [m/s] scaffolds is the most dense in terms of fiber intersection density and the most isotropic. Conversely, the 9.0 [m/s] scaffold has the smallest number of fiber intersections per area and the highest level of alignment, the 4.5 [m/s] is characterized by intermediate values of fiber intersections and orientation index (Figure 22-26).

The simulated fiber networks showed, as expected, that the fibers aligned in the scaffold cross preferred direction undergo higher strain values (Figure 27), this strain pattern is responsible for the difference the bulk response at observed and predicted at macro level (Figure 20). The latter confirms once again that the material morphology dictates the anisotropy in the mechanical response.

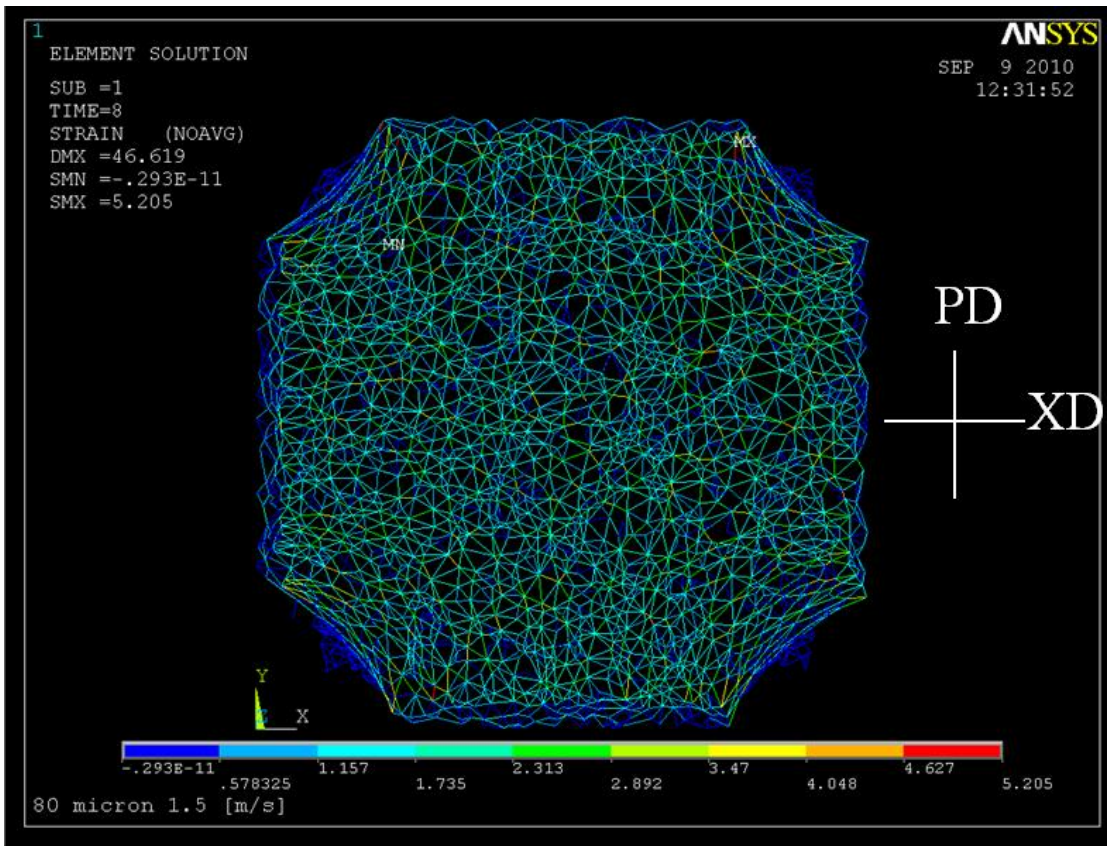

Figure 22. 1.5 [m/s] scaffold model, 80 µm, intersection density = 0.46, connectivity maximum at 3, fiber diameter 0.32 µm, orientation index = 0.5. Stress map at stretches $\lambda_{PD} = \lambda_{XD} = 1.35$.

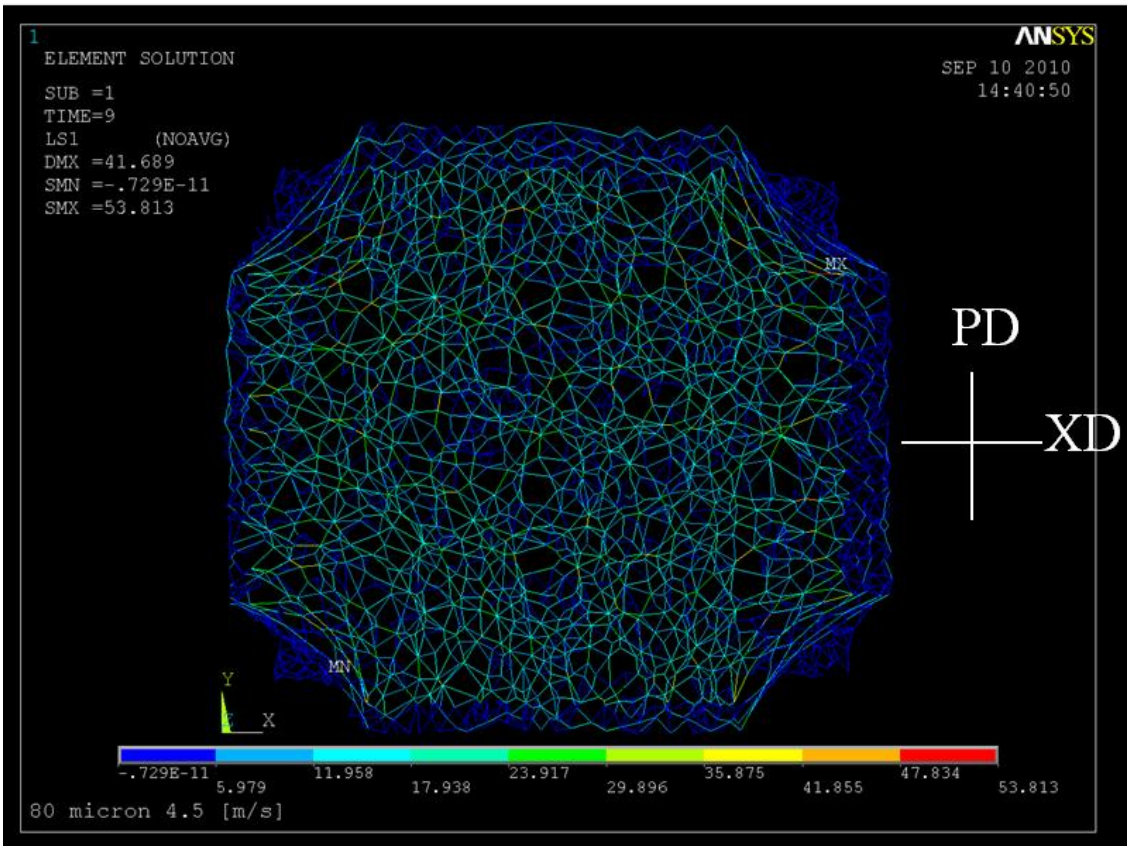

Figure 23. 4.5 [m/s] scaffold model, 80 µm, intersection density = 0.44, connectivity maximum at 3, fiber diameter 0.33 µm, orientation index = 0.52. Stress map at stretches λ_{PD}*= 1.32,* λ_{XD}*=1.45.*

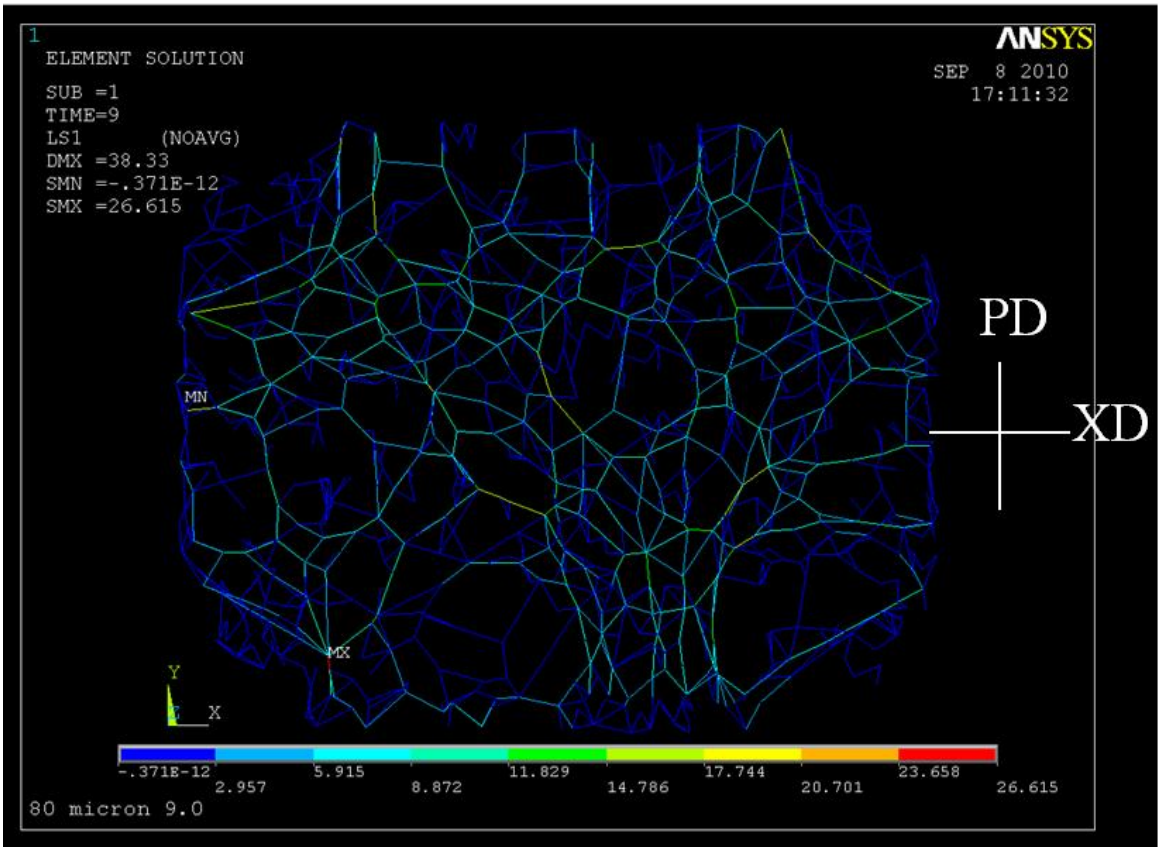

Figure 24. 9.0 [m/s] scaffold model, 80 µm, intersection density = 0.18, connectivity maximum at 3, fiber diameter 0.47 µm, orientation index = 0.65. Stress map at stretches λ_{PD}*=1.14,* λ_{XD}*=1.45.*

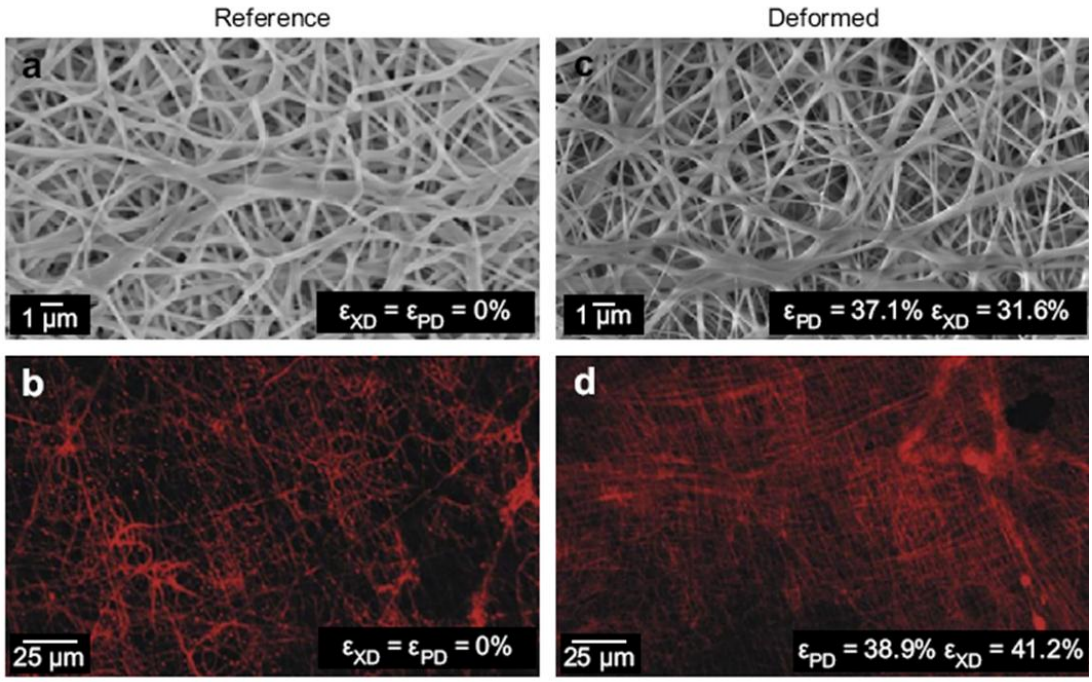

Figure. 25. Changes in ES-PEUU fiber microarchitecture under biaxial stretch. When strained equally in both orthogonal directions, PEUU fibers were observed by (a, c) SEM and (b, d) laser scanning confocal microscope to transition from a tortuous configuration in the unstrained state to an interconnected web-like architecture at a high strained state, adapted from [64].

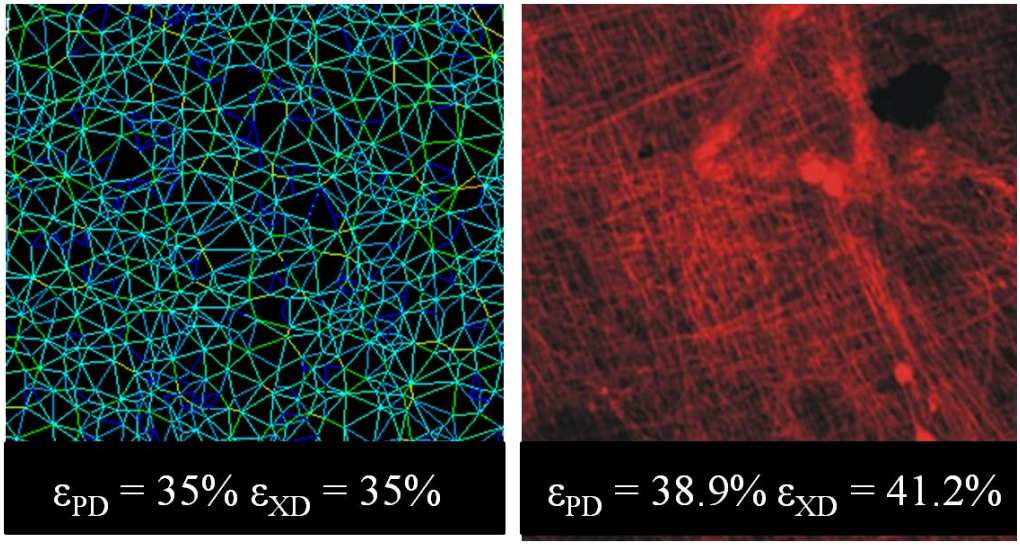

Figure 26. 1.5 [m/s] scaffold model, 80 µm, intersection density = 0.46, connectivity maximum at 3, fiber diameter 0.32 µm, orientation index = 0.5. Stress map at stretches $\lambda_{PD} = \lambda_{XD} = 1.35$.

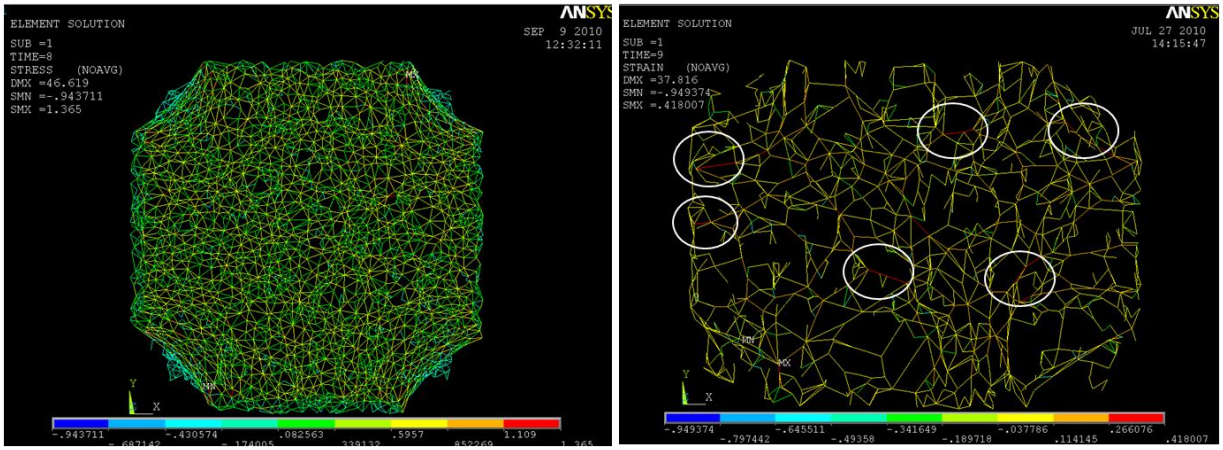

Figure 27. Strain map comparison: 1.5 [m/s] scaffold model (left) at $\lambda_{PD} = \lambda_{XD} = 1.35$, *9.0 [m/s] scaffold model (right) at* $\lambda_{PD} = 1.14$, $\lambda_{XD} = 1.45$. *While strain distribution is homogeneous in the isotropic scaffold the anisotropic one shows higher strain values mainly on fibers oriented over the XD (circles).*

4.3.3 – MICRO SCALE

In order to assess the model prediction capability at micro level the method illustrated in the previous paragraph (Figure 19) was adopted on an isotropic scaffold model simulating strip biaxial conditions (1.5 [m/s] scaffold model, 80 µm, intersection density = 0.46, connectivity maximum at 3, fiber diameter 0.32 µm, orientation index = 0.5). NAR delta – strain curve was obtained analyzing the NAR variation in subsequent strain configurations using a processing cascade that recapitulate the experimental method presented in [64]. The prediction was compared with its corresponding experimental data obtain on ES-PEUU scaffold micro-integrated with vascular smooth muscle cells [64] (Figure 28-30).

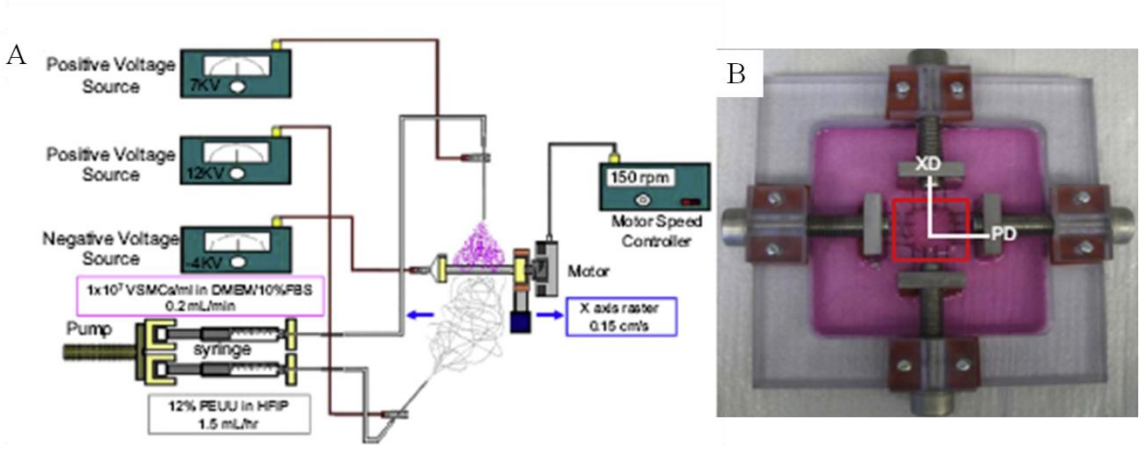

Figure. 28. Fabrication and imaging under biaxial stretch of cell integrated elastomeric scaffolds. a) Cell-scaffold constructs were fabricated via concurrently electrospinning polymer and electrospraying cells onto a 4.7 mm diameter rotating mandrel. b) Biaxial stretch device used to deform cell micro-integrated ES-PEUU scaffolds, adapted from [64].

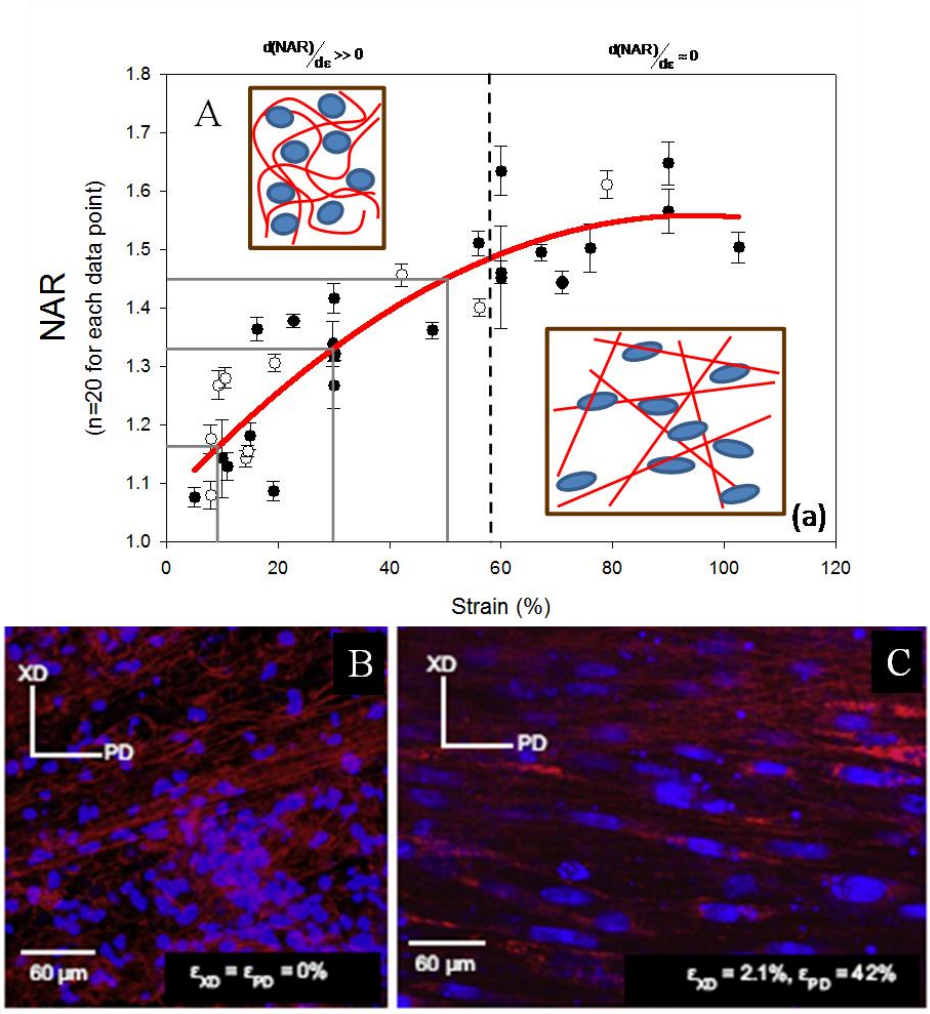

Figure 29. Change with deformation is closely related to local fiber microarchitecture. a) A composite of all NAR measurements (mean ± s.e.m) demonstrated a rapid increase to ≈60% strain, after which a plateau was observed with further strain increases, indicating that nuclei deformations are dominated by local fiber straightening. Here, the scaffold stretched either parallel (open symbols) or perpendicular (filled symbols) to the preferred fiber direction. A) and c) as the scaffold construct underwent strip biaxial deformation, the cell nuclei (blue) were observed to elongate and the fibers become straightened (red), adapted from [64].

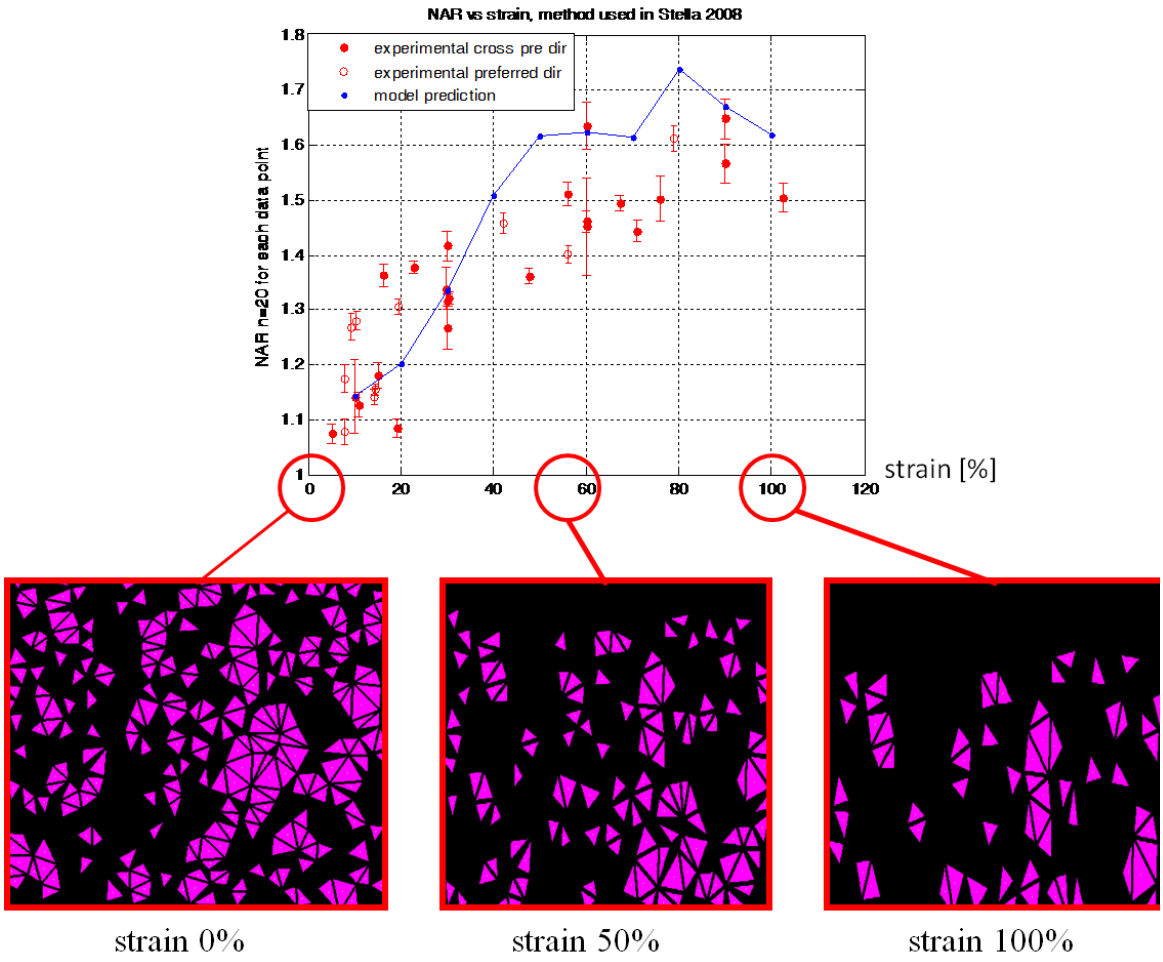

Figure 30. Model NAR vs. strain prediction for 1.5 [m/s] scaffold model under strip biaxial deformation. Red dots represent the experimental data from [64], model prediction in blue.

4.4 – BIOMECHANICAL MODEL OF ES-PEUU: CONCLUSION

As demonstrated in this chapter micro scale based mechanical models can be used to (1) guide tissue engineering scaffold design [55, 116-117, 121, 125, 128] (2) provide a better understanding of cellular mechanical and metabolic response to local micro-structural deformations [64, 123, 148] (3) investigate structural changes as a function of deformation across multiple scales [64, 148]. In spite of these potential the proposed approach presents several limitation: the computational cost of the structural deterministic method makes it feasible for two dimensional application only, finally the accuracy of the material topologic description strongly affects the quality of the result.

Cardiac valve and myocardium tissue regeneration has consistently raised interest for cell-seeded/micro-integrated planar fibrous scaffolds. Electrospun constructs, decellularized tissues and collagen/fibrin gels often show a similar microstructure mainly characterized by the fibrous network topology. Microstructure based mechanical modeling enable fabrication parameters and scaffolds morphology to be related to meso-macro mechanical response.

In summary:

- The developed image analysis and mesh generator algorithm produces a full description of the real fiber network geometry including connectivity information, and fiber orientation;

- The meso model formulation provides a tool to reproduce stochastically equivalent networks and to investigate through FEM simulation on the influence of each of the described micro-architectural features;
- The material bulk mechanical response can be predicted and related to a specific material morphology;
- At the micro scale NAR and stretch can be correlated to the material micro architecture;
- FEM results revealed the capacity of the approach to characterize the material mechanical behavior at the fiber level, where interaction between the fiber and the cells occur and at the global level, where bulk material properties are fundamental to guide and assist scaffold design.

The next chapter describe the animal model adopted and the three weeks explants result providing evidence of the feasibility of the suggested paradigm.

CHAPTER 5

ES-PEUU FOR TISSUE ENGINNERING HEART VALVES: A FEASIBILITY STUDY ON AN IN VIVO LARGE ANIMAL MODEL

5.1 – MATERIAL SELECTION

This last chapter discusses the in vivo testing of the Electrospun PEUU mats characterized and modeled in this work. A three weeks study was performed on an ovine model to assess surgical feasibility and valve functionality of ES-PEUU scaffolds. As illustrated in the third chapter previous studied demonstrated superior performances and in vivo remodeling capacity of wet Electrospun polyurethanes. In addition, combining low rastering and madrel velocity enable both physiologically relevant level of anisotropy to be achieved. The complete surgeries plan includes the testing of three groups: (I) dry Electrospun, (II) media wetted, (III) cell micro-integrated. The feasibility study presented in this thesis involves the first group only. PEUU scaffold produced at 0 [m/s] rastering speed and 12 [m/s] mandrel velocity were selected as the most suitable to duplicate the native pulmonary planar response (Figure 1).

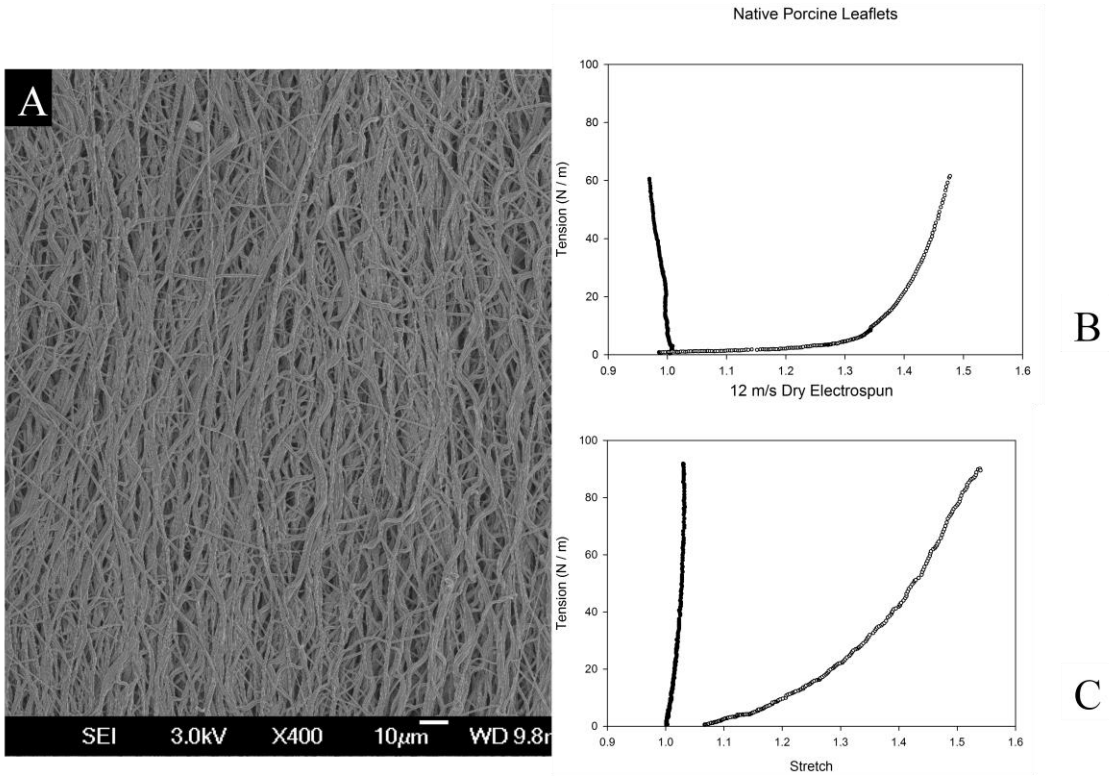

Figure 1. a) SEM images of the 12 [m/s] scaffold implanted, b) native porcine pulmonary valve leaflets mechanical response membrane tension vs stretch, c) scaffold mechanical response showing the remarkable capacity of this material to recapitulate the native pulmonary valve planar behavior.

Scaffold suture retention properties were tested on MTS machine displacement control (12.5 mm/s) on MTS Tytron™ 250 MicroForce Testing System for high speed uniaxial testing, showing failure at membrane tension ~ 75 [N/m] (Figure 2 c, n=5). The latter value is less than the predicted physiological value of 26 [N/m]. This value in turn was estimated by Laplace Law using anatomical and in vivo measurements, given a Right Ventricular Outflow Tract (RVOT) diastolic back pressure gradient of 15-20 mmHg.

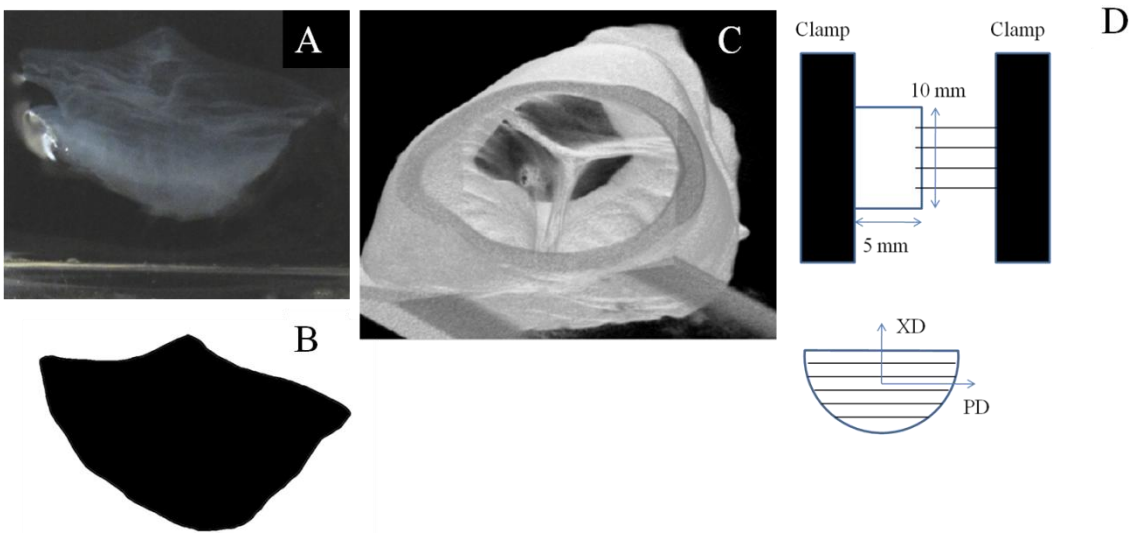

Figure 2. a) CT scan of native ovine pulmonary valve at fully loaded configuration ~15-20 mmHg, b) segmentation of a) for radial and circumferential measurement and PEUU leaflet sizing c) rendering of the pulmonary valve and conduit, d) layout adopted for the suture retention test, spacing between sutures was 2 mm.

Leaflet shape and size was approximated from CT scan images post – processing of ovine native pulmonary valve at the fully loaded configuration ~15-20 mmHg. Radial and circumferential sizes at unloaded configuration were determined scaling the measurements operated on CT scan data (Figure a, b)using the equibiaxial data (Figure 1 d).

5.2 – IN VIVO ANIMAL MODEL AND SURGERY

The surgery was conducted at the Children Hospital Boston, the Harvard Medical School by Prof J E Mayer's group. The lamb was sedated with ketamine and versed.

She was given inhalation of isoflurane by mask and subsequently intubated with a 8-0 endotracheal tube with balloon attachment protected. She was shaved around the neck and left side of the chest and abdomen and scrubbed with betadine. Two back leg peripheral catheters were placed. 1st dose solumedrol was given shaved around the neck and left side of the chest and abdomen. The operating table was elevated and the surgeons stood on a step stool in anticipation of the need to augment venous drainage.

She moved from the induction area to the operating table. The table was elevated and we stood on stepstools in anticipation of improved venous drainage via gravity. An 8 Fr Foley's catheter was placed in the urethra of the sheep. An initial 30 ml of urine output was returned. A 5 Fr. Catheter was placed in the left femoral artery. And 7 Fr. Catheters were placed in both the left femoral vein and the right internal jugular vein. She was then placed in the right lateral decubitus position, prepped and draped from neck to mid-abdomen. 1L of LR was given via back leg. The animal was maintained under general anaethesia using desflurane, fentanyl and versed. Ancef and baytril were given. Repeat dose of Sulmedrol was given after initiation of CPB. Cisatracurium was given x2 doses to achieve muscular paralysis.

The chest was prepped and draped in a sterile fashion. A left 4th interspace mini-thoracotomy was made. The muscle was divided in layers using electrocautery. Care was taken to preserve the long thoracic nerve. The chest was entered through the superior border of the fifth rib. The rib retractor was placed. For exposure the left lung was packed by sterile gauze. The left internal thoracic artery and vein were not divided. The pulmonary artery was well visualized.

A 24 single stage venous canula was used to canulate the right atrial appendage. A 16 Fr arterial canula was placed in the left carotid artery. This didn't maintain enough venous drainage so a 24 Fr. Single stage venous canula was placed in the left jugular vein, which provided a better visualization of the field.

A transverse incision was made at superior border of the sinus of the pulmonary root to reach the anterior pulmonary leaflet. The native leaflet was excised and the engineered leaflet was sawn in place. Intra-operative echocardiography was then performed. Short and long axis views of the 3 leaflets functioning were captured.

The sheep was then weaned of CPB. Haemostasis was maintained. The neck was closed in layers over the carotid artery using 3-0 vicryl sutures and skin was closed using 4-0 PDS sutures. Two chest tubes were placed, one in the anterior mediastinum and one in the left pleural space. Rib retractors were then removed and ribs were approximated using 0 vicryl sutures, then the chest were closed in layers using 3-0 vicryl. Skin was then closed using 4-0 PDS sutures. Urine output was adequate and the sheep was haemodynamically stable.

Post-operatively femoral arterial and venous lines were removed; compression over the femoral artery was maintained for 15 minutes to prevent femoral bleeding. Sheep was weaned of Desflurane, extubated, back leg line was removed. She was then put back in cage at 6:30 pm.

5.3 – EXPLANT ANALYSIS

The animal was scarified at week three and the full heart and the TEHV leaflet (Figure 3-4) was explanted and analyzed with conventional Hematoxylin and Eosin (H&E) staining.

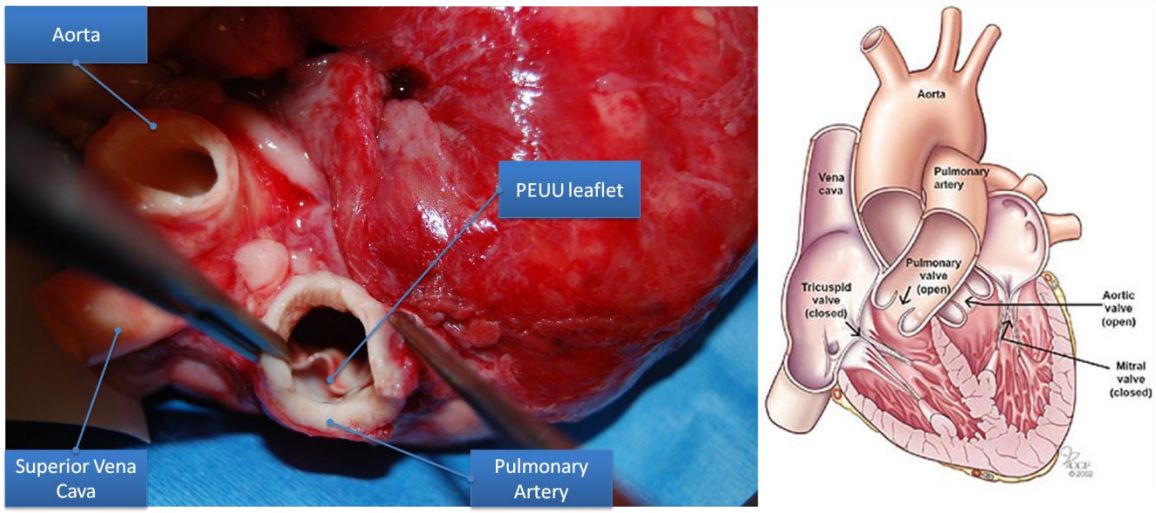

Figure 3. Anterior (Frontal) view of the explanted sheep heart showing aorta, superior vena cava, and pulmonary artery leading to the TE leaflet.

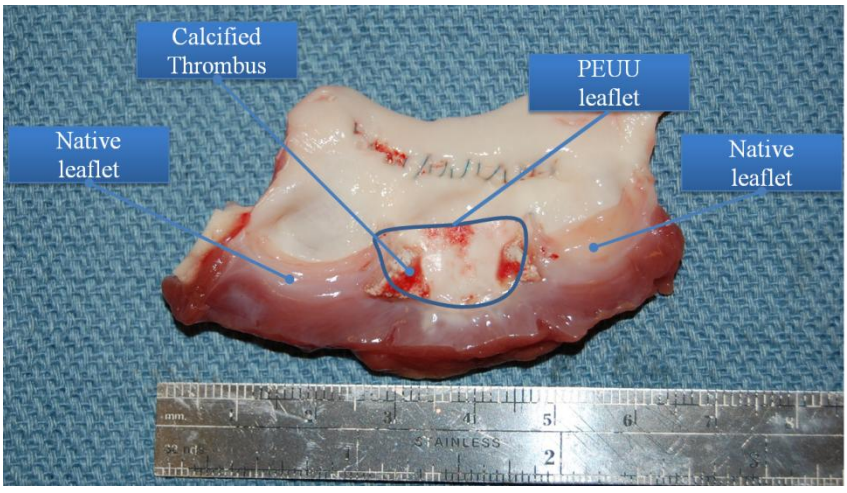

Figure 4. Explanted pulmonary valve and conduit, arrows highlight native , TE leaflets and calcified thrombi at each the TE leaflet commisures.

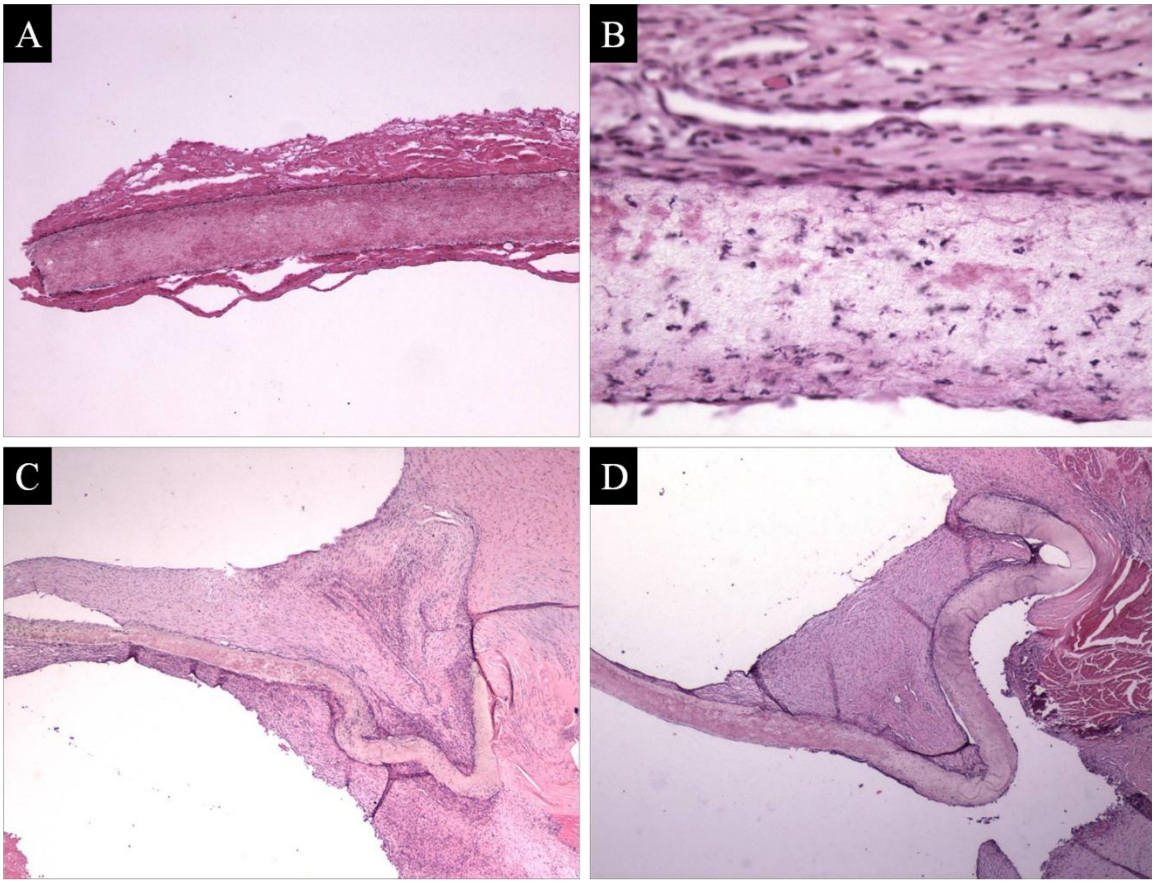

Figure 5. H&E staining of the TE leaflet. A) minimal thrombus at the edges. B) cellular infiltration at three weeks. C-D) calcified thrombus on the commisures.

The described preliminary surgical feasibility study demonstrated that the single leaflet implantation model is currently established and should be reproducible for the remaining animals. Valve Functionality was assessed throughout intra operative echocardiography: implanted leaflet showed restricted mobility as compared to the opposing native leaflets. In spite of this, the regurgitant fraction across the pulmonary valve was negligible. The material was intact at ex-plant with no sloughing. Two

irregular, calcified thrombi were detected one at each commisure of the implanted leaflet (Figure 4, 5 c-d). The thrombi were also detected during the echo analysis. Cellular infiltration shown by histology demonstrated tissue overgrowth from host (Figure 5 a-b). The latter was, as expected, minimal at the co-aptation region of the ventricular surface of the implanted leaflet. Scaffold was homogenously cellularized at some regions after 3 weeks of implantation (Figure 5 b).

It can be speculated that thrombi formation at the commisure regions is due to unnatural hemodynamic that in turn is produced by non-physiological bending properties of the PEUU scaffold. This hypothesis is supported by the intra-operative echocardiography data. The scaffold bending stiffness (or flexural rigidity) is given by $EI = E \cdot (bh^3/12)$ where b and h are the transverse section scaffold dimensions. Thus, EI can be reduced modifying the second moment of inertia or reducing E. The material characterization described in chapter three suggests that in plane properties can be maintained changing the fiber intersection density. Bending test on ES-PEUU showed a nice correlation between the fiber intersection density and the elasticity modulus. Hence, for a given biaxial response, different values of EI can be obtained changing the rastering velocity (Figure 6).

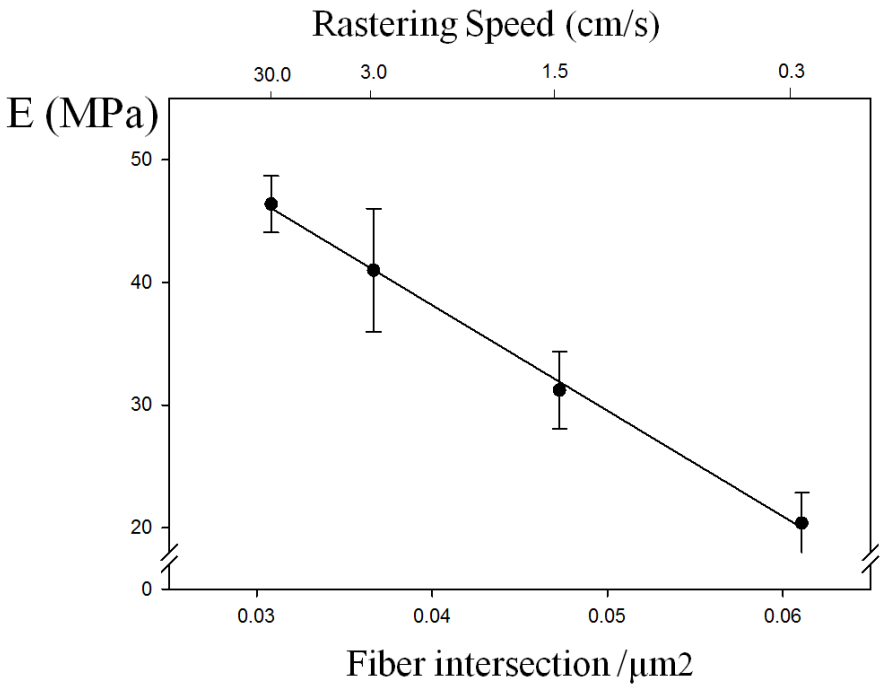

Figure 6. Increasing the translational velocity of the mandrel during fabrication resulted in a lower density of fiber intersections, without any significant change in fiber orientation (data not shown). Unexpectedly, the constructs that possessed lower fiber intersection densities were associated with stiffer responses to bending. When PEUU was concurrently electrospun with PEO at the identical rotational, and translational velocities, a lower bending modulus is observed (15 vs. 20 kPa, $p<.05$).

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 – USING THE DEVELOPED STRUCTURAL DETERMINISTIC MODEL TO ASSIST MATERIAL DESIGN AND FABRICATION FOR SOFT TISSUE ENGINEERING APPLICATIONS

The in vivo - study presented in chapter 5 demonstrated the feasibility of the adoption of ES-PEUU for TEHVs and more generally this material potential for soft tissue regeneration. The developed image analysis and design strategies enable the material topology to be both quantified and reproduced. Material fabrication parameters can be related to material micro-architecture (chapters 2 and 3). Similarly, the micro-architecture can be related to (I) macro scale mechanical responses such as in-plane reactions or flexural rigidity (chapters 3 and 4), (II) meso – micro scales mechanisms such NAR – stretch relation. The developed paradigm will be utilized to guide further surgeries aiming to identify the optimal material (dry, media wetted, cell micro-integrated scaffold) for TEHVs. A FEM models library will enable to associate a targeted NAR range and membrane tension vs. stretch response with a specific micro-structures. The latter was related by the fabrication study (chapter 3) to fabrication parameters. In conclusion, the modeling paradigm introduced in this work bridges for the first time the fabrication parameters with the mechanical response at different scale length (Figure 7).

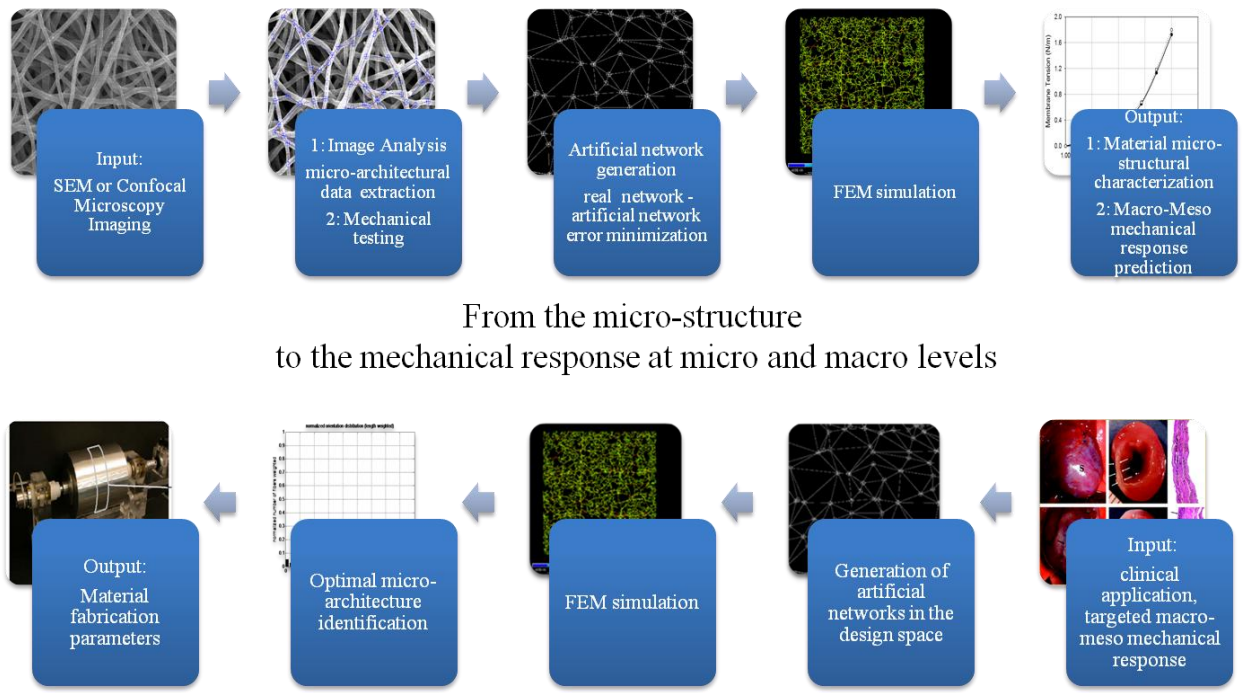

Figure 1. Full schematic of the model paradigm introduced in this thesis. From left to right, top line: image analysis, material mechanical characterization and micro-architectural data extraction (chapters 2-3), artificial network generation, FEA and macro – meso mechanical response prediction (chapter 4). Bottom line the starting point is the targeted mechanical response across the scales (chapter 5) optimal in plane, out of plane and NAR range are identified with respect to a specific clinical application. The structure able to generate the desired response is identified from the FE and network models library. Finally, the identified optimal material characteristics are associated to the fabrication parameters.

6.2 – FUTURE WORKS

In order to fully benefit from the described mechanical model the presented fabrication study needs to be extended to confirm the possibility to independently control scaffold fiber alignment and fiber intersection density throughout the mandrel velocity and the rastering velocity. The proposed study aiming to develop further electrospinning master curves is describes as follows (Table 1):

2 fabrication inputs:

- mandrel velocity;
- rastering velocity;

4 micro – structural outputs:

- the density of the fiber intersections;
- the connectivity distribution;
- fiber orientation angle distribution;
- fiber diameter distribution;

X	*Y*	*Z*
mandrel velocity	rastering velocity	density of the fiber intersections
mandrel velocity	rastering velocity	max of the connectivity distribution
mandrel velocity	rastering velocity	orientation index for fiber orientation angle distribution
mandrel velocity	rastering velocity	mean fiber diameter
Table 1, fabrication master curves		

Optimal NAR range for pediatric heart valves must be identified. The guiding criteria would be identifying the NAR range that produces, for a given physiological load, robust ECM formation. Once the NAR range and the desired bulk mechanical

response are known the correspondent structure and fabrication parameters will be identified by the model.

Finally, in vivo/vitro validation on TEHVs or cardiac patch will test the hypothesis according to which structurally optimized scaffolds able to recapitulate both macro and micro response at cell level perform better in vivo. Current in Vivo TEHV project results will suggest the most adequate paradigm within the followings: A) acellular scaffold (media wetted ES-PEUU) or B) micro-integrated (smooth muscle derived stem cells).

The final goal will be demonstrate that innovative materials that can be tailored on a specific clinical scenario knowing a priori the organ level mechanical response and the optimal micro mechanical environment. This research paradigm seems to be applicable to soft tissues other than cardiovascular soft tissues, a simple schematic of a potential in vivo validation study is proposed as conclusion of this section. The ideal application are the pediatric TEHV and the cardiac patch groups are defined in table 2. Comparing A (control) with previous results on non woven will reveal the benefit to adopt an elastomeric scaffold, comparing A (control) with B (Macro optimized) will reveal the benefit of controlling anisotropy and bulk macro response and finally comparing B (Macro optimized) and C (Macro Micro - optimized) will reveal the benefit of fully control the structure in a tissue engineered construct.

A) n=6 leaflets, 2 valves	**B)** n=6 leaflets, 2 valves	**C)** n=6 leaflets, 2 valves	**TOTAL surgeries**
Control Group ES-PEUU random scaffold. Not structurally optimized.	**Macro Optimized** ES-PEUU optimized with respect to the macro mechanical response: • recapitulate membrane tension vs. stretch native tissue curve;	**Macro-Micro Optimized** ES-PEUU optimized with respect to the macro and micro mechanical response: • recapitulate membrane tension vs. stretch native tissue curve; • recapitulate native tissue micro-mechanical environment (optimal NAR range under physiological load);	A) 3-6 B) 3-6 C) 3-6 Min: 9 Max:18

Table 2. Groups proposed for in vivo validation of the reseach hypotesis: "A structurally optimized scaffold able to recapitulate both macro and micro response at cell level perform better in vivo"

6.3 – CONCLUSION

This doctoral studies focused on the development and experimental validation of a structural deterministic modeling strategy to: A) guide tissue engineering scaffold design, B) provide a better understanding of cellular mechanical and metabolic response to local micro-structural deformations. Targeted clinical application was the

pulmonary heart valve. Electrospinning was selected as the optimal platform technology to implement, validate and test the presented designing strategy.

An innovative custom made software was developed and tested on Electrospun poly (ester urethane) urea scaffolds (ES-PEUU), decellularized native tissues and collagen gels to fully characterized engineered constructs morphology. The image analysis methodology extracted the following architectural descriptors: (1) fiber angle distribution, (2) fiber intersection density, (3) fiber connectivity , (4) fiber diameter, (5) scaffold porosity, (6) scaffold pores distribution. These structural information were adopted to feed and assist the mechanical modeling

Two previously unevaluated fabrication modalities were investigated throughout both mechanical testing and image analysis in order to explore further how the electrospinning fabrication process can alter the structure and mechanical response: (I) variation of mandrel translation velocity and (II) concurrent electrospraying of cell culture medium with or without cells or rigid particulates. These fabrication parameters were studied to enrich control in the electrospinning process.

The detected material topology and mechanical equi-biaxial data were adopted to generate statistically equivalent scaffold mechanical models. The structural determinist approach was applied to ES-PEUU scaffolds, validated and mechanical response at organ and cell level was produced through FEM simulation. Prediction included: membrane tension vs. stretch relation, elasticity moduli, Nuclear Aspect Ratio vs. stretch relation for the cells micro-integrated into the scaffold.

A three weeks in vivo - study on an ovine model was performed to demonstrated the feasibility of the adoption of ES-PEUU for TEHVs and more generally this

material potential for soft tissue regeneration. Explants analysis showed surgical feasibility and acceptable valve functionality.

The developed design strategies combining image analysis and structural deterministic modeling enabled the material topology to be both quantified and reproduced. Material fabrication parameters were related to material micro-architecture Similarly, the micro-architecture was related to macro scale mechanical responses such as in-plane reactions or flexural rigidity, and complex meso – micro scales mechanisms such NAR – stretch relation. In conclusion, the modeling approach introduced in this work bridges for the first time the scaffold fabrication parameters with the mechanical response at different scale length. The developed paradigm will be utilized to identify the optimal scaffold for a given soft tissue engineering application.

APPENDIX

APPENDIX A) : IMAGE ANALYSIS CODE (MATLAB)

```
node_detector_main_body_NEW8_NI

network_generator_full_general_1_1


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

%%%  NEW IMPLEMENTATION COMPLETE

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% CLEAN WSPACE & CW
clc
clear
close all
warning off all;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% LAYER SELECTION

[original_image,thre_image,layer_det,FILTER,FILTER2,T,fiber_diameter,D
,lastfilt,S] = layer_detector_fast_NEW15;



function
[original_image,thre_image,layer_det,FILTER,FILTER2,T,fiber_diameter,D
,lastfilt,S] = layer_detector_fast_NEW15;
% %
[original_image,thre_image,layer_det,FILTER,FILTER2,T,fiber_diameter,D
,lastfilt,S] = layer_detector_fast_NEW15;


%%%%%%%%%%%%%%%%%%%%%%%%%%%
% load data
warning off all


% validation pictures
% S=imread('1.5 m.s Reference x3500 BottomRight - crop.tif');  %
isotropic
% S=imread('9.0 m.s Reference x2500 #9_crop.tif');  % anisotropic
```

```
disp('please select the image file to process')

posfile=uigetfile({'*.jpg;*.tif;*.png;*.gif','All Image Files';...
          '*.*','All Files' },'mytitle',...
          'C:\Work\myfile.jpg');

save('current_test_name','posfile');

disp('please select the area to crop using the mouse, right click to
confirm the area selection')
S0=imread(posfile);
S = imcrop(S0);

% equalization to extend the dynamic range

 A=S;
%A = rgb2gray(S);
figure(1);
imshow(A);
title ('starting image');

A= histeq(A);
% figure(2);
% imshow(A);
% title('equalized image')


% median filtring to reduce the noise from equalization
gmed=medfilt2(A);
% figure(3);
% imshow(gmed);
A=gmed;

% figure(3)
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% interactive fiber_diameter detection
% scale = input('please provide the magnification   ');
disp ('please select a representative fiber diameter')
disp ('left click for initial point, right click for ending point')
%
[ai bi]=size(A);
zoom on
[c1,r1,P1] = impixel(A(1:ai/5,1:bi/5));

% [c1,r1,P1] = impixel(A(1:round(5*scale/100),1:round(5*scale/100)));
% [c1,r1,P1] = impixel(A);

title('median filter')
box = ((c1(2)-c1(1))^2)+((r1(2)-r1(1))^2);


fiber_diameter=round(sqrt(box));
disp(fiber_diameter);
```

```
fiber_diameter = input('please confirm or change the fiber_diameter
value, type an integer   ');
% round to the nearest even integer
D = round(1*fiber_diameter);
disp ('representative fiber diameter(pixels)selected:')
disp(fiber_diameter);

original_fiber_diameter=fiber_diameter;


close all
for optidia=1:11
       if optidia>1
          A=gmed;
          fiber_diameter=round(fiber_diameter - ((optidia-
1)/10)*(original_fiber_diameter/4));
       end
       if fiber_diameter>0
                fiber_diameter_test(optidia)=fiber_diameter;
%                 optidia

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
                % SCAFFOLD LAYER DETECTION OPTIMAL FD DETECTION



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
                % Local Thresholding with Otsu Method, subareas
controlled by fiber
                % diameter


                % subsets controlled by the fiber diameter
                [ai bi]=size(A);
                stepT=round(original_fiber_diameter*10);
                stepx=stepT; % column
                stepy=stepT*(ai/bi); %raw

                k=1;
                    for startY=1:stepy:ai-stepy-1
                        for startX=1:stepx:bi-stepx-1

subA(:,:,k)=A(startY:startY+stepy,startX:startX+stepx);
                            % median filetring on subareas
                            subA(:,:,k)=medfilt2(subA(:,:,k),[7 7],
'symmetric');
                            % Otsu thresholding on subareas
                            t = graythresh(subA(:,:,k));
                            T(k) = t*255;
                            subB(:,:,k)=subA(:,:,k)>T(k);
```

```
TOTrenew(startY:startY+stepy,startX:startX+stepx)=subB(:,:,k);
                            k=k+1;
                        end
                    end
%                   size(TOTrenew)
                % size re-definition for the starting image, the size
is now reduced due to
                % the local thresholding

                    B=A(1:startY+stepy,1:startX+stepx);
                    clear A subA subB
                    A=B;
                    clear B
                    original_image = A;

                % % Visualization check of subareas uncomment if
required
                % for j=1:k-1
                %     figure(j)
                %     imshow(subB(:,:,j));
                % end
                %
                % figure(j+1)
                % imshow(TOTrenew);
                % % T

                g=TOTrenew;
                clear TOTrenew
        %         figure (4);
        %         imshow(g);
        %         title ('locally thresholded image');
                % disp ('T used, Otsu method')
                % T;
                thre_image = g;
                %%%%%%%%%%%%%%%%%%%%%%%%%
                % Morphological processing

                % figure(5)
                Z = bwmorph(g,'thin');
                Z1 = bwmorph(Z,'majority');
                Z2 = bwmorph(Z1,'clean');
                % imshow(Z2)
                % title('thin, majority, clean')


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
                %operations dependent on fiber diameter

                %originally was 1/8
                % figure(6)
                se5_dim=round(fiber_diameter*1/6);
                % erode 12.5%
                se5 = strel('disk',se5_dim);
                erodedZ2 = imerode(Z2,se5);
```

```
                % imshow(erodedZ2)
                % title('diameter dependent erosion')

        %         close all

                %elimate everything smaller than 600x fiber diameter
originally was 600x

                % considering an 10 aggregated objects 20 fiber
diameters long
                % figure(7)
                Area_away=10*fiber_diameter*20;
                bwAreaOpenBW = bwareaopen(erodedZ2,Area_away);
                % imshow(bwAreaOpenBW)
                % title('remove small areas')

                %originally was 3/8
                % figure(8)
                se6_dim=round(fiber_diameter*1/3);
                % dilate 37.5%
                se6 = strel('disk',se6_dim);
                last = imdilate(bwAreaOpenBW,se6);
                % imshow(last)
                % title('diameter dependent dilatation')


                %originally was 2/8
                % figure(9)
                se7_dim=round(fiber_diameter*1/6);
                % erode 25%
                se7 = strel('disk',se7_dim);
                last1 = imerode(last,se7);
                % imshow(last1)
                % title('diameter dependent last erosion')
                FILTER=last1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
                % skeletonization
                % figure(10)
                last2 = bwmorph(last1,'shrink',Inf);
                % imshow(last2)
                % title('shrink')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%

                % BW filter produced
                [Aalt,Balt]= size(A);
                FILTER2 = last2;
                [Ablt,Bblt]= size(FILTER2);
                Amin=min([Aalt,Ablt]);
```

```
                Bmin=min([Balt,Bblt]);

                % last2=last2(1:Aalt,1:1:Balt);

                filtered_grey =
immultiply(A(1:Amin,1:1:Bmin),FILTER2(1:Amin,1:1:Bmin));
                layer_det = filtered_grey;
                % figure (11)
                % imshow(filtered_grey);
                % title ('filtered layer');


                Aul(:,:,1)=original_image;
                Aul(:,:,2)=original_image;
                Aul(:,:,3)=original_image;
                [a b]=size(FILTER2);
                for i=1:a
                    for j=1:b
                        if FILTER2(i,j)>0
                        Aul(i,j,1)=0;
                        Aul(i,j,2)=0;
                        Aul(i,j,3)=255;
                        end
                    end
                end


                figure(optidia)
                imshow(Aul);
                title('original + detected')
                clear Aul g
                sefilt = strel('disk',round(fiber_diameter/4));
                lastfilt = imdilate(last2,sefilt);
        end

end

bestskel = input('please select the smallest number that produces a
reasonable skeletonization (from image 1 to 10)    ');

fiber_diameter=fiber_diameter_test(bestskel);
D = round(1*fiber_diameter);
close all



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
        % SCAFFOLD LAYER DETECTION FINAL EVALUATION WITH OPTIMAL
        % REPRESENTATIVE DIAMETER
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
        % Local Thresholding with Otsu Method, subareas controlled by
fiber
        % diameter

        A=gmed;
        % subsets controlled by the fiber diameter
        [ai bi]=size(A);
        stepT=round(original_fiber_diameter*10);
        stepx=stepT; % column
        stepy=stepT*(ai/bi); %raw


        k=1;

            for startY=1:stepy:ai-stepy-1
                for startX=1:stepx:bi-stepx-1

subA(:,:,k)=A(startY:startY+stepy,startX:startX+stepx);
                    % median filetring on subareas
                    subA(:,:,k)=medfilt2(subA(:,:,k),[7 7],
'symmetric');
                    % Otsu thresholding on subareas
                    t = graythresh(subA(:,:,k));
                    T(k) = t*255;
                    subB(:,:,k)=subA(:,:,k)>T(k);

TOTrenew(startY:startY+stepy,startX:startX+stepx)=subB(:,:,k);
                    k=k+1;
                end
            end

        % size re-definition for the starting image, the size is now
reduced due to
        % the local thresholding
        B=A(1:startY+stepy,1:startX+stepx);
        clear A
        A=B;
        clear B
        original_image = A;

        % % Visualization check of subareas uncomment if required
        % for j=1:k-1
        %     figure(j)
        %     imshow(subB(:,:,j));
        % end
        %
        % figure(j+1)
        % imshow(TOTrenew);
        % % T

        g=TOTrenew;
```

```
%         figure (4);
%         imshow(g);
%         title ('locally thresholded image');
        % disp ('T used, Otsu method')
        % T;
        thre_image = g;
        %%%%%%%%%%%%%%%%%%%%%%%%%
        % Morphological processing

        % figure(5)
        Z = bwmorph(g,'thin');
        Z1 = bwmorph(Z,'majority');
        Z2 = bwmorph(Z1,'clean');
        % imshow(Z2)
        % title('thin, majority, clean')


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
        %operations dependent on fiber diameter

        %originally was 1/8
        % figure(6)
        se5_dim=round(fiber_diameter*1/6);
        % erode 12.5%
        se5 = strel('disk',se5_dim);
        erodedZ2 = imerode(Z2,se5);
        % imshow(erodedZ2)
        % title('diameter dependent erosion')

        close all

        %elimate everything smaller than 600x fiber diameter
originally was 600x

        % considering an 10 aggregated objects 20 fiber diameters long
        % figure(7)
        Area_away=10*fiber_diameter*20;
        bwAreaOpenBW = bwareaopen(erodedZ2,Area_away);
        % imshow(bwAreaOpenBW)
        % title('remove small areas')

        %originally was 3/8
        % figure(8)
        se6_dim=round(fiber_diameter*1/3);
        % dilate 37.5%
        se6 = strel('disk',se6_dim);
        last = imdilate(bwAreaOpenBW,se6);
        % imshow(last)
        % title('diameter dependent dilatation')


        %originally was 2/8
        % figure(9)
        se7_dim=round(fiber_diameter*1/6);
```

```
        % erode 25%
        se7 = strel('disk',se7_dim);
        last1 = imerode(last,se7);
        % imshow(last1)
        % title('diameter dependent last erosion')
        FILTER=last1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
        % skeletonization
        % figure(10)
        last2 = bwmorph(last1,'shrink',Inf);
        % imshow(last2)
        % title('shrink')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

        % BW filter produced
        [Aalt,Balt]= size(A);
        FILTER2 = last2;
        [Ablt,Bblt]= size(FILTER2);
        Amin=min([Aalt,Ablt]);
        Bmin=min([Balt,Bblt]);

        % last2=last2(1:Aalt,1:1:Balt);

        filtered_grey =
immultiply(A(1:Amin,1:1:Bmin),FILTER2(1:Amin,1:1:Bmin));
        layer_det = filtered_grey;
        % figure (11)
        % imshow(filtered_grey);
        % title ('filtered layer');


        Aul(:,:,1)=original_image;
        Aul(:,:,2)=original_image;
        Aul(:,:,3)=original_image;
        [a b]=size(FILTER2);
        for i=1:a
            for j=1:b
                if FILTER2(i,j)>0
                Aul(i,j,1)=0;
                Aul(i,j,2)=0;
                Aul(i,j,3)=255;
                end
            end
        end

        figure(1)
        imshow(Aul);
        title('original + detected')
```

```
        sefilt = strel('disk',round(fiber_diameter/4));
        lastfilt = imdilate(last2,sefilt);


% fiber_diameter=original_fiber_diameter;
% D = original_fiber_diameter;

original_image_copy=original_image;
representative_fiber=D;
close all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% LAYER ANALYSIS

layer_analyser_fast_NEW5_NI;




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%CONSTANTS DEFINITION AND INTERACTIVE AREA SELECTION
A = layer_det;

% % interactive area selection
% disp ('please define the area to inspect:')
% disp ('area shape = square')
% disp ('size = left click for top left, right click for bottom right
corners')
% [c,r,P] = impixel(A);
[as bs]=size(A);
r(1)=1;
r(2)=as;
c(1)=1;
c(2)=bs;
B = A;
xini=c(1);
xend=c(2);
yini=r(1);
yend=r(2);
B(r(1),c(1):c(2))=256;
B(r(2),c(1):c(2))=256;
B(r(1):r(2),c(1))=256;
B(r(1):r(2),c(2))=256;

% for final fisualisation image 16
H=B;
v(1)=yini+D;
v(2)=yend-D;
p(1)=xini+D;
p(2)=xend-D;
H(v(1),p(1):p(2))=256;
H(v(2),p(1):p(2))=256;
H(v(1):v(2),p(1))=256;
```

```
H(v(1):v(2),p(2))=256;
% figure (4)
% imshow(H)

% for final fisualisation image 17
original_image_copy(v(1),p(1):p(2))=256;
original_image_copy(v(2),p(1):p(2))=256;
original_image_copy(v(1):v(2),p(1))=256;
original_image_copy(v(1):v(2),p(2))=256;

%counters
nnormal=0;
nnormalA=0;
nnot=0;
nTOT=0;

starti=(D+1)+yini;
finali=yend-(D+1);
startj=(D+1)+xini;
finalj=xend-(D+1);

% filter for porosity extimation (conventionale from morphologic
processing)
FILTER1(starti:finali,startj:finalj)=FILTER(starti:finali,startj:final
j);
% filter for further analysis (skeletonized)
FILTER3(starti:finali,startj:finalj)=FILTER2(starti:finali,startj:fina
lj);
% filter for por size estimation
filter_por=FILTER1(starti:finali,startj:finalj);

% visualisation matrix definition
A_RED(starti:finali,startj:finalj)=0;
A_GREEN(starti:finali,startj:finalj)=0;
A_BLUE(starti:finali,startj:finalj)=0;
A_RED1(starti:finali,startj:finalj)=0;
A_GREEN1(starti:finali,startj:finalj)=0;
% operating area reduced to avoid boundary effects on nodes detection
for i= (D+1+yini):(yend-(D+1))       % all the rows from yini to yend
    for j = (D+1+xini):(xend-(D+1))  % all the coloumns from xini to
xend
        if A(i,j)~=0
           B(i,j)=255;
        end
    end
end
% figure (5)
% imshow(B)
% title('inspected area')


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%ANALYSIS
```

```
for i= (D+1+yini):(yend-(D+1))       % all the rows from yini to yend
    for j = (D+1+xini):(xend-(D+1))  % all the coloumns from xini to
xend
        if A (i,j) > 0
             % if inside previous filter operate following lines
             % nodes data collection
             nTOT=nTOT+1;
             [c, tetaD] = nodes_identifierNEW2(A,i,j);



function [c, tetaD] = nodes_identifierNEW2(A,Xinit,Yinit)
%test
%[c, tetaD] = nodes_identifierNEW1(A,Xinit,Yinit,D);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% teta = [0:pi/180:(2*pi-(pi/180))];




D=10;
count=0;
di=D/3:D/20:D/2;
adi=length(di);
ctotal2=zeros(360,adi);
teta = [0:pi/36:(2*pi)];
ctotal3=zeros(length(teta),1);

for di=D/3:D/20:D/2
    count=count+1;
x = Xinit + (di)*cos(teta);
y = Yinit + (di)*sin(teta);
Xr = round(x);
Yr = round(y);
N=length(Xr);
    for i=1:N
        ctotal2(i,count)=A(Xr(i),Yr(i));
        ctotal3(i,1)=sum(ctotal2(i,:));
    end
end

%%%%%%%

c=ctotal3;
L = length(c);
c (L) = [c(1)];


cave=sum(c)/length(c);
d=c-1*cave;
for i=1:length(d)
    if d(i)>=0;
        e(i)=d(i);
    else
        e(i)=0;
```

```
    end
end

   c=e;
   tetaD= (360/(2*pi))*teta;



              % spikes counting
              spikes_counter = 0;
              spikesfoot = [];
              for k = 1:(length(c')-1)
                 if c(k+1)>0 && c(k)==0
                    spikes_counter = spikes_counter + 1;
                    spikesfoot(spikes_counter)= k;
                 end
              end

               % spikes check

              sL= length(spikesfoot);
              if sL>2
                 nnormal = nnormal+1;
                 A_RED(i,j) = 1; % red pixels                    NORMAL
NODE
                 A_RED1(i,j) = 1;
                 nnormalA = nnormalA+1;
              end
              if sL<=2
                 A_BLUE(i,j) = 1; % blue pixels                  NOT A
NODE
                 nnot=nnot+1;
              end
        end %end if A (i,j) > 0
    end % end j
end % end i

savefile1='savefile_pack.mat';
pack(savefile1);

savefile2 = 'testmoment.mat';
save(savefile2, 'nnormal', 'nnormalA',...

'S','nTOT','nnot','xend','xini','yend','yini','D','v','p','FILTER1','F
ILTER3','A_RED','A_RED1','A_GREEN','A_GREEN1','A_BLUE','original_image
_copy','H','fiber_diameter','T','A','FILTER','original_image','filter_
por','representative_fiber','lastfilt','layer_det');
savefile3 = 'test_bis.mat';
save(savefile3,'A_RED','A_RED1','A_GREEN','A_GREEN1','A_BLUE');
clear
load('testmoment.mat');
clear A_RED A_RED1 A_GREEN A_GREEN1 A_BLUE

% disp('objects/background ratio from the starting BW filter')
```

```
bwobjects= length(find(FILTER1));
bwa=((yend-(D+1))-(D+1+yini));
bwb=(xend-(D+1))-(D+1+xini);
R1=(bwobjects)/(bwa*bwb);

% disp('porosity estimation (Void volume/Total volume)')
P=1-(pi/12)*(R1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% por size distribution

filter_por=imcomplement(filter_por);
filter_por_vis(:,:,1)=filter_por;
filter_por_vis(:,:,2)=filter_por;
filter_por_vis(:,:,3)=filter_por;
filter_por_vis=im2double(filter_por_vis);

% figure(6);
% imshow(filter_por);

filter_por = bwlabel(filter_por);
stats_por =
regionprops(filter_por,'Centroid','Orientation','Area','MajorAxisLengt
h','MinorAxisLength');
clear filter_por
por_area = [stats_por.Area];
por_orient = [stats_por.Orientation];
por_maj = [stats_por.MajorAxisLength];
por_min = [stats_por.MinorAxisLength];
por_AR=por_maj./por_min;
por_centroid= [stats_por.Centroid];

PO=zeros(length(por_orient),1);
for i=1:length(por_orient)
    if por_orient(i)<0
        PO(i)=por_orient(i)+180;
    else
        PO(i)=por_orient(i);
    end
end


Lpor = length(por_area);
j=0;
pores_positions=zeros(Lpor/2,2);

for i = 1 : (Lpor)
        pores_positions(i,1) =por_centroid(i+j);
        pores_positions(i,2) =por_centroid(i+j+1);
        j=j+1;
end
pores_positions_round= round(pores_positions);
```

```
for i=1:length(pores_positions_round)
            Xm=round(pores_positions_round(i,1)+(-
por_maj(1,i)/2:por_maj(1,i)/2)*cos(2*pi/360*(PO(i,1)+0)));
            Ym=round(pores_positions_round(i,2)+(-
por_maj(1,i)/2:por_maj(1,i)/2)*-sin(2*pi/360*(PO(i,1)+0)));
            for j=1:length(Xm)
                if Ym(j)>1&&Xm(j)>1
                    filter_por_vis(Ym(j),Xm(j),1)=0;
                    filter_por_vis(Ym(j),Xm(j),2)=0;
                    filter_por_vis(Ym(j),Xm(j),3)=255;
                end
            end
            clear Xm Ym
            Xm=round(pores_positions_round(i,1)+(-
por_min(1,i)/2:por_min(1,i)/2)*cos(2*pi/360*(PO(i,1)+90)));
            Ym=round(pores_positions_round(i,2)+(-
por_min(1,i)/2:por_min(1,i)/2)*-sin(2*pi/360*(PO(i,1)+90)));
            for j=1:length(Xm)
                if Ym(j)>1&&Xm(j)>1
                    filter_por_vis(Ym(j),Xm(j),1)=0;
                    filter_por_vis(Ym(j),Xm(j),2)=0;
                    filter_por_vis(Ym(j),Xm(j),3)=255;
                end
            end
            clear Xm Ym
end

load('test_bis.mat');

% figure(7);
% imshow(filter_por_vis);
% title('pors and their major axis on the detected scaffold layer')




clear FILTER3
savefile = 'filterPor.mat';
save(savefile ,'filter_por_vis');

clear filter_por filter_por_vis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% DISPLAY RESULTS 1

nodemap(:,:,1)=A_RED1;         % red
clear A_RED1
nodemap(:,:,2)=A_GREEN1;       % green
clear A_GREEN1
nodemap(:,:,3)=A_BLUE;        % blue to map the not node areas
clear A_BLUE
% close all
```

```
% temporarly closed all of the images
% figure (6)
% imshow(nodemap);
% title ('nodes map');
savefile1='savefile_pack.mat';
pack(savefile1);

[a,b]=size(A_RED);

nodemap1(:,:,1)=A_RED;        % red
clear A_RED
nodemap1(:,:,2)=A_GREEN;      % green
clear A_GREEN
A_BLUE2=zeros(a,b);
nodemap1(:,:,3)=A_BLUE2;       % blue to map the not node areas
clear A_BLUE2
% figure (7)
% imshow(nodemap1);
% title ('nodes areas');

%%%%%%%%%%%%%%%%%
% save porosity data
% porosity
porosity=P;
%%%%%%%%%%%%%%%%%%

savefile = 'test.mat';
save(savefile, 'nnormal', 'nnormalA',...

'S','nTOT','nnot','xend','xini','yend','yini','D','v','p','FILTER1','n
odemap1','original_image_copy','H','fiber_diameter','T','A','FILTER','
original_image','nodemap','porosity','por_area','representative_fiber'
,'por_orient','por_AR','R1','lastfilt','layer_det');
clear
load('test.mat');


I = rgb2gray(nodemap1);
clear nodemap1
% figure (8)
% imshow(I);
T1=1/256;
I1 = im2bw(I,T1);
% figure (9)
% imshow(I1);
%


I2 = bwmorph(I1,'clean');
% figure (10)
% imshow(I2);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
```

```
%NODES POSITION IDENTIFICATION AND DISPLAY RESULTS 2
savefile = 'test.mat';
save(savefile, 'nnormal', 'nnormalA',...

'S','nTOT','nnot','xend','xini','yend','yini','D','v','p','FILTER1','I
2','original_image_copy','H','fiber_diameter','T','A','FILTER','origin
al_image','nodemap','porosity','por_area','representative_fiber','por_
orient','por_AR','R1','lastfilt','layer_det');
clear
load('test.mat');


A =I2;
F = bwlabel(A);
stats = regionprops(F,'Centroid','Area');
D= [stats.Centroid];
D1= [stats.Area];
Ld = length(D);
Area_averadge= sum(D1)/(Ld/2);
% eliminate the region smaller than the 80% of the averadge
toosmall= round(0.8*Area_averadge);

Anew = bwareaopen(A,toosmall);
[asel bsel]=size(Anew);
Fnew = bwlabel(Anew);
stats = regionprops(Fnew,'Centroid','Area');
D= [stats.Centroid];
Ld = length(D);

j=0;
nodes_positions=zeros(Ld/2,2);
if Ld>=1
    disp('ok node detected')
  else
    disp('no nodes detected the analysis area is too small')
end


    for i = 1 : (Ld/2)
        nodes_positions(i,1) = D(i+j);
        nodes_positions(i,2) =D(i+j+1);
        j=j+1;
    end
    nodes_positions_round= round(nodes_positions);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% procedure to eliminate nodes that are too close because of the
% skelotonization procedure. minimum distance is given by the
```

```
% representative fiber. remove the lines from 118 to 156 to consider
teh
% totality of the nodes

 Asel=zeros(asel,bsel);
 for i=1:Ld/2
     Asel(nodes_positions_round(i,2),nodes_positions_round(i,1))=1;
 end
%  figure(11);
%  imshow(Asel);
% originally 0.8
se_sel = strel('disk',round(0.4*representative_fiber));
Asel_dil = imdilate(Asel,se_sel);
% figure(12)
% imshow(Asel_dil)

clear D stats
Asel_dil = bwlabel(Asel_dil);
stats = regionprops(Asel_dil,'Centroid');
D= [stats.Centroid];
Ld = length(D);

clear nodes_positions nodes_positions_round
j=0;
nodes_positions=zeros(Ld/2,2);

for i = 1 : (Ld/2)
        nodes_positions(i,1) = D(i+j);
        nodes_positions(i,2) =D(i+j+1);
        j=j+1;
end
nodes_positions_round= round(nodes_positions);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%



    [a,b]=size(F);
    clear C
    C=zeros(a,b);
    for i =1:Ld/2
        Xinit=nodes_positions_round(i,1);
        Yinit=nodes_positions_round(i,2);
        C((Yinit-(1)):(Yinit+(1)),Xinit)=255;
        C(Yinit,(Xinit-(1)):(Xinit+(1)))=255;
    end
    clear nodemap Asel Fnew Anew I2
%     F = im2double(A);
%     nodemap1(:,:,1)=C;            % red
    clear C A
```

```
%     nodemap1(:,:,2)=0;            % green
%     nodemap1(:,:,3)=F;            % blue
%     clear F

%     figure(15)
%     imshow(nodemap1);
%     title ('nodes postions and areas');

%     %visualisation rgb matrix
    [ah,bh]=size(H);

    nodemap2(:,:,1)=H;            % red
    nodemap2(:,:,2)=H;             % green

    nodemap3(:,:,1)=original_image_copy;            % red
    nodemap3(:,:,2)=original_image_copy;            % green

    for i =1:Ld/2
        Xinit=nodes_positions_round(i,1);
        Yinit=nodes_positions_round(i,2);
        % 255 in classic visualisation on BW 3 matrix required for rgb
        nodemap2((Yinit-
(fiber_diameter/2)):(Yinit+(fiber_diameter/2)),Xinit,1)=0;
        nodemap2(Yinit,(Xinit-
(fiber_diameter/2)):(Xinit+(fiber_diameter/2)),1)=0;
        nodemap2((Yinit-
(fiber_diameter/2)):(Yinit+(fiber_diameter/2)),Xinit,2)=0;
        nodemap2(Yinit,(Xinit-
(fiber_diameter/2)):(Xinit+(fiber_diameter/2)),2)=0;

        nodemap3((Yinit-
(fiber_diameter/2)):(Yinit+(fiber_diameter/2)),Xinit,1)=0;
        nodemap3(Yinit,(Xinit-
(fiber_diameter/2)):(Xinit+(fiber_diameter/2)),1)=0;
        nodemap3((Yinit-
(fiber_diameter/2)):(Yinit+(fiber_diameter/2)),Xinit,2)=0;
        nodemap3(Yinit,(Xinit-
(fiber_diameter/2)):(Xinit+(fiber_diameter/2)),2)=0;


        H((Yinit-
(fiber_diameter/2)):(Yinit+(fiber_diameter/2)),Xinit)=255;
        H(Yinit,(Xinit-
(fiber_diameter/2)):(Xinit+(fiber_diameter/2)))=255;

        original_image_copy((Yinit-
(fiber_diameter/2)):(Yinit+(fiber_diameter/2)),Xinit)=255;
        original_image_copy(Yinit,(Xinit-
(fiber_diameter/2)):(Xinit+(fiber_diameter/2)))=255;
    end



    nodemap2(:,:,3)=H;
```

```
    clear H
    nodemap3(:,:,3)=original_image_copy;
    clear original_image_copy


    % H and original_image_copy stay the same as before
    figure(16)
    imshow(nodemap2);
    title ('nodes postions');
%     disp('total number of nodes detected')
    N=Ld/2;

%     figure(17)
%     imshow(nodemap3);
%     title ('nodes postions on original image');



savefile = 'test.mat';
save(savefile, 'nnormal', 'nnormalA',...

'S','nTOT','nnot','xend','xini','yend','yini','v','p','FILTER1','nodes
_positions_round','fiber_diameter','T','FILTER','original_image','node
map2','nodemap3','porosity','por_area','por_orient','por_AR','R1','las
tfilt','Asel_dil','layer_det');
clear
savefile = 'test.mat';
load('test.mat')
load('filterPor.mat')
save(savefile, 'nnormal', 'nnormalA',...

'S','nTOT','nnot','xend','xini','yend','yini','v','p','FILTER1','nodes
_positions_round','fiber_diameter','T','FILTER','original_image','node
map2','nodemap3','porosity','por_area','por_orient','por_AR','filter_p
or_vis','R1','lastfilt','Asel_dil','layer_det');
clear
load('test.mat');




clc
close all
clear
warning off all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% load data from analysis
% load ('1.5_3test.mat')
% load ('9.0_full_test.mat')
% load ('test4-5-atRVE.mat')
% load ('full9_test.mat')
% load ('small_test.mat')
% load ('doublestream_test.mat')
```

```
% load('valid1_test')
load ('test.mat')
clear  nodemap3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% scale input
magn=input('please specify SEM image magnification X  ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% delaunay network creation

% DI_index=input('please provide a Delanauy-Real Network index, it has
to be a number between 0 (Delanauy)and 1(Real)');
DI_index=0.8;

[add_nodes_positions_round] =
network_generator_add_nodesB(layer_det,fiber_diameter);



function [add_nodes_positions_round] =
network_generator_add_nodesB(layer_det,fiber_diameter)

% clc
% close all
% clear
% warning off all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
[original_image,thre_image,layer_det,FILTER,FILTER2,T,fiber_diameter,D
] = layer_detector_fast_NEW11;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% scale input
% magn=input('please specify SEM image magnification X  ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% load('starting_data.mat')
% close all
T=1/256;
filtadd=im2bw(layer_det,T);
[fadda faddb]=size(filtadd);
grid=zeros(fadda,faddb);
% best parameter 1*fiber_diameter
% it means using a grid with a spatial length of 1 fiber diameter
steadd=round(1.5*fiber_diameter);
for i=1:steadd:fadda
    grid(i,1:faddb)=1;
end
for j=1:steadd:faddb
    grid(1:fadda,j)=1;
end
addnod=filtadd&grid;

% figure(1)
```

```
% imshow(filtadd)
% figure(2)
% imshow(grid)
% figure(3)
% imshow(addnod)

% figure(4)
seadd = strel('disk',5);
addnod = imdilate(addnod,seadd);

% imshow(addnod)
ADDNOD(:,:,1)=im2double(addnod);
ADDNOD(:,:,2)=im2double(addnod);
ADDNOD(:,:,3)=im2double(addnod);
addnodlabel = bwlabel(addnod);
stats = regionprops(addnodlabel,'Centroid');
additional_nodes= [stats.Centroid];
j=0;
    for i = 1 : (length(additional_nodes)/2)
        add_nodes_positions(i,1) = additional_nodes(i+j);
        add_nodes_positions(i,2) =additional_nodes(i+j+1);
        j=j+1;
    end
    add_nodes_positions_round= round(add_nodes_positions);

    % Y rows add_nodes_positions_round(i,2)
    % X columns add_nodes_positions_round(i,1)

for i=1:length(add_nodes_positions_round)

ADDNOD(add_nodes_positions_round(i,2),add_nodes_positions_round(i,1),1
)=0;

ADDNOD(add_nodes_positions_round(i,2),add_nodes_positions_round(i,1),2
)=0;

ADDNOD(add_nodes_positions_round(i,2),add_nodes_positions_round(i,1),3
)=255;
end

% figure(5)
% imshow(ADDNOD);



nodes_positions_round_nuo=nodes_positions_round;
stopat=length(nodes_positions_round_nuo);
clear nodes_positions_round

[ao bo]=size(original_image);
redu1=zeros(ao,bo);
redu2=zeros(ao,bo);
for i=1:length(nodes_positions_round_nuo(:,1))
```

```
redu1(nodes_positions_round_nuo(i,2),nodes_positions_round_nuo(i,1))=1
;
end

for j=1:length(add_nodes_positions_round(:,1))

redu2(add_nodes_positions_round(j,2),add_nodes_positions_round(j,1))=1
;
end
clear add_nodes_positions_round   add_nodes_positions stats
seredu = strel('disk',(1)*fiber_diameter);
lastredu = imdilate(redu1,seredu);
clear redu1
% figure(41)
% imshow(lastredu);
% lastredu2 = imdilate(redu2,seredu);
% figure(42)
% imshow(lastredu2);
redu3=imsubtract(lastredu,redu2);

% figure(43)
% imshow(redu3);

% redu4=imabsdiff(lastredu,redu3);
redu4=~redu3;
clear redu3
% figure(44)
% imshow(redu4);
redu5=~lastredu;
clear lastredu
% figure(45)
% imshow(redu5);

redu6=imsubtract(redu4,redu5);
clear redu4 redu5
% figure(45)
% imshow(redu6);

redu7=imabsdiff(redu2,redu6);
clear redu2 redu6
% figure(46)
% imshow(redu7);


lastredulabel = bwlabel(redu7);
clear redu7
stats = regionprops(lastredulabel,'Centroid');
clear lastredulabel
lastredulabel_nodes= [stats.Centroid];
j=0;
    for i = 1 : (length(lastredulabel_nodes)/2)
        add_nodes_positions(i,1) = lastredulabel_nodes(i+j);
        add_nodes_positions(i,2) =lastredulabel_nodes(i+j+1);
        j=j+1;
```

```
    end
    add_nodes_positions_round= round(add_nodes_positions);



nodes_positions_round(:,1)=[nodes_positions_round_nuo(:,1)'
add_nodes_positions_round(:,1)'];
nodes_positions_round(:,2)=[nodes_positions_round_nuo(:,2)'
add_nodes_positions_round(:,2)'];
clear nodes_positions_round_nuo add_nodes_positions_round

%%%%%%%%%%%%%%%%%%%

x=nodes_positions_round(:,1);
y=nodes_positions_round(:,2);
TRI = delaunay(x,y);
Lfriends=length(x);
friends=zeros(Lfriends,Lfriends);

Ltri=length(TRI);
V1=zeros(Ltri,1);
V2=zeros(Ltri,1);
V3=zeros(Ltri,1);
for k=1:Ltri
    V1(k)=TRI(k,1);
    V2(k)=TRI(k,2);
    V3(k)=TRI(k,3);
end
clear TRI x y

for k=1:Ltri
    friends(V1(k),V2(k))=1;
    friends(V2(k),V1(k))=1;
    friends(V1(k),V3(k))=1;
    friends(V3(k),V1(k))=1;
    friends(V2(k),V3(k))=1;
    friends(V3(k),V2(k))=1;
end
clear V1 V2 V3 redu1 redu2 redu3 redu4 redu5 redu6 redu7 stats FILTER1
Asel_dil layer_det add_nodes_positions lastredulabel_nodes
save('porefil','filter_por_vis')
clear lastredu lastredulabel filter_por_vis seredu
% clear FILTER
save('Fr','friends');
clear friends
cwd = pwd;
cd(tempdir);
pack
cd(cwd)



c=length(nodes_positions_round);
friends_angle=zeros(c,c);
original_imagex=original_image;
% clear original_image;
```

```
original_imagex(v(1),p(1):p(2))=255;
original_imagex(v(2),p(1):p(2))=255;
original_imagex(v(1):v(2),p(1))=255;
original_imagex(v(1):v(2),p(2))=255;

%for rgb visualisation
original_image2(:,:,1)=original_imagex;
original_image2(:,:,2)=original_imagex;
original_image2(:,:,3)=original_imagex;
clear original_imagex

% data to visualize nodes on the image
teta = 0:pi/180:(2*pi-(pi/180));
diam=(1/2)*fiber_diameter;

Xinit=zeros(c,1);
Yinit=zeros(c,1);
nodemap2(:,:,1:3)=nodemap2;
[a b]=size(nodemap2(:,:,1));
refinement=zeros(a,b);
% START THE NETWORK GENERATION
    for i =1:c

        Xinit(i)=nodes_positions_round(i,1);    % increasing order
from the left to the right side of the image
        Yinit(i)=nodes_positions_round(i,2);    % increasing order
from the top to the bottom of the image

         if i<=stopat
            original_image2((Yinit(i)-
(diam)):(Yinit(i)+(diam)),Xinit(i),1)=0;
            original_image2(Yinit(i),(Xinit(i)-
(diam)):(Xinit(i)+(diam)),1)=0;
            original_image2((Yinit(i)-
(diam)):(Yinit(i)+(diam)),Xinit(i),2)=0;
            original_image2(Yinit(i),(Xinit(i)-
(diam)):(Xinit(i)+(diam)),2)=0;
            original_image2((Yinit(i)-
(diam)):(Yinit(i)+(diam)),Xinit(i),3)=255;
            original_image2(Yinit(i),(Xinit(i)-
(diam)):(Xinit(i)+(diam)),3)=255;
         end


        xcir = Xinit(i) + (diam)*cos(teta);
        ycir = Yinit(i) + (diam)*sin (teta);
        Xr = round(xcir);
        Yr = round(ycir);
        % to control the refinement circles radius crosslinking
        xcir1 = Xinit(i) + (1/2*diam)*cos(teta);
        ycir1 = Yinit(i) + (1/2*diam)*sin (teta);
        Xr1 = round(xcir1);
        Yr1 = round(ycir1);
        % to control the refinement circles radius connecting points
        xcir3 = Xinit(i) + (1/2*diam)*cos(teta);
```

```
        ycir3 = Yinit(i) + (1/2*diam)*sin (teta);
        Xr3 = round(xcir3);
        Yr3 = round(ycir3);



        % to distinguish crosslinking from connecting points
        xcir2 = Xinit(i) + (1/4*diam)*cos(teta);
        ycir2 = Yinit(i) + (1/4*diam)*sin (teta);
        Xr2 = round(xcir2);
        Yr2 = round(ycir2);

        N=length(Xr);
        for j=1:N
           if i<=stopat
               original_image2(Yr(j),Xr(j),1)=0;
               original_image2(Yr(j),Xr(j),2)=0;
               original_image2(Yr(j),Xr(j),3)=255;

               nodemap2(Yr(j),Xr(j),1)=0;
               nodemap2(Yr(j),Xr(j),2)=0;
               nodemap2(Yr(j),Xr(j),3)=255;


           if (Yr1(j)>0&&Xr1(j)>0)
               refinement(Yr1(j),Xr1(j))=1;
           end

           elseif i>stopat
               original_image2(Yr2(j),Xr2(j),1)=0;
               original_image2(Yr2(j),Xr2(j),2)=0;
               original_image2(Yr2(j),Xr2(j),3)=255;

               nodemap2(Yr2(j),Xr2(j),1)=0;
               nodemap2(Yr2(j),Xr2(j),2)=0;
               nodemap2(Yr2(j),Xr2(j),3)=255;

           if (Yr3(j)>0&&Xr3(j)>0)
               refinement(Yr3(j),Xr3(j))=1;
           end

           end


        end % for

    end % for


% refinement matrix to avoid the multiconnections effect

refinement = imfill(refinement,'holes');
```

```
% imshow(refinement);
% figure(31)

DIST=zeros(c,c);
    for i=1:c
        for j=1:c
            box = ((nodes_positions_round(i,2)-
nodes_positions_round(j,2))^2)+((nodes_positions_round(i,1)-
nodes_positions_round(j,1))^2);
            DIST(i,j)=sqrt(box);
        end
    end
COUNTER=zeros(c,c);
counter=zeros(c,c);
counref=zeros(c,c);
POSX=zeros(c,c);
POSY=zeros(c,c);
friends_NEW=zeros(c,c);
Xcenter=zeros(c,1);
Ycenter=zeros(c,1);
Xcenter_friend=zeros(c,1);
Ycenter_friend=zeros(c,1);
load('Fr')
  for i=1:c;
    Xcenter(i)=nodes_positions_round(i,1);
    Ycenter(i)=nodes_positions_round(i,2);
        for j = 1:c
            counter(i,j)=0;
            if friends(i,j)==1
                %&i<j
               Xcenter_friend(j)=nodes_positions_round(j,1);
               Ycenter_friend(j)=nodes_positions_round(j,2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
               % middle point position for diameter check
               if Xcenter_friend(j)>Xcenter(i)
                  POSX(i,j)=round(Xcenter_friend(j)-
(Xcenter_friend(j)-Xcenter(i))/2);
               elseif Xcenter_friend(j)<Xcenter(i)
                  POSX(i,j)=round(Xcenter(i)+(Xcenter_friend(j)-
Xcenter(i))/2);
               elseif Xcenter_friend(j)==Xcenter(i)
                  POSX(i,j)=round(Xcenter_friend(j));
               end

               if Ycenter_friend(j)>Ycenter(i)
                  POSY(i,j)=round(Ycenter_friend(j)-
(Ycenter_friend(j)-Ycenter(i))/2);
               elseif Ycenter_friend(j)<Ycenter(i)
                  POSY(i,j)=round(Ycenter(i)+(Ycenter_friend(j)-
Ycenter(i))/2);
               elseif Ycenter_friend(j)==Ycenter(i)
                  POSY(i,j)=round(Ycenter_friend(j));
               end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                friends_angle(i,j)=atan((Ycenter_friend(j)-
Ycenter(i))/(Xcenter_friend(j)-Xcenter(i)));
                %  friends_angle(i,j) in radiant multiply by
(360/(2*pi))*
                %  to have it n degree
                if friends_angle(i,j)<0
                    friends_angle(i,j)=friends_angle(i,j)+2*pi;
                end

                if Xcenter(i)<Xcenter_friend(j)
                  xf= Xcenter(i):0.01:Xcenter_friend(j);
                elseif Xcenter(i)>Xcenter_friend(j)
                  xf= Xcenter_friend(j):0.01:Xcenter(i);
                elseif Xcenter(i)==Xcenter_friend(j)
                       xf= Xcenter(i);
                end

                if length(xf)>1
                    yf= (xf-
Xcenter(i))*(tan(friends_angle(i,j)))+Ycenter(i);
                elseif length(xf)<=1
                       if Ycenter(i)<=Ycenter_friend(j)
                           yf=Ycenter(i):0.01:Ycenter_friend(j);
                       elseif Ycenter(i)>Ycenter_friend(j)
                           yf=Ycenter(i):0.01:Ycenter_friend(j);
                       end
                    if length(yf)>1
                    xf=Xcenter(i)*ones(1,length(yf));
                    elseif length(yf)<=1
                    yf=Ycenter(i);
                    xf=Xcenter(i)*ones(1,length(yf));
                    end
                end
                Xr1 = round(xf);
                Yr1 = round(yf);
                N(i,j)=max(length(Xr1),length(Yr1));
                for k=1:N(i,j)
                     if lastfilt(Yr1(k),Xr1(k))==1
                        counter(i,j)=counter(i,j)+1;
                     end
                     if k>1
                         if
refinement(Yr1(k),Xr1(k))==1&&refinement(Yr1(k-1),Xr1(k-1))==0
                             % it controls how many time the segment
enters from
                             % the refinement filter
                             counref(i,j)=counref(i,j)+1;
                         end
                     end
                end
                if N(i,j)>1
                   COUNTER(i,j)=counter(i,j)/N(i,j);
                elseif N(i,j)==1
```

```
                    COUNTER(i,j)=0;
                end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                %0 for Delanauy
                %1 for 100% match with the fiber network


                for k=1:N(i,j)
                      if (COUNTER(i,j))>=(DI_index)&&counref(i,j)<=1

                            original_image2(Yr1(k),Xr1(k),1)=0;
                            original_image2(Yr1(k),Xr1(k),2)=0;
                            original_image2(Yr1(k),Xr1(k),3)=255;

                            nodemap2(Yr1(k),Xr1(k),1)=0;
                            nodemap2(Yr1(k),Xr1(k),2)=0;
                            nodemap2(Yr1(k),Xr1(k),3)=255;
                           friends_NEW(i,j)=1;
                      elseif (COUNTER(i,j))<(DI_index)||counref(i,j)>1
                           friends_NEW(i,j)=0;
                      end %end filter check if
                end %end k
            end % end friends if
        end % end j
  end % end i


clear friends
friends=friends_NEW;
clear friends_NEW

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% Fiber diameter estimation



disp ('please select the biggest fiber diameter, the smallest will be
1/10 ')
disp ('left click for initial point, right click for ending point')
% [c1,r1,P1] =
impixel(original_image2(1:round(20*magn/100),1:round(20*magn/100)));
[aio bio]=size(original_image2);
[c1,r1,P1] =
impixel(original_image2(1:20*fiber_diameter,1:20*fiber_diameter));
% [c1,r1,P1] = impixel(original_image2);
title('max fiber diameter selection')
box = ((c1(2)-c1(1))^2)+((r1(2)-r1(1))^2);
D=round(sqrt(box));
disp(D)
D = input('please confirm or change the max fiber_diameter value, type
an integer   ');
Lseg=-D/2:D/2;
```

```
% originally the use of FILTER was necessary, now diameter_identifier7
% solves everything
% filtered_diam_layer = immultiply(original_image,FILTER);

filtered_diam(:,:,1)=original_image;
filtered_diam(:,:,2)=original_image;
filtered_diam(:,:,3)=original_image;

[afilt,bfilt]=size(lastfilt);
FILTERdiam=zeros(afilt,bfilt);
actionflag=zeros(1,length(Lseg));
diamcounter=0;
diamcounter1=0;
diamcounter2=0;

for i=1:c
    for j=1:c;
        if friends(i,j)==1&&i>j

         % intermediate point for diameter
         original_image2(POSY(i,j),POSX(i,j),1)=0;
         original_image2(POSY(i,j),POSX(i,j),2)=255;
         original_image2(POSY(i,j),POSX(i,j),3)=0;

         nodemap2(POSY(i,j),POSX(i,j),1)=0;
         nodemap2(POSY(i,j),POSX(i,j),2)=255;
         nodemap2(POSY(i,j),POSX(i,j),3)=0;

         % identifying segment

         % tetaseg=friends_angle(i,j);

         % central segment

xsegm(1:length(Lseg))=round(POSX(i,j)+(Lseg)*sin(friends_angle(i,j)));
         ysegm(1:length(Lseg))=round(POSY(i,j)-
(Lseg)*cos(friends_angle(i,j)));

         % segment +5
         s1x=round(POSX(i,j)+5*sin(friends_angle(i,j)+pi/2));
         s1y=round(POSY(i,j)-5*cos(friends_angle(i,j)+pi/2));


xsegm1(1:length(Lseg))=round(s1x+(Lseg)*sin(friends_angle(i,j)));
         ysegm1(1:length(Lseg))=round(s1y-
(Lseg)*cos(friends_angle(i,j)));
         % segment -5

         s2x=round(POSX(i,j)-5*sin(friends_angle(i,j)+pi/2));
         s2y=round(POSY(i,j)+5*cos(friends_angle(i,j)+pi/2));


xsegm2(1:length(Lseg))=round(s2x+(Lseg)*sin(friends_angle(i,j)));
```

```
         ysegm2(1:length(Lseg))=round(s2y-
(Lseg)*cos(friends_angle(i,j)));




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
         %%%% diameter_identifier
         [DIAM,YsegTOT,XsegTOT] =
diameter_identifier7(xsegm,ysegm,original_image,POSX(i,j),POSY(i,j),D)
;


function [DIAM,YsegTOT,XsegTOT] =
diameter_identifier7(Xseg,Yseg,filtered_diam_layer,XO,YO,maxdiam)
% [DIAM,YsegTOT,XsegTOT] =
diameter_identifier7(Xseg,Yseg,filtered_diam_layer,XO,YO,maxdiam)




% values collected at:
diam_raw_data=zeros(1,length(Xseg));
[limita limitb]=size(filtered_diam_layer);


% for loop to avoid collection of points outside from the original
image
    for i=1:length(Xseg)
        if
(Yseg(i)<limita)&&(Xseg(i)<limitb)&&(Yseg(i)>0)&&(Xseg(i)>0)
           diam_raw_data(i)=filtered_diam_layer(Yseg(i),Xseg(i));
        else
           diam_raw_data(i)=0;
        end
    end




% FIVE POINT COLLECTION
% 1 max of the values
[abig, bbig]=max(diam_raw_data);

FXdiam_raw_data = (gradient(diam_raw_data)); % gradient
% 2 getting in the fiber
[amax, bmax]=max(FXdiam_raw_data);

% 3 going out from the fiber
[amin, bmin]=min(FXdiam_raw_data);

% 4 internal point getting in
```

```
% sum for internal point, max and min of the gradient must be
eliminated
% first
FXdiam_raw_dataRed=FXdiam_raw_data;
FXdiam_raw_dataRed(bmin)=0;
FXdiam_raw_dataRed(bmax)=0;
SUMdiam_raw_data=1*FXdiam_raw_dataRed/max(FXdiam_raw_dataRed)+diam_raw
_data/max(diam_raw_data);
[aint bint]=max(SUMdiam_raw_data);
% 5 internal point getting out
SUBdiam_raw_data=1*(FXdiam_raw_dataRed/min(FXdiam_raw_dataRed))+diam_r
aw_data/max(diam_raw_data);
[aint1 bint1]=max(SUBdiam_raw_data);


% Condition to performe the analysis
% in - initial point1 - max - internal point2 - out
    if bmax>0&&bint>0&&bbig>0&&bint1>0&&bmin
        if (bmax<bint)&&(bint<=bbig)&&(bbig<=bint1)&&(bint1<bmin)
            dist1=abs(bmax-bint);
            dist2=abs(bint-bbig);
            dist3=abs(bbig-bint1);
            dist4=abs(bint1-bmin);
            dist_total=[dist1,dist2,dist3,dist4];
            [DIAM,posDIM]=max(dist_total);
        else
            DIAM=0;
            XsegTOT=0;
            YsegTOT=0;
        end
    else
          DIAM=0;
          XsegTOT=0;
          YsegTOT=0;
    end
if DIAM>0
            if posDIM==1
                D1=bmax;
                D2=bint;
            elseif posDIM==2
                D1=bint;
                D2=bbig;
            elseif posDIM==3
                D1=bint1;
                D2=bbig;
            elseif posDIM==4
                D1=bint1;
                D2=bmin;
            end
            % diameter value
            DIAM=(sqrt(((Yseg(D1)-Yseg(D2))^2)+((Xseg(D1)-
Xseg(D2))^2)));
    if Xseg(D1)<=Xseg(D2)
            Xstart=Xseg(D1);
            Ystart=Yseg(D1);
            Xend=Xseg(D2);
```

```
            Yend=Yseg(D2);
    elseif Xseg(D1)>Xseg(D2)
            Xstart=Xseg(D2);
            Ystart=Yseg(D2);
            Xend=Xseg(D1);
            Yend=Yseg(D1);
    end
        LDiamseg=sqrt(((Xseg(D2)-Xseg(D1))^2)+((Yseg(D2)-
Yseg(D1))^2));
        Diam_angle=atan((Yend-Ystart)/(Xend-Xstart));
        XsegTOT=round(Xstart+(0:0.01:1)*LDiamseg*cos(Diam_angle));
        YsegTOT=round(Ystart+(0:0.01:1)*LDiamseg*sin(Diam_angle));

        dis1=(sqrt(((Yseg(D1)-YO)^2)+((Xseg(D1)-XO)^2)));
        dis2=(sqrt(((Yseg(D2)-YO)^2)+((Xseg(D2)-XO)^2)));
end



% correction intersection
checkinters=0;
for i=1:length(XsegTOT)
        if  XsegTOT(i)==XO&&YsegTOT(i)==YO
            checkinters=1;
        end
end
% correction intersection + angle correction + error too small 1/10 of
the
% maximum
% (dis1>1.5*dis2)||(dis2>1.5*dis1)||
if
checkinters<1||(dis1>1.3*dis2)||(dis1<0.7*dis2)||(dis2>1.3*dis1)||(dis
2<0.7*dis1)||DIAM<0.2*maxdiam
    XsegTOT=0;
    YsegTOT=0;
    DIAM=0;
end




            if DIAM > 0
               diamcounter=diamcounter+1;
               Diam2(diamcounter)= DIAM;
                  for kse=1:length(YsegTOT)
                    original_image2(YsegTOT(kse),XsegTOT(kse),1)=255;
                    original_image2(YsegTOT(kse),XsegTOT(kse),2)=0;
                    original_image2(YsegTOT(kse),XsegTOT(kse),3)=0;
                  end
            end

            [DIAM1,YsegTOT1,XsegTOT1] =
diameter_identifier7(xsegm1,ysegm1,original_image,s1x,s1y,D);
            if DIAM1 > 0
```

```
                diamcounter1=diamcounter1+1;
                Diam3(diamcounter1)= DIAM1;
                   for kse=1:length(YsegTOT1)

original_image2(YsegTOT1(kse),XsegTOT1(kse),1)=255;
                     original_image2(YsegTOT1(kse),XsegTOT1(kse),2)=0;
                     original_image2(YsegTOT1(kse),XsegTOT1(kse),3)=0;
                   end
             end


             [DIAM2,YsegTOT2,XsegTOT2] =
diameter_identifier7(xsegm2,ysegm2,original_image,s2x,s2y,D);
             if DIAM2 > 0
                diamcounter2=diamcounter2+1;
                Diam4(diamcounter2)= DIAM2;
                   for kse=1:length(YsegTOT2)

original_image2(YsegTOT2(kse),XsegTOT2(kse),1)=255;
                     original_image2(YsegTOT2(kse),XsegTOT2(kse),2)=0;
                     original_image2(YsegTOT2(kse),XsegTOT2(kse),3)=0;
                   end
             end




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
         clear xsegm ysegm DIAM Yseg Xseg xsegm1 ysegm1 DIAM1 Yseg1
Xseg1 xsegm2 ysegm2 DIAM2 Yseg2 Xseg2



        end % end if friend
    end %end for
end % end for

% FILTERdiam_F = bwlabel(FILTERdiam);
% stats = regionprops(FILTERdiam_F,'MajorAxisLength','Area');
% Diam1= [stats.MajorAxisLength];
% Diam2= [stats.Area];
if length(Diam2)<=1
    Diam2=0;
elseif length(Diam3)<=1
    Diam3=0;
elseif length(Diam4)<=1
    Diam4=0;
end
Diam2=[Diam2 Diam3 Diam4];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
```

```
friends_angle=(360/(2*pi))*friends_angle;
[a,b]=size(friends_angle);
Fangle=zeros(a,b);
C=zeros(a,b);
good_friends=zeros(a,b);
LogicA=friends_angle>180;


for i = 1:a
    for j=1:b
        if i<j&&friends(i,j)==1
            Fangle(i,j)=friends_angle(i,j);
            C(i,j)=1;
            if LogicA(i,j)==1
                Fangle(i,j)=friends_angle(i,j)-360;

            end
            good_friends(i,j)=friends(i,j);
        end
        if i>j
            good_friends(i,j)=friends(j,i);
        end
    end
end

%OUTPUTS FILTERED
% friendship matrix correct = good_friends
% angle matrix correct = Fangle

% disp('number of nodes')
Nnodes=length(good_friends);
% disp('number of connections')
Connections=sum(sum(good_friends))/2;

connectivity=zeros(Nnodes,1);
for f=1:Nnodes
    connectivity(f)=sum(good_friends(f,1:Nnodes));
 end
% connectivity is a vector that represents the number of connection of
each
% node, in this way we count the connection between i and j twice but
we
% make a correct count of single node connections. If normalization is
% required is necessary to divide for the total number of element in
the
% matrix good_friends. However the actual number of connecion will be
half
% the value of the non-zero element in good-friends


% connectivity filtered
% min_con=min(connectivity);
```

```
min_con=3;
max_con=max(connectivity);
conn_interval=(min_con:max_con);
Conn_count=zeros(length(min_con:max_con),1);
k=1;
for conn=min_con:max_con
    [conn_vect]=find(connectivity==conn);
    Conn_count(k)=length(conn_vect);
    k=k+1;
end

%normalized connectivity count
Conn_count_norm=Conn_count/(sum(Conn_count));

clear C COUNTER FILTERdiam FILTERdiam_F counter POSX POSY lastfilt
FILTER1 logicA N



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
% Results Visualization









%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% FEM model data production UNCOMMENT IF REQUIRED


% nodes data no longer required uncomment if necessary

% nodes_identifier=1:Nnodes;
% nodes_identifier=nodes_identifier';
% nodes(:,1)=nodes_identifier;
% nodes(:,2)=nodes_positions_round(:,1);
% nodes(:,3)=nodes_positions_round(:,2);
```

```
for i = 1:a
    for j=1:b
        if i<j
            good_friends(i,j)=good_friends(i,j);
        end
        if i>j
            good_friends(i,j)=0;
        end
    end
end

% elements data no longer required uncomment if necessary

% counterE=0;
% for i = 1:a
%     [E1]=find(good_friends(i,:));
%     LE1=length(E1);
%        for k=1:LE1
%            E=[i,E1(k)];
%            counterE=counterE+1;
%            Etot(counterE,1)=i;
%            Etot(counterE,2)=E1(k);
%        end
% end

% fiber stiffness data no longer required uncomment if necessary

% counterK=0;
% for i=1:a
%     for j=1:b
%         if good_friends(i,j)==1
%             counterK=counterK+1;
%             Ktot(counterK,1)=COUNTER(i,j);
%         end
%     end
% end



% end FEM data generation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% Area data
Area=(p(2)-p(1))*(v(2)-v(1));
% p and v define the inner square in "nodesmap2" this represents the
% actual analized area  in pixels
```

```
% magn=input('please specify SEM image magnification X  ');

s_pixel=1/((magn/100)*(magn/100));

Real_area=Area*s_pixel;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% nodes density
            % old implementation
            % if min_con==0
            %     Non_conn_nodes=Conn_count(1);
            %     disp('some nodes are not connected please select a
smaller D-R index')
            % else
            %     Non_conn_nodes=0;
            % end

            % effective number of nodes
            % Nnodes=Nnodes-Non_conn_nodes;

% with additional helping nodes the nodes density is defined by the
% original value given by the number of crosslinking identified
% Non_conn_nodes is set = 0 by definition

Non_conn_nodes=0;
Nnodes=stopat;
nodes_density=Nnodes/Real_area;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% connectivity
figure(4)
ymax=max(Conn_count_norm)+0.1; % the theoritical value is 100% --> 1
but usually < 50%-->0.5
La=num2str(nodes_density);
La=strcat('Nodes density = ','',La,'',' Nodes/micrometers^2');
bar(conn_interval,Conn_count_norm,'k'),xlabel('number of connection
related to a node'),ylabel('norm number of
nodes'),axis([min(conn_interval)-1 max(conn_interval)+1 0 ymax])
legend(La);
grid on
title ('connectivity after BW filtering')




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

% Fiber Orientation Distribution

[a,b]=size(good_friends);
B=zeros(a,b);
```

```
C=zeros(a,b);

for i = 1:a
    for j=1:b
        if i<j
            if Fangle(i,j)>=0
            B(i,j)=Fangle(i,j);
            elseif Fangle(i,j)<0
            B(i,j)=Fangle(i,j)+180;
            end
            if good_friends(i,j)==1
               C(i,j)=DIST(i,j);
            end
        end
    end
end


k=sum(sum(good_friends));
value=zeros(k,1);
pos_raw=zeros(k,1);
pos_column=zeros(k,1);
k=0;
for i=1:a
    for j=1:b
         if good_friends(i,j)==1
             k=k+1;
             value(k)=B(i,j);
             pos_raw(k)=i;
             pos_column(k)=j;

         end
    end
end


f=length(pos_raw);
weight=zeros(f,1);
for i=1:f
    weight(i)=C(pos_raw(i),pos_column(i));
end

% stepangle=input('please provide the desired binning for angle
distribution  ');
stepangle=10;
g=length(value);
interval=(0:stepangle:180);
k=length(interval);
counter=zeros(k,1);
counter2=zeros(k,1);
for i=1:(g)
    for j=1:(k)
        if j<k
            if (value(i)<interval(j+1))&&(value(i)>=interval(j))
                counter(j)=counter(j)+1;
```

```
                counter2(j)=weight(i)+ counter2(j);
            end
        end
    end
end

% counter_weighted=counter.*counter2;
counter_weighted=counter2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% fiber orientation index from simple fiber angle distribution
[acou bcou]=max(counter2);
disp('maximum in the fiber angle distribution from')
disp(interval(bcou));
disp('to');
disp(interval(bcou+1));
tetadefault=input('please provide the supposed orientation angle  ');
deltateta=(value-tetadefault);
deltatetaradiant = (2*pi/360)*deltateta;
oi=sum((cos(deltatetaradiant).^2));
OI=oi/length(value);

% fiber orientation index from weighted fiber angle distribution

% tetadefault=input('please provide the supposed orientation angle
');
% deltateta=(value-tetadefault);
% deltatetaradiant = (2*pi/360)*deltateta;
oiW=sum(weight.*(cos(deltatetaradiant).^2)/sum(weight));
OIW=oiW;
% OIW=oiW/length(value);




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

% fiber angle distribution
figure(5)
x_hist=(0:stepangle:180);

% using the built in function for histograms
% [n,xout]=hist(value,x_hist);
%  n1=n/length(value);
La=num2str(OI);
La=strcat('Orientation Index = ','',La);
n1=counter./sum(counter);
bar(0:stepangle:180,n1,'k')
axis([0 180 0 max(n1)+0.1])
legend(La);
grid on
xlabel ('angle [degrees]');
ylabel ('normalized number of fibers');
title ('normalized orientation distribution (not weighted)');
```

```
% fiber angle distribution weighted
La=num2str(OIW);
La=strcat('Orientation Index Weighted = ','',La);
counter_weighted_norm=counter_weighted/sum(counter_weighted);
figure(6)
bar(0:stepangle:180,counter_weighted_norm,'k')
axis([0 180 0 max(counter_weighted_norm)+0.1])
legend(La);
grid on
xlabel ('angle [degrees]');
ylabel ('normalized number of fibers weighted');
title ('normalized orientation distribution (length weighted)')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fiber length distribution

[a b]=size(DIST);
countdist=0;
DIST1=zeros(sum(sum(good_friends)),1);
DIST2=zeros(sum(sum(good_friends)),1);
DIST3=zeros(sum(sum(good_friends)),1);
for i=1:a
    for j=1:b
       if good_friends(i,j)==1
        countdist=1+countdist;
        DIST1(countdist)=DIST(i,j)*(100/magn);

DIST2(countdist)=abs(DIST(i,j).*cos(B(i,j)*(pi/180))*(100/magn));

DIST3(countdist)=abs(DIST(i,j).*sin(B(i,j)*(pi/180))*(100/magn));
       end
    end
end
% using a step of 0.1 microns
stepL=0.1;
length_int=(min(DIST1):stepL:max(DIST1));

% number of fibers used for fiber length estimation
n_fiber_used_length=length(DIST1);
[nlength,length_int]=hist(DIST1,length_int);
norm_nlength=nlength/sum(nlength);
% mean
Length_ave=sum(DIST1)/length(DIST1);
La=num2str(Length_ave);
% standard deviation
stand_dev_Length=std(DIST1);
Lst=num2str(stand_dev_Length);
La=strcat('mean length = ','',La,'+/-',Lst,'','micrometers');
% figure(7);
% bar(length_int,norm_nlength,'k');
% legend(La);
% axis([min(DIST1)-stepL max(DIST1) 0 max(norm_nlength)+0.1]);
% grid on;
% xlabel ('node to node length [micrometers]');
```

```
% ylabel ('normalized number of fibers');
% title ('normalized node to node length distribution');



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% fiber diameter plot

% estimation on the basis of the segment area in FILTERdiam


Diam2=2*Diam2;
Diam2=Diam2*100/magn;
% mean
Diam_ave=sum(Diam2)/length(Diam2);
% standard deviation
stand_dev_Diam2=std(Diam2);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% no longer necessary filtering in the diameter selection operated by
the
% function diameter_identifiier7
% considering a normal distribution eliminating the values
% greater and smaller than mean+3*std_deviation means not considering
% the 0.1% only of the results or mean+2*std_deviation 2.1%
% this eliminate the possibility to use fiber bundles for fiber
diameter
% estimation
% Alogic1=(Diam2<(Diam_ave+3*stand_dev_Diam2));
% Alogic2=(Diam2>(Diam_ave-3*stand_dev_Diam2));
%
% [adpos]=find(Alogic1&Alogic2);
% % number of fibers used to estimate the fiber diameter
% number_fiber_dia=length(adpos);
% Diam_used=zeros(number_fiber_dia,1);
% for kd=1:number_fiber_dia
%     Diam_used(kd)=Diam2(adpos(kd));
% end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
clear Diam_used
Diam_used=Diam2;
number_fiber_dia=length(Diam_used);
Diam_ave_used=sum(Diam_used)/length(Diam_used);
stepD=Diam_ave_used/10;
% stepD=0.001;
length_int_diam=(min(Diam_used):stepD:max(Diam_used));
[ndiam,length_int_diam]=hist(Diam_used,length_int_diam);
norm_ndiam=ndiam/sum(ndiam);
% mean
Diam_ave_used=sum(Diam_used)/length(Diam_used);
% standard deviation
```

```
stand_dev_Diam_used=std(Diam_used);
La=num2str(Diam_ave_used);
Lst=num2str(stand_dev_Diam_used);
La=strcat('mean diameter = ','',La,'+/-',Lst,'','micrometers');
figure(8);
bar(length_int_diam,norm_ndiam,'k');
legend(La);
axis([min(Diam_used) max(Diam_used) 0
max(norm_ndiam)+(max(norm_ndiam)-min(norm_ndiam))/3]);
grid on;
xlabel ('fiber diameter [micrometers]');
ylabel ('normalized number of fibers');
title ('normalized fiber diameter distribution');



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% Porosity & por sizes

por_sizes=por_area*(100/magn)*(100/magn);
stepP=0.5;
int_por_sizes=(min(por_sizes):stepP:max(por_sizes));
[psizes,int_por_sizes]=hist(por_sizes,int_por_sizes);
norm_psizes=psizes/sum(psizes);
% mean
por_sizes_ave=sum(por_sizes)/length(por_sizes);
% standard deviation
stand_dev_por_sizes=std(por_sizes);
La=num2str(por_sizes_ave);
Lst=num2str(stand_dev_por_sizes);
La=strcat('mean por size = ','',La,'+/-',Lst,'','micrometers^2');
figure(9);
bar(int_por_sizes,norm_psizes,'k');
legend(La);
axis([min(por_sizes)-stepP max(por_sizes) 0 max(norm_psizes)+0.1]);
grid on;
xlabel ('por size [micrometers^2]');
ylabel ('normalized number of pors');
title ('normalized por size  distribution');

npors_used=length(por_sizes);



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% Por orientation

% loop to express the por angle in the range 0 -180
gp=length(por_orient);
POR_OR=zeros(gp,1);
for i=1:gp
    if por_orient(i)<0
        POR_OR(i)=por_orient(i)+180;
```

```
    else
        POR_OR(i)=por_orient(i);
    end
end

interval=(0:stepangle:180);
kp=length(interval);
counter_por=zeros(k,1);
% loop to create the histogram equivalent to the fiber angle
distribution
% (value)/counter equivalent to (por_orient)/counter_por
for i=1:(gp)
    for j=1:(kp)
        if j<kp
            if (POR_OR(i)<interval(j+1))&&(POR_OR(i)>=interval(j))
                counter_por(j)=counter_por(j)+1;
            end
        end
    end
end

% pore orientation index from simple pore angle distribution

deltateta_por=(POR_OR-tetadefault);
deltatetaradiant_por = (2*pi/360)*deltateta_por;
oi_por=sum((cos(deltatetaradiant_por).^2));
OI_por=oi_por/length(POR_OR);


int_por_angle=(interval);
norm_porient=counter_por/sum(counter_por);
La=num2str(OI_por);
La=strcat('Orientation Index = ','',La);
figure(10);
bar(int_por_angle,norm_porient,'k');
legend(La);
axis([0 180 0 max(norm_porient)+0.1])
grid on;
xlabel ('angle [degrees]');
ylabel ('normalized number of pors');
title ('normalized por angle distribution');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%pors aspect ratio

% all groups of pors
int_por_AR=(min(por_AR):0.1:max(por_AR));
[pAR,int_por_AR]=hist(por_AR,int_por_AR);
norm_pAR=pAR/sum(pAR);
% mean
por_AR_ave=sum(por_AR)/length(por_AR);
% standard deviation
stand_dev_por_AR=std(por_AR);
La=num2str(por_AR_ave);
```

```
Lst=num2str(stand_dev_por_AR);
La=strcat('mean AR = ','',La,'+/-',Lst);
figure(11);
bar(int_por_AR,norm_pAR,'k');
legend(La);
axis([1 max(por_AR) 0 max(norm_pAR)+0.1]);
grid on;
xlabel ('por Aspect Ratio');
ylabel ('normalized number of pors');
title ('normalized por Aspect Ratio  distribution');
npors_used_AR=length(por_AR);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% pors potentially able to host cells
% j=0;
% upper_bound=input('please select upper bound for cells area
micrometer^2  ');
% lower_bound=input('please select lower bound for cells area
micrometer^2  ');
%
% Blogic1=(por_sizes<upper_bound);
% Blogic2=(por_sizes>lower_bound);
% [adPOR]=find(Blogic1&Blogic2);
% selected_pAR=zeros(length(adPOR),1);
% for i=1:length(por_sizes)
%     if (por_sizes(i)<upper_bound)&&(por_sizes(i)>lower_bound)
%         j=j+1;
%         selected_pAR(j)=por_AR(i);
%     end
% end
%
% int_por_selectedAR=(min(selected_pAR):0.5:max(selected_pAR));
% [selpAR,int_por_selectedAR]=hist(selected_pAR,int_por_selectedAR);
% norm_selpAR=selpAR/sum(selpAR);
% % mean
% selec_por_AR_ave=sum(selected_pAR)/length(selected_pAR);
% % standard deviation
% stand_dev_selec_por_AR=std(selected_pAR);
% La=num2str(selec_por_AR_ave);
% Lst=num2str(stand_dev_selec_por_AR);
% La=strcat('mean AR = ','',La,'+/-',Lst);
% figure(12);
% bar(int_por_selectedAR,norm_selpAR,'k');
% legend(La);
% axis([0 max(selected_pAR) 0 0.4]);
% grid on;
% xlabel ('por Aspect Ratio');
% ylabel ('normalized number of pors');
% title ('SELECTED PORS normalized por Aspect Ratio  distribution');
% n_selectedpors_used_AR=length(selected_pAR);



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
```

```
% fiber length sum
TotL=sum(DIST1);
TotLArea=TotL/Real_area;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% fiber length over the horizontal direction


% using a step of 0.1 microns
stepLX=0.1;
length_intX=(min(DIST2):stepL:max(DIST2));

% number of fibers used for fiber length estimation
n_fiber_used_lengthX=length(DIST2);
[nlengthX,length_intX]=hist(DIST2,length_intX);
norm_nlengthX=nlengthX/sum(nlengthX);
% mean
Length_aveX=sum(DIST2)/length(DIST2);
La=num2str(Length_aveX);
% standard deviation
stand_dev_LengthX=std(DIST2);
Lst=num2str(stand_dev_LengthX);
La=strcat('mean length = ','',La,'+/-',Lst,'','micrometers');
% figure(12);
% bar(length_intX,norm_nlengthX,'k');
% legend(La);
% axis([min(DIST2)-stepLX max(DIST2) 0 max(norm_nlengthX)+0.1]);
% grid on;
% xlabel ('node to node length along XP [micrometers]');
% ylabel ('normalized number of fibers');
% title ('normalized node to node length XP distribution');



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% fiber length over the vertical  direction


% using a step of 0.1 microns
stepLY=0.1;
length_intY=(min(DIST3):stepL:max(DIST3));

% number of fibers used for fiber length estimation
n_fiber_used_lengthY=length(DIST3);
[nlengthY,length_intY]=hist(DIST3,length_intY);
norm_nlengthY=nlengthY/sum(nlengthY);
% mean
Length_aveY=sum(DIST3)/length(DIST3);
La=num2str(Length_aveY);
% standard deviation
stand_dev_LengthY=std(DIST3);
Lst=num2str(stand_dev_LengthY);
La=strcat('mean length = ','',La,'+/-',Lst,'','micrometers');
```

```
% figure(13);
% bar(length_intY,norm_nlengthY,'k');
% legend(La);
% axis([min(DIST3)-stepLY max(DIST3) 0 max(norm_nlengthY)+0.1]);
% grid on;
% xlabel ('node to node length along PD [micrometers]');
% ylabel ('normalized number of fibers');
% title ('normalized node to node length PD distribution');



[tetastart,tetaend,n1fitted,x_histfitted,Mc,OI_sals_def] =
OI_SALS_2(x_hist,n1);


function [tetastart,tetaend,n1fitted,x_histfitted,Mc,OI_sals_def] =
OI_SALS_2(x_hist,n1);
%
% % clear
% % clc
% % close all
% % load('matrixdata');
stepangle=10;
x_hist=(0:stepangle:180);

n1fitted = fit(x_hist',n1,'cubicspline');
x_histfitted=[1:180];

%
for i=1:180-1
Mcnum(i)=(x_histfitted(i+1)-
min(x_histfitted)).*((n1fitted(i+1)+n1fitted(i))/2);
Mcden(i)=((n1fitted(i+1)+n1fitted(i))/2);
end
Mc=round(sum(Mcnum)/sum(Mcden));

if Mc>180
    Mc=round(Mc-180);
end


[a b]=min(abs(x_histfitted-Mc));
Mcdisc=x_histfitted(b);


Z = trapz(x_histfitted,n1fitted(1:180)); % total area

for j=1:(length(x_histfitted)/2)

    if (b-j>0)&&(b+j<181)
       Zrow(j) = trapz(x_histfitted(b-j:b+j),n1fitted(b-j:b+j));
    end
```

```
end

ZTOT=Zrow/Z;
% fiber angle distribution

[az bz]=min(abs(ZTOT-0.5));
tetastart=x_histfitted(b-bz);
tetaend=x_histfitted(b+bz);

OI_sals_def=tetaend-tetastart;


% %
% % figure(2)
% % plot(0:180,n1fitted(0:180)),hold
on,plot(0:stepangle:180,n1,'k'),plot(0:stepangle:180,n1,'.k')...
% %
,plot(Mc,0,'.r'),area((tetastart:tetaend),n1fitted((tetastart:tetaend)
)),hold off
% % axis([0 180 0 max(n1)+0.1])
% % legend('fitted','real data');
% % grid on
% % xlabel ('angle [degrees]');
% % ylabel ('normalized number of fibers');
% % title ('normalized orientation distribution + fitting');






Lst=num2str(OI_sals_def);
La=strcat('Orientation Index (SALS) = ','',Lst);
figure(14)
plot(0:180,n1fitted(0:180)),hold on...

,plot(Mc,0,'.r'),area((tetastart:tetaend),n1fitted((tetastart:tetaend)
)),hold off
axis([0 180 0 max(n1)+0.1])
% ,plot(0:stepangle:180,n1,'k'),plot(0:stepangle:180,n1,'.k')
legend(La);
grid on
xlabel ('angle [degrees]');
ylabel ('normalized number of fibers');
title ('normalized orientation distribution + fitting');



save('matrixdata','nodemap2');
clear nodemap2

%
%  bar scale on the image
% [aba bba]=size(original_image2(:,:,1));
```

```
% xbars=(bba-magn/100-50);
% ybars=(aba-magn/100-50);
% xbars=round(xbars);
% ybars=round(ybars);
% % original_image2(ybars-20-magn/100:ybars+25,xbars-20-
magn/100:xbars+20+magn/100,1:3)=0;
% original_image2(ybars:ybars+5,xbars:xbars+magn/100,1:3)=255;


% [aba bba]=size(original_image2(:,:,1));
% xbars=(bba-100);
% ybars=(aba-100);
% xbars=round(xbars);
% ybars=round(ybars);
% original_image2(ybars:ybars+5,xbars:xbars+magn/100,1:3)=255;
%









% % main direction of alignment
% yarrow=round(ybars-magn/100+(1:magn/100)*sin(tetadefault*pi/180));
% xarrow=round(xbars+(1:magn/100)*cos(tetadefault*pi/180));
%     original_image2(ybars-5-magn/100:ybars+5-magn/100,xbars,1)=255;
%     original_image2(ybars-5-magn/100:ybars+5-magn/100,xbars,2)=255;
%     original_image2(ybars-5-magn/100:ybars+5-magn/100,xbars,3)=0;
%     original_image2(ybars-magn/100,xbars-5:xbars+5,1)=255;
%     original_image2(ybars-magn/100,xbars-5:xbars+5,2)=255;
%     original_image2(ybars-magn/100,xbars-5:xbars+5,3)=0;
%




% for i=1:length(yarrow)
%     original_image2(yarrow(i):yarrow(i)+1,xarrow(i),1)=255;
%     original_image2(yarrow(i):yarrow(i)+1,xarrow(i),2)=255;
%     original_image2(yarrow(i):yarrow(i)+1,xarrow(i),3)=0;
% end


load('matrixdata')
save('matrixdata','nodemap2','original_image2');
clear original_image2
load('porefil.mat')
%  bar scale on the image
% [aba bba]=size(filter_por_vis(:,:,1));
% xbars=(bba-magn/100-50);
% ybars=(aba-magn/100-50);
% xbars=round(xbars);
% ybars=round(ybars);
```

```
% filter_por_vis(ybars:ybars+5,xbars:xbars+magn/100,1)=0;
% filter_por_vis(ybars:ybars+5,xbars:xbars+magn/100,2)=0;
% filter_por_vis(ybars:ybars+5,xbars:xbars+magn/100,3)=255;
% %


load('matrixdata')
save('matrixdata','nodemap2','original_image2','filter_por_vis');
clear filter_por_vis




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% save data

% uncomment the following if FEM analysis required
% save('nodesdata','nodes','-ascii');
% save('elementsdata','Etot','-ascii');
% save('stiffnessdata','Ktot','-ascii');
% save('angular_stiffnessdata','conn_norm','-ascii');
load('current_test_name')
load('matrixdata');

save('matrixdata','posfile','v','p','magn','good_friends','Fangle','DI
ST',...
    'Area','Real_area','Nnodes','nodes_density',...
    'Connections','connectivity','Conn_count_norm','conn_interval',...

'counter_weighted_norm','n1','OI','OIW','interval','value','weight','c
ounter',...

'length_int','norm_nlength','Length_ave','stand_dev_Length','n_fiber_u
sed_length',...

'stand_dev_Diam_used','Diam_ave_used','norm_ndiam','length_int_diam','
number_fiber_dia',...

'R1','porosity','norm_psizes','int_por_sizes','por_sizes_ave','stand_d
ev_por_sizes','npors_used',...
    'OI_por','norm_porient',...

'TotL','TotLArea','length_intX','norm_nlengthX','length_intY','norm_nl
engthY',...

'Length_aveY','Length_aveX','stand_dev_LengthY','stand_dev_LengthX',..
.

'norm_pAR','int_por_AR','npors_used_AR','por_AR_ave','stand_dev_por_AR
',...

'nodemap2','original_image2','filter_por_vis','OI_sals_def','Mc');%,..
.
```

```
%
'norm_selpAR','int_por_selectedAR','n_selectedpors_used_AR','selec_por
_AR_ave','stand_dev_selec_por_AR');

clear
% load('nodesdata')
% load('elementsdata')
% load('stiffnessdata')
% load('angular_stiffnessdata')
load('matrixdata');




figure(1)
imshow(nodemap2)
pixval on
title('final result on layer')

figure(2)
imshow(original_image2)
pixval on
title('final result')

figure(3)
imshow(filter_por_vis)
title('pores and their major axis on the detected scaffold layer')
```

APPENDIX B)

MESH GENERATOR CODE (MATLAB)

```
% function
[Nodes_densityS,OIS,counter_weighted_normS,nS,Conn_count_normS,Diameter
rS,Real_areaS,model_sizeS]=
fem_generator_9(TotalAreaModel,Element_size,DEF,ConnIndex,node_density
,fiber_diameter,magn);

%
[Nodes_densityS,OIS,counter_weighted_normS,nS,Conn_count_normS,Diameter
rS,Real_areaS,model_sizeS]=
fem_generator_9(1000000,1000,0,0,0.46,0.32,3500);

%
[Nodes_densityS,OIS,counter_weighted_normS,nS,Conn_count_normS,Diameter
rS,Real_areaS,model_sizeS]=
fem_generator_9(9000000,3000,0,0,0.46,0.32,3500);

% tic

% % function inputs typical for isotropic scaffold 1.5

magn=2500;
node_density=0.1842;
fiber_diameter=0.4790;
DEF=0.59;
ConnIndex=0;
TotalAreaModel=1000000;
Element_size=1000;


% function outputs
% %
% % Nodes_densityS------------OK
% % OIS-----------------------OK
% % counter_weighted_normS----OK
% % nS------------------------OK
% % Conn_count_normS----------OK
% % DiameterS-----------------OK
% % Real_areaS ---------------OK
% % model_sizeS --------------OK


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

%MODEL GENERATOR MAIN BODY
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% MODEL INPUTS



save('data_magn','magn')
% clear
% load('data_magn')

DiameterS=fiber_diameter;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
    %  NETWORK SIMULATION MAIN SETTINGS

     %input 1)
     % magn


     %input 2)

     % converted node_density from N/microns^2 into N/pixels^2

     node_density=node_density/(magn/100*magn/100);

     %input 3)

     % converted fiber diameter from microns into pixels length

     fiber_diameter =round(fiber_diameter*magn/100);

     % external borders size in pixels required for visualization
purposes only
     bou=500;

     % CONTROL PARAMETERS

     %input 4)
     % (I a) LEVEL OF FIBER ALIGNMENT DESIDERED
     %       DEF;
            % patches dimensions to feel holes due to subareas network
definition
```

```
            % originally patchstep 250 and patchstep/5 X ---
patchstep/3 Y

            patchstep=150;  % 150 is the minimum to get a random
configuration
            %within the patch
            %patch x direction is reduced when 90 is the main
direction of
            %alignment
            patchx=patchstep-DEF*patchstep/2;
            %patch y direction is icreased when 90 is the main
direction of
            %alignment
            patchy=patchstep+DEF*patchstep/2;

            patchstepMAX=max(patchx,patchy);

            % (I b)
            % correction index, deformation greater than 5% along the
cross preferred dir
            % will be erased   originally 5% (0.05) it compensates the
creation of
            % alignment in the XD
%             counttime=0;
%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=100;
%
            % set a constant value if required
%             CI=0.01;
            % function of DEF if the alignment goes up the CI goes
down
            CI=0.05*DEF/0.7;
            % the 0.7 constant was selected on a trial and error base
            % improve this choice if any time left

            % (I c)
            % correction angle for CI , originally was 25
            % (eliminate the fibers with a CI% stretch on the
direction perpendiular
            % to the main alignment +- an angle of range)
            %range=60;
            % function of DEF if the alignment goes up the range goes
up too
            range=30*DEF/0.7;   %20 and 0.7 was selected improve this
choice if any time left

     % LEVEL OF CONNECTIVITY DESIDERED
     % Connectivity control index from 0 (0% eliminated)to 1 (100%
eliminated)
     % fibers to be eliminated are randomly selected
     %(II)

     %input 5)
     % ConnIndex;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
    % MODEL AREA
    %(III)

        % MACRO target= 3 X 3 mm
        % for 9.0 magnification of 2500 this means 75000 x 75000
pixels
        % for 1.5 and 4.5 magnification of 3500 this means 95000 x
95000 pixels
        % MESO target= 0.5 X 0.5 mm
        % for 9.0 magnification of 2500 this means 12500 x 12500
pixels
        % for 1.5 and 4.5 magnification of 3500 this means 17500 x
17500 pixels

        % Total Model Area in pixels
        %  TotalAreaModel
        %  Element_size

        % ASSUMING TO GENERATE SQAURE ONLY Element_sizeX=Element_sizeY
        save('model_size','TotalAreaModel','Element_size','bou');
        %number of sub areas
        Nmatrix=TotalAreaModel/(Element_size*Element_size);
        % number of sub areas per side
        SNmatrix=sqrt(Nmatrix);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
        % Real area in square micrometers
        Area_new=Element_size*Element_size*Nmatrix;
        %magnification factor used
        magn_fact=magn/100;
        % real model area and size
%         disp('model area in micrometers^2 ')
        model_area=Area_new/(magn_fact^2);
%         disp(model_area)
        Real_areaS=model_area;

%         disp('model size in micrometers')
        model_size=(Element_size*SNmatrix)/magn_fact;
%         disp(model_size)
        model_sizeS=model_size;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
```

```
% BOUNDARIES
Ln_new=Element_size;
Lm_new=Element_size;

% subareas definition for fiber alignment procedure
% X EXTENSION
% 15 subareas every 1500 pixels
xsubareas=Ln_new*(15/1500);
 stepx=round(Ln_new/xsubareas);
% dispacement of DEF% subareas x length (originally was 70%  0.7)
 deg_alx=round(DEF*(stepx/2));

% Y EXTENSION
% 7 subareas every 1500 pixels
 ysubareas=Lm_new*(15/1500);
 stepy=round(Lm_new/ysubareas);
  % dispacement of DEF% subareas x length (originally was 70%  0.7)
 deg_aly=round(DEF*(stepy/2));

 %creates a matrix to identify areas boundaries and neighbours
borders=zeros(SNmatrix+2,SNmatrix+2);
borders(2:SNmatrix+1,2:SNmatrix+1)=1;
% disp(borders(2:SNmatrix+1,2:SNmatrix+1));

fiber_diameter_pixels=fiber_diameter;
save('SIMULATIONDATA','magn','node_density','fiber_diameter_pixels','b
ou','DEF','CI','range','ConnIndex','TotalAreaModel','Element_size','bo
rders');

%borders for points at the borders identification
borders_conn_hor_right=borders;
borders_conn_hor_left=borders;
borders_conn_vert_up=borders;
borders_conn_vert_down=borders;

%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=203;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%STARTS SUB-AREAS ---- NETWORKS GENERATION 1/2
% sub areas generation
% total nodes counter
Nodes_counter=0;
% subareas counter
Mcounter=1;
for i=1:SNmatrix+2
    for j=1:SNmatrix+2
        %generates areas only if inside the borders matrix
        if borders(i,j)==1
```

```
%             i
%             j
            %areas offsets, offsets depend also on applied deformation
(deg_alx and deg_aly)
            Xoffset=(j-2)*Element_size - round((j-2)*deg_alx);
%             disp(Xoffset)
            Yoffset=(i-2)*Element_size + round((i-2)*deg_aly);
%             disp(Yoffset)
           [Nodes_counter_done] =
orientationadaptation_comp29_novis1(node_density,Xoffset,Yoffset,i,j,E
lement_size,Mcounter,Nodes_counter,DEF,fiber_diameter,CI,bou,ConnIndex
,range,patchstepMAX);
           % Nodes_counter counts the total number of nodes
           Nodes_counter=Nodes_counter_done;
           % Mcounter counts the total number of areas
           Mcounter=Mcounter+1;
        end
    end
end
%ENDS SUB-AREAS  -----   NETWORKS GENERATION 1/2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=239;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%STARTS SUB-AREAS CONNECTING  -----  NETWORKS GENERATION 2/2
new_conn_count=1;
conn_subareas_count=0;


for i=1:SNmatrix+2
    for j=1:SNmatrix+2
        % operates on original sub networks only
        if borders(i,j)==1
            % UP AND
DOWN%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            for k=i-1:i+1   % check up and down the analyzed element
(i,j)
                if k~=i     % to eliminate the element position i

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                        % DOWN
                        if
((k>i)&&(borders_conn_vert_down(i,j)==1)&&(borders(k,j)==1)) % looking
down with respect to subarea (i,j)
                            % load (i,j) data analyzed subarea
                            str1c = int2str(i);
                            str2c = int2str(j);
```

```
                                filename1=('FEM_nodesdata_');
                                filename2=('FEM_conn_nodesdata_');
                                str3c='_';
                                S1 = (strcat(filename1,str1c,str3c,str2c));
                                 % load (k,j) data subarea below (i,j)
                                str1v = int2str(k);
                                str2v = int2str(j);
                                str3v='_';
                                S2 = (strcat(filename1,str1v,str3v,str2v));
                                Acenter=load (S1);
                                Avert=load(S2);
                                % subareas overlapping areas definition
                                conn_subareas_count=conn_subareas_count+1;
                                Acent2=max(Acenter(:,3));
                                Acent1=Acent2-patchy;

                                Avert2=min(Avert(:,3));
                                Avert1=Avert2+patchy;
                                [ac,bc]=size(Acenter);
                               % nodes of (i,j) at the boundary with (k,j)
                               for p=1:ac
                                   if
(Acenter(p,3)<=Acent2)&&(Acenter(p,3)>=Acent1)

New_conn_net(new_conn_count,1)=Acenter(p,1);

New_conn_net(new_conn_count,2)=Acenter(p,2);

New_conn_net(new_conn_count,3)=Acenter(p,3);
                                       new_conn_count=new_conn_count+1;
                                   end
                               end
                               [av,bv]=size(Avert);
                               % nodes of (k,j) at the boundary with (i,j)
                               for p=1:av
                                   if
(Avert(p,3)<=Avert1)&&(Avert(p,3)>=Avert2)

New_conn_net(new_conn_count,1)=Avert(p,1);

New_conn_net(new_conn_count,2)=Avert(p,2);

New_conn_net(new_conn_count,3)=Avert(p,3);
                                       new_conn_count=new_conn_count+1;
                                   end
                               end
                           id = int2str(conn_subareas_count);
                           Sstore = (strcat(filename2,str3c,id));
                           save(Sstore,'New_conn_net');     % save sub-
areas connecting zones points network creation
                           new_conn_count=1;
                           clear New_conn_net
                           borders_conn_vert_down(i,j)=0; % this avoid to
connect twice the same set
                           borders_conn_vert_up(k,j)=0;
                           end  % if k > i
```

```
                        % UP
                        if
((k<i)&&(borders_conn_vert_up(i,j)==1)&&(borders(k,j)==1))  % looking
up with respect to subarea (i,j)
                            str1c = int2str(i);
                            str2c = int2str(j);
                            filename1=('FEM_nodesdata_');
                            filename2=('FEM_conn_nodesdata_');
                            str3c='_';
                            S1 = (strcat(filename1,str1c,str3c,str2c));
                            str1v = int2str(k);
                            str2v = int2str(j);
                            str3v='_';
                            S2 = (strcat(filename1,str1v,str3v,str2v));
                            Acenter=load (S1);
                            Avert=load(S2);
                           conn_subareas_count=conn_subareas_count+1;
                           Acent2=min(Acenter(:,3));
                           Acent1=Acent2+patchy;
                           Avert2=max(Avert(:,3));
                           Avert1=Avert2-patchy;
                           [ac,bc]=size(Acenter);
                           % nodes of (i,j) at the boundary with (k,j)
                           for p=1:ac
                               if
(Acenter(p,3)>=Acent2)&&(Acenter(p,3)<=Acent1)

New_conn_net(new_conn_count,1)=Acenter(p,1);

New_conn_net(new_conn_count,2)=Acenter(p,2);

New_conn_net(new_conn_count,3)=Acenter(p,3);
                                    new_conn_count=new_conn_count+1;
                               end
                           end
                           [av,bv]=size(Avert);
                           % nodes of (k,j) at the boundary with (i,j)
                           for p=1:av
                               if
(Avert(p,3)>=Avert1)&&(Avert(p,3)<=Avert2)

New_conn_net(new_conn_count,1)=Avert(p,1);

New_conn_net(new_conn_count,2)=Avert(p,2);

New_conn_net(new_conn_count,3)=Avert(p,3);
                                    new_conn_count=new_conn_count+1;
                               end
                           end
                       id = int2str(conn_subareas_count);
                       Sstore = (strcat(filename2,str3c,id));
                       save(Sstore,'New_conn_net');     % save sub-
areas connecting zones points network creation
                       new_conn_count=1;
                       clear New_conn_net
```

```
                        borders_conn_vert_up(i,j)=0; % this avoid to
connect twice the same set
                        borders_conn_vert_down(k,j)=0;
                        end % if k < i
%                       end
                end
            end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

            % LEFT AND
RIGHT%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            for k=j-1:j+1   % check left and right the analyzed
elelment
                if k~=j
                        %RIGHT
                        if
((k>j)&&(borders_conn_hor_right(i,j)==1)&&(borders(i,k)==1))      %
looking right with respect to subarea (i,j)
                            str1c = int2str(i);
                            str2c = int2str(j);
                            filename1=('FEM_nodesdata_');
                            filename2=('FEM_conn_nodesdata_');
                            str3c='_';
                            S1 = (strcat(filename1,str1c,str3c,str2c));
                            str1v = int2str(i);
                            str2v = int2str(k);
                            str3v='_';
                            S2 = (strcat(filename1,str1v,str3v,str2v));
                            Acenter=load(S1);
                            Ahor=load(S2);
                            conn_subareas_count=conn_subareas_count+1;
                           Acent2=max(Acenter(:,2));
                           Acent1=Acent2-patchx;
                           Ahor2=min(Ahor(:,2));
                           Ahor1=Ahor2+patchx;
                           [ac,bc]=size(Acenter);
                           % nodes of (i,j) at the boundary with (i,k)
                           for p=1:ac
                               if
(Acenter(p,2)<=Acent2)&&(Acenter(p,2)>=Acent1)

New_conn_net(new_conn_count,1)=Acenter(p,1);

New_conn_net(new_conn_count,2)=Acenter(p,2);

New_conn_net(new_conn_count,3)=Acenter(p,3);
                                    new_conn_count=new_conn_count+1;
                               end
                           end
                           [av,bv]=size(Ahor);
                           % nodes of (i,k) at the boundary with (i,j)
                           for p=1:av
                               if
(Ahor(p,2)<=Ahor1)&&(Ahor(p,2)>=Ahor2)

New_conn_net(new_conn_count,1)=Ahor(p,1);
```

```
New_conn_net(new_conn_count,2)=Ahor(p,2);

New_conn_net(new_conn_count,3)=Ahor(p,3);
                                    new_conn_count=new_conn_count+1;
                                end
                            end
                        id = int2str(conn_subareas_count);
                        Sstore = (strcat(filename2,str3c,id));
                        save(Sstore,'New_conn_net');     % save sub-
areas connecting zones points network creation
                        new_conn_count=1;
                        clear New_conn_net
                        borders_conn_hor_right(i,j)=0; % this avoid to
connect twice the same set
                        borders_conn_hor_left(i,k)=0;
                        end  % if k > i

                        %LEFT
                        if
((k<j)&&(borders_conn_hor_left(i,j)==1)&&(borders(i,k)==1))        %
looking left with respect to subarea (i,j)
                             str1c = int2str(i);
                             str2c = int2str(j);
                             filename1=('FEM_nodesdata_');
                             filename2=('FEM_conn_nodesdata_');
                             str3c='_';
                             S1 = (strcat(filename1,str1c,str3c,str2c));
                             str1v = int2str(i);
                             str2v = int2str(k);
                             str3v='_';
                             S2 = (strcat(filename1,str1v,str3v,str2v));
                             Acenter=load(S1);
                             Ahor=load(S2);
                             conn_subareas_count=conn_subareas_count+1;
                            Acent2=min(Acenter(:,2));
                            Acent1=Acent2+patchx;
                            Ahor2=max(Ahor(:,2));
                            Ahor1=Ahor2-patchx;
                            [ac,bc]=size(Acenter);
                            % nodes of (i,j) at the boundary with (i,k)
                            for p=1:ac
                                if
(Acenter(p,2)>=Acent2)&&(Acenter(p,2)<=Acent1)

New_conn_net(new_conn_count,1)=Acenter(p,1);

New_conn_net(new_conn_count,2)=Acenter(p,2);

New_conn_net(new_conn_count,3)=Acenter(p,3);
                                    new_conn_count=new_conn_count+1;
                                end
                            end
                            [av,bv]=size(Ahor);
                            % nodes of (i,k) at the boundary with (i,j)
                            for p=1:av
```

```
                                if
(Ahor(p,2)>=Ahor1)&&(Ahor(p,2)<=Ahor2)

New_conn_net(new_conn_count,1)=Ahor(p,1);

New_conn_net(new_conn_count,2)=Ahor(p,2);

New_conn_net(new_conn_count,3)=Ahor(p,3);
                                    new_conn_count=new_conn_count+1;
                                end
                            end
                        id = int2str(conn_subareas_count);
                        Sstore = (strcat(filename2,str3c,id));
                        save(Sstore,'New_conn_net');     % save sub-
areas connecting zones points network creation
                        new_conn_count=1;
                        clear New_conn_net
                        borders_conn_hor_left(i,j)=0; % this avoid to
connect twice the same set
                        borders_conn_hor_right(i,k)=0;
                        end % if k < i
%                       end %
                end % end if k
            end % end for k
        end % end if border
    end % end main for j
end % end main for i

% save('New_conn_net','New_conn_net');


%
%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=467;

%ENDS SUB-AREAS ---- NETWORKS GENERATION 2/2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% VISUALIZATION & GEOMETRY DATA COLLECTION
% Visualize all the created struts and nodes to verify correctivness
load('model_size');
%number of sub areas
Nmatrix=TotalAreaModel/(Element_size*Element_size);
% number of sub areas per side
```

```
SNmatrix=sqrt(Nmatrix);


%
finalmatrix=zeros(Element_size*SNmatrix+2*bou,Element_size*SNmatrix+2*
bou);


startvalue2=1;
storagecounter=0;
for i=1:SNmatrix+2
    for j=1:SNmatrix+2
        if borders(i,j)==1
        storagecounter=storagecounter+1;
        str1 = int2str(i);
        str2 = int2str(j);
        filename1=('FEM_nodesdata_');
        filename2=('angledata_');
        filename3=('geodata_');
        str3='_';
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %load
        % FEM_nodesdata
        S1 = (strcat(filename1,str1,str3,str2));
        Anodes=load (S1);
        % angledata
        S2 = (strcat(filename2,str1,str3,str2));
        load (S2);
        % geodata
        S3 = (strcat(filename3,str1,str3,str2));
        load (S3);
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %Visualizer

%         [finalmatrixdone] =
networkvisualizer_single5_novis1(good_friends,Anodes,Element_size,SNma
trix,borders,finalmatrix,Mcounter,fiber_diameter,bou);
%         finalmatrix=finalmatrixdone;
%
        Mcounter=Mcounter+1;
        % Angle distribution
        n2TOT(storagecounter,1:length(n2))=n2;
        n2TOTinterval=xout2;
        % Angle values
%         value2TOT(storagecounter,1:length(value2))=value2;
        value2TOT(startvalue2:startvalue2+length(value2)-1)=value2;
        startvalue2=startvalue2+length(value2);
        value2TOTinterval=xout2;
        % Weighted angle distribution

n3TOT(storagecounter,1:length(counter_weighted3))=counter_weighted3;
        n3TOTinterval=interval3;
        % Connectivity distribution

Conn_count_newTOT(storagecounter,1:length(Conn_count_new))=Conn_count_
new;
```

```
        conn_interval_newTOT=conn_interval_new;
        end
    end
end
% figure(1)
% imshow(finalmatrixdone);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% CONNECTING NETWORKS CREATION
% additional connection creation and storing
aa=0;
for vi=1:conn_subareas_count
    storagecounter=storagecounter+1;

    filename2=('FEM_conn_nodesdata_');
    str3c='_';
    id = int2str(vi);
    Sstore = (strcat(filename2,str3c,id));
    filename3=('FEMelements');

    Sstorelements =(strcat(filename3,str3c,id));
    clear New_conn_net
    load (Sstore)

[FEMelements,n,xout,value,Conn_count,conn_interval,counter_weighted] =
orientationadaptation_con_net15_novis1(New_conn_net,fiber_diameter,bou
,patchstepMAX,ConnIndex,Ln_new,Lm_new);
    a=length(FEMelements);
    ba(vi)=a;
    aa=aa+a;
%     finalmatrixdone=finalmatrix;
    save (Sstorelements,'FEMelements');
    % storage of connetting subareas elements
%     clear finalmatrix
    % Angle distribution
    n2TOT(storagecounter,1:length(n))=n;
    n2TOTinterval=xout;
    % Angle values
%     value2TOT(storagecounter,1:length(value))=value;
    value2TOT(startvalue2:startvalue2+length(value)-1)=value;
    startvalue2=startvalue2+length(value);
    value2TOTinterval=xout;
    % Angle weighted distribution
    n3TOT(storagecounter,1:length(counter_weighted))=counter_weighted;
    n3TOTinterval=xout;
    % Connectivity distribution
    Conn_count_newTOT(storagecounter,1:length(Conn_count))=Conn_count;
    conn_interval_newTOT=conn_interval_new;
end
%     figure(2)
%     imshow(finalmatrixdone);



%             counttime=counttime+1;
```

```
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=586;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% FEM DATA
% FEM data generator in ASCII format
kA=1;
kB=1;
for i=1:SNmatrix+2
    for j=1:SNmatrix+2
        if borders(i,j)==1
        str1 = int2str(i);
        str2 = int2str(j);
        filename1=('FEM_nodesdata_');
        filename2=('FEM_elementsdata_');
        str3='_';
        S1 = (strcat(filename1,str1,str3,str2));

        % nodes FEM data formation
        A=load (S1);
        idA=A(length(A),1);
        Anodes(kA:idA,1:3)=A;
        kA=idA+1;

        % elements FEM data formation original subareas
        S2 = (strcat(filename2,str1,str3,str2));
        B=load (S2);
        [idB b]=size(B);
        Aelements(kB:kB+idB-1,1:2)=B;
        kB=idB+kB;

        end
    end
end
save('FEM_nodesdata_TOT','Anodes','-ascii');
load('FEM_nodesdata_TOT');

% save('FEM_elementsdata_TOT','Aelements','-ascii');
% clear
% load('FEM_elementsdata_TOT');



 % elements FEM data formation connecting subareas
kB=1;
for vi=1:conn_subareas_count
    str3c='_';
    id = int2str(vi);
    filename3=('FEMelements');
```

```
    Sstorelements =(strcat(filename3,str3c,id));
    load (Sstorelements)
    BO=FEMelements;

    [idB b]=size(BO);
    Aelements2(kB:kB+idB-1,1:2)=BO;
    kB=idB+kB;
    clear FEMelements
    clear BO
end


if sum(sum(borders))==1
    % if to verify if subareas have been generated
   AelementsTOT=Aelements;
else
   AelementsTOT=[Aelements;Aelements2];
end
 save('FEM_elementsdata_TOT','AelementsTOT','-ascii');
 load('FEM_elementsdata_TOT');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%   MODEL GENERATED --> FROM NOW ON DATA COLLECTION AND STORAGE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% ARCHITECTURE DESCRIPTION
% get the global geometry  info
% disp('MODEL ARCHITECTURAL DESCRIPTORS')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% Angular Stiffness data
clear FEM_angular_stiffness_TOT
FEM_angular_stiffness_TOT(:,1)=FEM_nodesdata_TOT(:,1);
FEM_angular_stiffness_TOT(:,2)=0;

for i=1:length(FEM_nodesdata_TOT)
    for j=1:length(FEM_elementsdata_TOT)
          if
(FEM_nodesdata_TOT(i,1)==FEM_elementsdata_TOT(j,1))||(FEM_nodesdata_TO
T(i,1)==FEM_elementsdata_TOT(j,2))

FEM_angular_stiffness_TOT(i,2)=FEM_angular_stiffness_TOT(i,2)+1;
          end
    end
end
save('FEM_angular_stiffness_TOT','FEM_angular_stiffness_TOT','-
ascii');
```

```
%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=685;

%%%%%%%%%%%%%%
% connectivity

% total nuber of overlaps
total_number_overlaps=length(FEM_nodesdata_TOT);

%connectivity plot
Conn_count_newTOT=zeros(1,51);
for i=1:length(FEM_angular_stiffness_TOT)
    for j=1:51
        conncheck=j-1;
        if FEM_angular_stiffness_TOT(i,2)==conncheck;
            Conn_count_newTOT(1,j)=Conn_count_newTOT(1,j)+1;
        end
    end
end

% total number of lost overlaps
overlaps_lost=Conn_count_newTOT(1);
% check they must be the same
overlap_lost1=0;
for i=1:length(FEM_angular_stiffness_TOT)
    if FEM_angular_stiffness_TOT(i,2)==0
        overlap_lost1=overlap_lost1+1;
    end
end

% total number of bar elements
Number_bar_elements=sum(Conn_count_newTOT.*(0:(length(Conn_count_newTO
T)-1)))/2;
% check they must be the same
Number_bar_elements1=length(FEM_elementsdata_TOT);

% total number of overlaps including the lost overlaps
Number_nodes_conn=sum(Conn_count_newTOT);
% check they must be the same
Number_nodes_conn1=length(FEM_nodesdata_TOT);

%connectiviy plot
% Conn_count_newTOT=Conn_count_newTOT/(sum(Conn_count_newTOT));
[acon bcon]=find(Conn_count_newTOT>0);
xlimitmin=conn_interval_newTOT(min(bcon));
xlimitmax=conn_interval_newTOT(max(bcon));
ylimit=max(Conn_count_newTOT);
% ylimit=1;
% figure(3)
% bar(conn_interval_newTOT,Conn_count_newTOT,'k')
% axis([xlimitmin-1 xlimitmax+1 0 ylimit])
% title ('MODEL normalized  number of fiber overlaps VS connections')
```

```
% xlabel('connections')
% ylabel('normalized number of fiber overlaps')
Conn_count_normS=Conn_count_newTOT;
%%%%%%%%%%%%%%%
% overlaps/area

% disp('overlaps density [N overlaps/square micrometers]')
Noverlaps=Number_nodes_conn;
% overlaps number needs to be reduced because some connections could
be
% lost during the processing
Eff_Noverlaps=Noverlaps-overlaps_lost;
overlaps_density=Eff_Noverlaps/model_area;
% disp(overlaps_density);
Nodes_densityS=overlaps_density;
%%%%%%%%%%%%%%%%%%%%%%%%%%
% orientation index from fiber anlge values

    %  fiber angles values
%     [a b]=size(value2TOT);
%     value2TOT=sum(value2TOT((1:a),:));
    value2TOT=value2TOT(:);
%     tetadefault=input('please provide the supposed orientation angle
');

    tetadefault=90;
    deltateta=(value2TOT-tetadefault);
    deltatetaradiant = (2*pi/360)*deltateta;
    oi=sum((cos(deltatetaradiant).^2));
    OI=oi/length(value2TOT);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% fiber angle distribution

    % normalized fiber alignment
    [a b]=size(n2TOT);
    if a>1
    n2TOT=sum(n2TOT((1:a),:));

    end
%normalization
    n2TOT=n2TOT/(sum(n2TOT));
    ylimit=max(n2TOT)+0.01;
%     ylimit=0.5;
%     figure(4)
%     bar(n2TOTinterval,n2TOT,'k')
%     axis([0 180 0 ylimit])
%     title ('MODEL normalized fiber count VS angle')
%     xlabel('angle')
%     ylabel('normalized fiber count')
    nS=n2TOT;

%
%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=788;
```

```
%     amid=round(length(n2TOT)/2);
%     text(n2TOT(amid),n2TOTinterval(amid),[num2str(OI),' Orientation
Index'],'FontSize',10);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% fiber angle weighted distribution

    % normalized fiber alignment
    [a b]=size(n3TOT);
    n3TOT=sum(n3TOT((1:a),:));
    % normalization
        n3TOT=n3TOT/(sum(n3TOT));
%         ylimit=max(n3TOT)+0.1;

%         figure(5)
%         bar(n3TOTinterval,n3TOT,'k')
%         axis([0 180 0 ylimit])
%         title ('MODEL normalized weighted fiber count VS angle')
%         xlabel('angle')
%         ylabel('normalized weighted fiber count')


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%

        n2TOTNEW(1)=sum(n2TOT(1:2));
        n2TOTNEW(2)=sum(n2TOT(3:4));
        n2TOTNEW(3)=sum(n2TOT(5:6));
        n2TOTNEW(4)=sum(n2TOT(7:8));
        n2TOTNEW(5)=sum(n2TOT(9:10));
        n2TOTNEW(6)=sum(n2TOT(11:12));
        n2TOTNEW(7)=sum(n2TOT(13:14));
        n2TOTNEW(8)=sum(n2TOT(15:16));
        n2TOTNEW(9)=sum(n2TOT(17:18));
        n2TOTNEW(10)=sum(n2TOT(19:20));
        n2TOTNEW(11)=sum(n2TOT(21:22));
        n2TOTNEW(12)=sum(n2TOT(23:24));
        n2TOTNEW(13)=sum(n2TOT(25:26));
        n2TOTNEW(14)=sum(n2TOT(27:28));
        n2TOTNEW(15)=sum(n2TOT(29:30));
        n2TOTNEW(16)=sum(n2TOT(31:32));
        n2TOTNEW(17)=sum(n2TOT(33:34));
        n2TOTNEW(18)=sum(n2TOT(35:36));
        n2TOTNEW(19)=sum(n2TOT(37));



        n2TOTintervalNEW=0:10:180;
```

```
        n2TOTNEW=n2TOTNEW/(sum(n2TOTNEW));
%         ylimit=max(n3TOTNEW);
%         ylimit=0.4;

%         figure(6)
%         ylimit=max(n2TOTNEW)+0.01;
%         bar(n2TOTintervalNEW,n2TOTNEW,'k')
%         axis([0 180 0 ylimit])
%         title ('MODEL normalized fiber count VS angle')
%         xlabel('angle')
%         ylabel('normalized fiber count')
        counter_weighted_normS=n2TOTNEW;

%%%%%%%%%%%%%%%%
% model geometry

% disp('model area in micrometers^2 ')

model_area=Area_new/(magn_fact^2);
% disp(model_area)

% disp('model size in micrometers')

model_size=(Element_size*SNmatrix)/magn_fact;
% disp(model_size)
%
% disp('Orientation Index')
% disp(OI)
OIS=OI;

% fiber diameter again converted in micrometers before being stored

fiber_diameter=fiber_diameter*100/magn;

% store final geometry information
save('MODELDATA','model_sizeS','Real_areaS','Nodes_densityS','Noverlap
s','Eff_Noverlaps','overlaps_lost','Conn_count_normS','conn_interval_n
ewTOT','n2TOTinterval','counter_weighted_normS','nS','OIS','Number_bar
_elements','DiameterS')


% % Nodes_densityS------------OK
% % OIS-----------------------OK
% % counter_weighted_normS----OK
% % nS------------------------OK
% % Conn_count_normS----------OK
% % DiameterS-----------------OK
% % Real_areaS ---------------OK
% % model_sizeS --------------OK


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% ASCII

for i=1:length(FEM_nodesdata_TOT)
    A=[FEM_nodesdata_TOT(i,1) FEM_nodesdata_TOT(i,2)
FEM_nodesdata_TOT(i,3)];
    FEM_ndata_TOT_ST(i,1:3)=A;

end

% convert to real size data, all in micrometers
FEM_nodesdata_TOT(:,2)=FEM_nodesdata_TOT(:,2)*100/magn;
FEM_nodesdata_TOT(:,3)=FEM_nodesdata_TOT(:,3)*100/magn;

FEM_ndata_TOT_ST(:,2)=FEM_ndata_TOT_ST(:,2)*100/magn;
FEM_ndata_TOT_ST(:,3)=FEM_ndata_TOT_ST(:,3)*100/magn;


save('FEM_nodesdata_TOT','FEM_nodesdata_TOT','-ascii');
dlmwrite('FEM_ndata_TOT_ST.txt',FEM_ndata_TOT_ST, 'delimiter', ',')




for i=1:length(FEM_elementsdata_TOT)
    A=[FEM_elementsdata_TOT(i,1) FEM_elementsdata_TOT(i,2)];
    FEM_edata_TOT_ST(i,1:2)=A;

end
dlmwrite('FEM_edata_TOT_ST.txt',FEM_edata_TOT_ST, 'delimiter', ',')



for i=1:length(FEM_angular_stiffness_TOT)
    A=[FEM_angular_stiffness_TOT(i,1) FEM_angular_stiffness_TOT(i,2)];
    FEM_angular_stiffness_TOT_ST(i,1:2)=A;

end
dlmwrite('FEM_angular_stiffness_TOT_ST.txt',FEM_angular_stiffness_TOT_
ST, 'delimiter', ',')

% re-format the elements and nodes filese in the ANSYS required format

nodes=dlmread('FEM_ndata_TOT_ST.txt');
nodesfinal=[nodes(:,1) nodes(:,2)/1E6 nodes(:,3)/1E6];
fid=fopen('nodes.txt', 'wt');
fprintf(fid, '%8d %16.8f %16.8f\n', nodesfinal');
fclose(fid);

elements=dlmread('FEM_edata_TOT_ST.txt');
i=length(elements);
```

```
A=[elements zeros(i,6) ones(i,4) zeros(i,1)]; %8 nodes defining the
element, since we are using links/beams only need 2, 4 ones are Mat,
type, Real, secnum, Esys = 0,element number.
B=1:i; % generates element number
C=[A B']; %compiles the matrix
dlmwrite('elements.txt',C);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%

% check variables

% Number_nodes_conn
% Number_nodes_conn1
%
% overlaps_lost
% overlap_lost1
%
% Eff_Noverlaps
%
% Number_bar_elements
% Number_bar_elements1
%

%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=945;
%             FX = diff(timeperf);
%             figure(7)
%             grid on
%             plot(linetemp,timeperf,'xb');hold on
%             plot(linetemp,timeperf,'b');
%             plot(linetemp(1:length(linetemp)-1),FX,'xr');
%             plot(linetemp(1:length(linetemp)-1),FX,'r');hold off
% %
%             title ('algorithm performance evaluation, code vs time
and its gradient')
%             xlabel('code line')
%             ylabel('time [seconds]')
clear
load('FEM_nodesdata_TOT');
load('FEM_ndata_TOT_ST.txt');

load('FEM_elementsdata_TOT');
load('FEM_edata_TOT_ST.txt');

load('FEM_angular_stiffness_TOT');
load('FEM_angular_stiffness_TOT_ST.txt');

load('MODELDATA')
load('SIMULATIONDATA')
             %%%%%%%%%%%%%%%%%%%%%%%%%
            % debug completed May 1st 2009
             %%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function [Nodes_counter_done] =
orientationadaptation_comp29_novis1(node_density,Xoffset,Yoffset,gen_p
os_m,gen_pos_n,Element_size,Mcounter,Nodes_counter,DEF,fiber_diameter,
CI,bou,ConnIndex,range,patchstepMAX)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%% load initial data
% clc
% clear
% close all
warning off
%%%
%             tic
%             counttime=0;



save('data_step1','node_density','Xoffset','Yoffset','gen_pos_m','gen_
pos_n','Element_size','Mcounter','Nodes_counter','DEF','fiber_diameter
','CI','bou','ConnIndex','range','patchstepMAX','fiber_diameter')
% save('timevaluation','counttime');
clear
load('data_step1')
% load('timevaluation')
%%%
%%%%%%%%%%%%%%%%%%%%%%%% STEP 1 NODES CLOUD
GENERATION%%%%%%%%%%%%%%%%%%%%
% nodes density = nodes/area pixels

Ln_new=Element_size;
Lm_new=Element_size;
%new area in pixels
Area_new=Lm_new*Ln_new;
% magn_fact=magn/100;
%new number of nodes
Nnodes_new=round(node_density*Area_new);
% using the discrete uniform distribution DUD
%X coordinates random points
n_nodes_new = unidrnd(Ln_new,1,Nnodes_new);

%     npopulation=[1:Ln_new];
%     y = randsample(npopulation,Nnodes_new);
%     n_nodes_new=y;
%     n_nodes_new = round(n_nodes_new);
%     clear y
%
%Y coordinates random points
m_nodes_new = unidrnd(Lm_new,1,Nnodes_new);

%     mpopulation=[1:Lm_new];
%     y = randsample(mpopulation,Nnodes_new);
%     m_nodes_new=y;
%     m_nodes_new = round(m_nodes_new);
%     clear y
```

```
%


%enlarge the visualization area to visualize the nodes
% % % Anew=zeros(Lm_new+2*bou,Ln_new+2*bou);
% % % xini=bou;
% % % xend=Ln_new+bou;
% % % yini=bou;
% % % yend=Lm_new+bou;
% D=bou;
% the white line will identify the real area
% % % v(1)=yini;
% % % v(2)=yend;
% % % p(1)=xini;
% % % p(2)=xend;
% % % Anew(v(1),p(1):p(2))=255;
% % % Anew(v(2),p(1):p(2))=255;
% % % Anew(v(1):v(2),p(1))=255;
% % % Anew(v(1):v(2),p(2))=255;
% % % Anew3 for visualiztion issues (called Anew3 to recall the old
implementation where 2 differents matrixes were required )
% % % Anew3=Anew;
% % % clear Anew
% save('A_data3','Anew3','Mcounter');
save('A_data3','Mcounter');
%%%


save('data_step2','Nnodes_new','n_nodes_new','m_nodes_new','bou','Ln_n
ew','Lm_new','DEF','CI','fiber_diameter','ConnIndex','range','patchste
pMAX','fiber_diameter')
% save('timevaluation','counttime');
clear
load('data_step2')
% load('timevaluation')
cwd = pwd;
cd(tempdir);
pack
cd(cwd)
%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

%diam is a visualization variable only
% diam=(2/3)*fiber_diameter;
% teta = [0:pi/180:(2*pi-(pi/180))];

%%%
save('data_step3','fiber_diameter','Nnodes_new','n_nodes_new','m_nodes
_new','bou','Ln_new','Lm_new','DEF','CI','ConnIndex','range','patchste
pMAX','fiber_diameter')
% save('timevaluation','counttime');
clear
load('data_step3')
```

```
% load('timevaluation')
cwd = pwd;
cd(tempdir);
pack
cd(cwd)
%%%

%%%%%%%%%%%%%%%%%%%%%%%% STEP 2 DELANAUY NETWORK GENERATION %%%%%
% friendship matrix creation
TRI = delaunay(n_nodes_new,m_nodes_new);
friends_new=zeros(Nnodes_new,Nnodes_new,'int8');

%              counttime=counttime+1;
%              toc;
%              timeperf(counttime)=toc;
%              linetemp(counttime)=104;

% k triangles, V1, V2, V3 are the vertex
Ltri=length(TRI); % number of triangle elements
V1=TRI(1:Ltri,1);
V2=TRI(1:Ltri,2);
V3=TRI(1:Ltri,3);
clear TRI
for k=1:Ltri
    friends_new(V1(k),V2(k))=1;
    friends_new(V2(k),V1(k))=1;
    friends_new(V1(k),V3(k))=1;
    friends_new(V3(k),V1(k))=1;
    friends_new(V2(k),V3(k))=1;
    friends_new(V3(k),V2(k))=1;
end
clear V1 V2 V3
friends_new2=friends_new;
clear friends_new
for i =1:Nnodes_new
    for j=1:Nnodes_new
       if i>j
           friends_new2(i,j)=0;
       end
    end
end
% save the original connectivity
% save('or_connect','friends_new')

% original connectivity from the delanuay definition
% friends_new   full matrix
% friends_new2  half matrix


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
clear DIST
```

```
% additional connectivity reduction
%(proximity criterion, points that are too far from each other will be
disconnected DIST(i,j)>=(6/2)*(patchstepMAX)
%(proximity criterion, points that are too close to each other will be
disconnected  (DIST(i,j)<5*(fiber_diameter))

DIST=zeros(Nnodes_new,Nnodes_new);
  for i=1:Nnodes_new
        for j=1:Nnodes_new
            box = ((m_nodes_new(i)-
m_nodes_new(j))^2)+((n_nodes_new(i)-n_nodes_new(j))^2);
            DIST(i,j)=sqrt(box);
        end
  end
  for i=1:Nnodes_new
        for j=1:Nnodes_new
            % originally >=3 and <1
            if
(DIST(i,j)>=(3)*(patchstepMAX))||(DIST(i,j)<2*(fiber_diameter))

%                 friends_new(i,j)=0;
                friends_new2(i,j)=0;
            end
        end
  end

  clear DIST

% connectivity reduced by the proximity criterion
% friends_new   full matrix
% friends_new2  half matrix


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=176;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% Connectivity control algorithm the ConnIndex % of connections will
be
% eliminated
% flag=0;
% friends_new2 must be non simmetric, friends_new is simmetric
% clear friends_new
number_connect=sum(sum(friends_new2));
```

```
number_reductions=round(ConnIndex*number_connect);
% killer_space=number_connect;
%number and position of connection to erase, randomly generate with
DUD
% operates onle if necessary (ConnIndex>0)
if ConnIndex>0
%    next 3 code lines replace the following with the benefit of not
generating the same number twice
%    killer = unidrnd(killer_space,1,number_reductions);

    killer=1:number_reductions;
    y = randsample(killer,number_reductions);
    killer=y;
    killer = round(killer);
    nu_killer=length(killer);
    killed=0;
    count=0;
    for i=1:Nnodes_new
        for j=1:Nnodes_new
          if friends_new2(i,j)==1
            flag=0;
            count=count+1;
            for k=1:nu_killer
                if (count==killer(k))&&(flag==0)
                    flag=1;  % killer is a vector with multiple equal
values this flag avoid multiple
                             % use of the logical condition
(count==killer(k))
                    friends_new2(i,j)=0;
%                     friends_new(i,j)=0;
                    killed=killed+1; %variable to verify the precise
number of connections eliminated
                end

            end

          end

        end
    end
%     friends_new=friends_new2+friends_new2';

    clear count
end
 friends_new=friends_new2+friends_new2';
% connectivity reduced by the proximity criterion + the ConnIndex
% friends_new   full matrix
% friends_new2  half matrix
%               counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=233;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% ORIGINAL ANGLE DISTRIBUTION DUE TO DELANAUY
Xcenter=zeros(1,Nnodes_new);
Ycenter=zeros(1,Nnodes_new);
Xcenter_friend=zeros(1,Nnodes_new);
Ycenter_friend=zeros(1,Nnodes_new);
friends_angle=zeros(Nnodes_new,Nnodes_new);

% friends_new is full at this point

% delanauy network definition (later on after step 2 adaptation)
  for i=1:Nnodes_new;
    Xcenter(i)=n_nodes_new(i)+bou;
    Ycenter(i)=m_nodes_new(i)+bou;
        for j =1:Nnodes_new
            if friends_new(i,j)==1
               Xcenter_friend(j)=n_nodes_new(j)+bou;
               Ycenter_friend(j)=m_nodes_new(j)+bou;
               friends_angle(i,j)=atan((Ycenter_friend(j)-
Ycenter(i))/(Xcenter_friend(j)-Xcenter(i)));
               %  friends_angle(i,j) in radiant multiply by
(360/(2*pi))*
               %  to have it n degree
               if friends_angle(i,j)<0
                   friends_angle(i,j)=friends_angle(i,j)+2*pi;
               end
            end
        end
  end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
  % Eliminate boundaries


friends_angle=(360/(2*pi))*friends_angle;
% friends_angle full in degrees from 0 to 360
Fangle=zeros(Nnodes_new,Nnodes_new);
LogicA=friends_angle>180;
for i = 1:Nnodes_new
    for j=1:Nnodes_new
        if i<j&&friends_new2(i,j)==1
            Fangle(i,j)=friends_angle(i,j);
            if LogicA(i,j)==1
                Fangle(i,j)=friends_angle(i,j)-360;
            end
```

```
        end

    end
end
save('friends_angle_temp','friends_angle')
clear friends_angle LogicA
BFangle=zeros(Nnodes_new,Nnodes_new);
for i = 1:Nnodes_new
    for j=1:Nnodes_new
        if i<j
            if Fangle(i,j)>=0
            BFangle(i,j)=Fangle(i,j);
            elseif Fangle(i,j)<0
            BFangle(i,j)=Fangle(i,j)+180;
            end
        end
    end
end


%              counttime=counttime+1;
%              toc;
%              timeperf(counttime)=toc;
%              linetemp(counttime)=302;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%
                % starts the routine to eliminate the elements on the
subareas boundaries that are
                % at 0 90 180 360 degrees +- 30 degree (the area frame
is an unnecessary artifact )
                % BFangle is the value to use, half matrix from 0 to
180 degtrees
                %     logicount=1;
                % 300 pixels thick
                if DEF==0
                   spepcut=100;
                elseif DEF>0
                    spepcut=300;
                end

                    for i=1:Nnodes_new;
                    Xcenter(i)=n_nodes_new(i)+bou;
                    Ycenter(i)=m_nodes_new(i)+bou;
                        for j =1:Nnodes_new
                               Xcenter_friend(j)=n_nodes_new(j)+bou;
                               Ycenter_friend(j)=m_nodes_new(j)+bou;
                               if
((((Xcenter(i)>bou)&&(Xcenter(i)<spepcut+bou))||((Xcenter(i)>bou+Ln_ne
w-
spepcut)&&(Xcenter(i)<bou+Ln_new)))&&(((Xcenter_friend(j)>bou)&&(Xcent
er_friend(j)<spepcut+bou))||((Xcenter_friend(i)>bou+Ln_new-
spepcut)&&(Xcenter_friend(i)<bou+Ln_new))))
```

```
                                  if
((BFangle(i,j)>=80)&&(BFangle(i,j)<=100))
                                      friends_new(i,j)=0;
                                      friends_new2(i,j)=0;
                                  end
                              end
                              if
((((Ycenter(i)>bou)&&(Ycenter(i)<spepcut+bou))||((Ycenter(i)>bou+Lm_ne
w-
spepcut)&&(Ycenter(i)<bou+Lm_new)))&&(((Ycenter_friend(j)>bou)&&(Ycent
er_friend(j)<spepcut+bou))||((Ycenter_friend(j)>bou+Lm_new-
spepcut)&&(Ycenter_friend(j)<bou+Lm_new))))
                                  if
((BFangle(i,j)>=0)&&(BFangle(i,j)<=10))||((BFangle(i,j)>=170)&&(BFangl
e(i,j)<=180))
                                      friends_new(i,j)=0;
                                      friends_new2(i,j)=0;
                                  end
                              end
                       end
                   end
                % save('Lo','BFangle')
                % ends the frame cutting routine


                % connectivity reduced:
                % by the proximity criterion +
                % the ConnIndex +
                % frame cutting routine
                % friends_new   full matrix
                % friends_new2  half matrix




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


                for i =1:Nnodes_new
                    for j=1:Nnodes_new
                       if i>j
                           friends_new(i,j)=0;
                       end
                    end
                end

                % connectivity reduced:
                % by the proximity criterion +
                % the ConnIndex +
                % frame cutting routine
                % friends_new   half matrix
```

```
                % friends_new2  half matrix


                % friends_new   half matrix
                save('or_connect','friends_new')
                clear BFangle

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
                load('friends_angle_temp')

save('data_step4','Nnodes_new','n_nodes_new','m_nodes_new','bou','Ln_n
ew','Lm_new','DEF','CI','friends_new2','friends_new','friends_angle','
range','patchstepMAX','fiber_diameter')
%
save('timevaluation','timeperf','linetemp','counttime');
                clear
%                 load('timevaluation')
                load('data_step4')
                cwd = pwd;
                cd(tempdir);
                pack
                cd(cwd)

%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=395;
%



                  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% orientation
extration start
                DIST=zeros(Nnodes_new,Nnodes_new);
                  for i=1:Nnodes_new
                        for j=1:Nnodes_new
                            box = ((m_nodes_new(i)-
m_nodes_new(j))^2)+((n_nodes_new(i)-n_nodes_new(j))^2);
                            DIST(i,j)=sqrt(box);
                        end
                  end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
%
%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=414;
%
```

```
                C=zeros(Nnodes_new,Nnodes_new);
                for i = 1:Nnodes_new
                    for j=1:Nnodes_new
                        if i<j
                            if friends_new2(i,j)==1
                               C(i,j)=DIST(i,j);
                            end
                        end
                    end
                end

save('data_step5','Nnodes_new','n_nodes_new','m_nodes_new','bou','Ln_n
ew','Lm_new','DEF','CI','C','friends_new2','friends_new','range','patc
hstepMAX','fiber_diameter')
%
save('timevaluation','timeperf','linetemp','counttime');
                clear
                load('data_step5')
%                 load('timevaluation')
                cwd = pwd;
                cd(tempdir);
                pack
                cd(cwd)

%
%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=415;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
 % NEW SCAFFOLD CREATION FIBER ALIGNMENT CONTROL
 % preferential direction teta=90;
 % later on interactively defined

 add_n_nodes_new=n_nodes_new+bou;
 add_m_nodes_new=m_nodes_new+bou;

% X COMPRESSION
% 7 subareas every 1500 pixels
 xsubareas=Ln_new*(15/1500);
 stepx=round(Ln_new/xsubareas);
  % dispacement of 70% subareas x length
 deg_alx=round(DEF*(stepx/2));
 % move nodes over the x direction, n controls the columns thus x
 for startx=bou:stepx:Ln_new+bou;
     for i=1:Nnodes_new
         if
(add_n_nodes_new(i)<=(startx+stepx/2))&&(add_n_nodes_new(i)>(startx))
```

```
            add_n_nodes_new(i)=add_n_nodes_new(i)+deg_alx;
        elseif
(add_n_nodes_new(i)<=(startx+stepx))&&(n_nodes_new(i)>(startx+stepx/2)
)
            add_n_nodes_new(i)=add_n_nodes_new(i)-deg_alx;
        end
    end
 end

% Y EXTENSION
% 7 subareas every 1000 pixels
 ysubareas=Lm_new*(15/1500);
 stepy=round(Lm_new/ysubareas);
  % dispacement of DEF% subareas x length (originally was 70%  0.7)
 deg_aly=round(DEF*(stepy/2));
   % move nodes over the y direction, m controls the columns thus y
  for starty=bou:stepy:Lm_new+bou;
     for i=1:Nnodes_new
         if
(add_m_nodes_new(i)<=(starty+stepy/2))&&(add_m_nodes_new(i)>(starty))
            add_m_nodes_new(i)=add_m_nodes_new(i)-deg_aly;
        elseif
(add_m_nodes_new(i)<=(starty+stepy))&&(m_nodes_new(i)>(starty+stepy/2)
)
            add_m_nodes_new(i)=add_m_nodes_new(i)+deg_aly;
        end
    end
  end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
clear DIST
% **********************SECOND LINE PROXIMITY
CRITERION*************************
%(proximity criterion, points that are too far from each other will be
disconnected DIST(i,j)>=(6/2)*(patchstepMAX)
%(proximity criterion, points that are too close to each other will be
disconnected  (DIST(i,j)<5*(fiber_diameter))

DIST=zeros(Nnodes_new,Nnodes_new);
  for i=1:Nnodes_new
        for j=1:Nnodes_new
            box = ((m_nodes_new(i)-
m_nodes_new(j))^2)+((n_nodes_new(i)-n_nodes_new(j))^2);
            DIST(i,j)=sqrt(box);
        end
  end
  for i=1:Nnodes_new
        for j=1:Nnodes_new
            % originally >=3 and <1
            if
(DIST(i,j)>=(3)*(patchstepMAX))||(DIST(i,j)<2*(fiber_diameter))
```

```
                friends_new(i,j)=0;
                friends_new2(i,j)=0;
            end
        end
  end

% connectivity reduced by the proximity criterion
% friends_new   full matrix
% friends_new2  half matrix


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%


 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %new scaffold nodes
n_nodes_new= round(add_n_nodes_new)-bou;
m_nodes_new= round(add_m_nodes_new)-bou;
% the new nodes position produces a newer angle distribution
Xcenter=zeros(1,Nnodes_new);
Ycenter=zeros(1,Nnodes_new);
Xcenter_friend=zeros(1,Nnodes_new);
Ycenter_friend=zeros(1,Nnodes_new);
friends_angle=zeros(Nnodes_new,Nnodes_new);
  for i=1:Nnodes_new;
    Xcenter(i)=n_nodes_new(i)+bou;
    Ycenter(i)=m_nodes_new(i)+bou;
        for j =1:Nnodes_new
             if friends_new(i,j)==1
               Xcenter_friend(j)=n_nodes_new(j)+bou;
               Ycenter_friend(j)=m_nodes_new(j)+bou;
               friends_angle(i,j)=atan((Ycenter_friend(j)-
Ycenter(i))/(Xcenter_friend(j)-Xcenter(i)));
               %  friends_angle(i,j) now in radiants multiply by
(360/(2*pi))*
               %  to have it in degree
               if friends_angle(i,j)<0
                   friends_angle(i,j)=friends_angle(i,j)+2*pi;
               end
             end
        end
  end

save('data_step6','Nnodes_new','n_nodes_new','m_nodes_new','bou','DEF'
,'CI','C','friends_new2','friends_new','friends_angle','range','patchs
tepMAX')
% save('timevaluation','timeperf','linetemp','counttime');
clear
load('data_step6')
```

```
% load('timevaluation')
cwd = pwd;
cd(tempdir);
pack
cd(cwd)

%
%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=499;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%

  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% geometry estraction starts for
the
  % new scaffold with the new alignment
    DIST=zeros(Nnodes_new,Nnodes_new);
    for i=1:Nnodes_new
        for j=1:Nnodes_new
            box = ((m_nodes_new(i)-
m_nodes_new(j))^2)+((n_nodes_new(i)-n_nodes_new(j))^2);
            DIST(i,j)=sqrt(box);
        end
    end

B2=zeros(Nnodes_new,Nnodes_new);
good_friends=zeros(Nnodes_new,Nnodes_new,'int8');
C2=B2;

% friends_angle are in radiant and are now transformated in degrees
friends_angle=(360/(2*pi))*friends_angle;
Fangle=zeros(Nnodes_new,Nnodes_new);
LogicA=friends_angle>180;

for i = 1:Nnodes_new
    for j=1:Nnodes_new
        if i<j&&friends_new2(i,j)==1
            Fangle(i,j)=friends_angle(i,j);
            if LogicA(i,j)==1
                Fangle(i,j)=friends_angle(i,j)-360;
            end
            good_friends(i,j)=friends_new2(i,j);
        end
        if i>j
            good_friends(i,j)=friends_new2(j,i);
        end
    end
end
```

```
% connectivity reduced:
% 1)by the proximity criterion +
% 2)the ConnIndex +
% 3)frame cutting routine
% friends_new   half matrix
% friends_new2  half matrix
% good friends  full matrix = friends_new2

% friends_angle half from 0 to 360
% Fangle -180 to 180

for i = 1:Nnodes_new
    for j=1:Nnodes_new
        if i<j
            if Fangle(i,j)>=0
            B2(i,j)=Fangle(i,j);
            elseif Fangle(i,j)<0
            B2(i,j)=Fangle(i,j)+180;
            end
            if friends_new2(i,j)==1
               C2(i,j)=DIST(i,j);
            end
        end
    end
end

% B2 0 to 180 half ready for the estraction
% C2 distance half only where connection are present

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% cut connection on the XD, variable used:
%%%%%%%  +- pi adaptation  B2, C2, friends_new2, C distance matrix in
the
%%%%%%%  original scaffold
clear teta
teta = 90;
perp=90;
% range is now function input variable ; % xpreferred =- range
%%% eliminate def greater than the CI% in the interval 180 +- range, 0
%%% +-range

for i=1:Nnodes_new
    for j=1:Nnodes_new
        if friends_new(i,j)==1

            % first range @@@@@@

            if (B2(i,j)<((teta+perp)))&&(B2(i,j)>((teta+perp-range)))
                % eliminate abs for a flat distr
                lo1=abs((C2(i,j)-C(i,j)));
                lo2=(C(i,j)*CI);
```

```
                if  lo1>lo2
                    friends_new2(i,j)=0;
                    B2(i,j)=0;
                    C2(i,j)=0;
                end
            end %if
            if (B2(i,j)<((teta-perp)+range))&&(B2(i,j)>((teta-perp)))
                 % eliminate abs for a flat distr
                lo1=abs((C2(i,j)-C(i,j)));
                lo2=(C(i,j)*CI);
                if  lo1>lo2
                    friends_new2(i,j)=0;
                    B2(i,j)=0;
                    C2(i,j)=0;
                end
            end %if

            % second range @@@@@@
            if (B2(i,j)<((teta+perp)-range))&&(B2(i,j)>((teta+perp-
2*range)))
                % eliminate abs for a flat distr
                lo1=abs((C2(i,j)-C(i,j)));
                lo2=(C(i,j)*(CI+0.3));
                if  lo1>lo2
                    friends_new2(i,j)=0;
                    B2(i,j)=0;
                    C2(i,j)=0;
                end
            end %if
            if (B2(i,j)<((teta-perp)+2*range))&&(B2(i,j)>((teta-
perp+range)))
                 % eliminate abs for a flat distr
                lo1=abs((C2(i,j)-C(i,j)));
                lo2=(C(i,j)*(CI+0.3));
                if  lo1>lo2
                    friends_new2(i,j)=0;
                    B2(i,j)=0;
                    C2(i,j)=0;
                end
            end %if

        end % if
    end %j
end %i


% ATTENTION friends_new2 has now the correct updated values after
cutting
% the connection over the XD, friends_new represetns the old scaffold
% B2=0 if no connections B2 angle in degrees half
% C2=0 if no connections C2 distance matrix half
% or_connect_new --> correct connectivity

% friends_new=friends_new2;
% good_friends=friends_new2+friends_new2';
```

```
% connectivity reduced:
% 1)by the proximity criterion +
% 2)the ConnIndex +
% 3)frame cutting routine
% 4) artifact reduction over the XD


% friends_new   half matrix
% friends_new2  half matrix
% good friends full matrix = friends_new2



% friends_new2  half matrix
save('or_connect_new','friends_new2');

%
%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=663;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% Anew3 for visualization
% save('A_data3','Anew3','Mcounter');
load('A_data3');

% clear('friends_new2')
% load('or_connect_new')
Xcenter=zeros(1,Nnodes_new);
Ycenter=zeros(1,Nnodes_new);
Xcenter_friend=zeros(1,Nnodes_new);
Ycenter_friend=zeros(1,Nnodes_new);
friends_angle=zeros(Nnodes_new,Nnodes_new);
%  % modified scaffolds last
%  teta = 0:pi/180:(2*pi-(pi/180));
  for i=1:Nnodes_new;
    Xcenter(i)=n_nodes_new(i)+bou;
    Ycenter(i)=m_nodes_new(i)+bou;
        for j =1:Nnodes_new
             if (friends_new2(i,j)==1)
                 %&&(sum(friends_new2(i,:)>1)
               Xcenter_friend(j)=n_nodes_new(j)+bou;
               Ycenter_friend(j)=m_nodes_new(j)+bou;
               friends_angle(i,j)=atan((Ycenter_friend(j)-
Ycenter(i))/(Xcenter_friend(j)-Xcenter(i)));
               %  friends_angle(i,j) in radiant multiply by
(360/(2*pi))*
               %  to have it n degree
               if friends_angle(i,j)<0
                   friends_angle(i,j)=friends_angle(i,j)+2*pi;
               end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % %                 % main difference with network_generator_full_H
% % %                 if Xcenter(i)<=Xcenter_friend(j)
% % %                   xf= Xcenter(i):0.01:Xcenter_friend(j);
% % %                 elseif Xcenter(i)>Xcenter_friend(j)
% % %                   xf= Xcenter_friend(j):0.01:Xcenter(i);
% % %                 end
% % %
% % %                 if length(xf)>1
% % %                     yf= (xf-
Xcenter(i))*(tan(friends_angle(i,j)))+Ycenter(i);
% % %                 elseif length(xf)<=1
% % %                        if Ycenter(i)<=Ycenter_friend(j)
% % %                            yf=Ycenter(i):0.01:Ycenter_friend(j);
% % %                        elseif Ycenter(i)>Ycenter_friend(j)
% % %                            yf=Ycenter(i):0.01:Ycenter_friend(j);
% % %                        end
% % %                     if length(yf)>1
% % %                     xf=Xcenter(i)*ones(1,length(yf));
% % %                     elseif length(yf)<=1
% % %                     yf=Ycenter(i);
% % %                     xf=Xcenter(i)*ones(1,length(yf));
% % %                     end
% % %                 end
% % %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % %                 Xr1 = round(xf);
% % %                 Yr1 = round(yf);
% % %                 N=length(Xr1);
% % %                 for k=1:N
% % %                     Anew3(Yr1(k),Xr1(k))=255;
% % %                 end %end k
            end % end friends if
        end % end j
  end % end i

                friends_new2=friends_new2+friends_new2';


%                             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=735;
%
            % % startreactivate islands removal

            % ISOLATED ISLANDS REMOVAL MODULE

            % STEP 1) ADJACENCY LIST CREATION


            [aadj, badj]=size(friends_new2);
```

```
            rootnode=zeros(aadj,1);
            adjacency=zeros(aadj,badj);
            for i=1:aadj
                countadj=1;
                rootnode(i,1)=sum(friends_new2(i,:));
                for j=1:badj
                    adjacency(i,1)=i;
                    if friends_new2(i,j)>0
                       countadj=countadj+1;
                       adjacency(i,countadj)=j;
                    end
                end
            end

            % STEP 2) INIZIALING FLAGS ARRAY

            [rootnodevalue,K]=max(rootnode);

            flagislandkiller=zeros(aadj,1);

            % starting node in the network, the most connected
            % K is the most connected
            flagislandkiller(K)=1;



            % STEP 3) FIND NEIGHBORS OF K

            [aADJA bADJA]=size(adjacency);
            Kneighbors=adjacency(K,2:bADJA);
            Kneighbors=Kneighbors(Kneighbors>0);
            flagislandkiller(Kneighbors)=1;



            % store all of the connected intersections in
queeconnected
            % K and neighbors are now on the quee


            queeconnected=Kneighbors;


            while ~isempty(queeconnected)

                % if the quee has still nodes retrieve the next call
it K and move to
                % step: find the neighbors

                      K=queeconnected(1);


queeconnected_new=queeconnected(2:length(queeconnected));
```

```
                        clear queeconnected
                        queeconnected=queeconnected_new;
                        clear queeconnected_new
                        Kneighbors=adjacency(K,2:bADJA);


                        Kneighbors=Kneighbors(Kneighbors>0);

Kneighbors=(~flagislandkiller(Kneighbors))'.*Kneighbors;
                        Kneighbors=Kneighbors(Kneighbors>0);

                        flagislandkiller(Kneighbors)=1;
            %             disp(flagislandkiller);
                        startquee=length(queeconnected);

queeconnected(startquee+1:startquee+length(Kneighbors))=Kneighbors;
            %             disp(queeconnected)



            end


            % flagislandkiller is teh main module outcome
            % it is a vector of 1 and 0
            % 1 for the intersections associated the main network
            % 0 for the intersections associated the isolated islands


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%
%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=828;



            % now update the network
            n_nodes_new=n_nodes_new+100;
            n_nodes_new=flagislandkiller'.*n_nodes_new;
            n_nodes_new=n_nodes_new(n_nodes_new>0);
            m_nodes_new=m_nodes_new+100;
            m_nodes_new=flagislandkiller'.*m_nodes_new;
            m_nodes_new=m_nodes_new(m_nodes_new>0);

            n_nodes_new=n_nodes_new-100;
```

```
            m_nodes_new=m_nodes_new-100;


islandkiller=flagislandkiller'.*(1:length(flagislandkiller));
            islandkiller=islandkiller(islandkiller>0);

            friends_new2new=friends_new2(islandkiller,islandkiller);
%
friends_anglenew=friends_angle(islandkiller,islandkiller);

            Nnodes_new=length(islandkiller);
            save('anglelastsaving','friends_angle');
            clear friends_new2 DIST Fangle friends_angle

            %Job done all isoleted nodes and islands have been killed
now store new data

            friends_new2=friends_new2new;
%             friends_angle=friends_anglenew;



            % % endreactivate islands removal


% % %               xcir=zeros(1,Nnodes_new);
% % %               ycir=zeros(1,Nnodes_new);
% % %             %   friends_new2=[friends_new2+friends_new2'];
% % %               for i =1:Nnodes_new
% % %                     % USE PREVIOUS COORDINATES FOR NODES
STATISTICAL DISTRIBUTION EVALUATION
% % %                     %traslating point distribution of bou for
visualisation purposes
% % %                     %real area is Ln_new X Lm_new, real
coordinates will be Xinit-bou,
% % %                     %Yinit-bou
% % %             %         if(sum(friends_new2(i,:))>0)
% % %                               Xinit(i)=[n_nodes_new(1,i)+bou];
% increasing order from the left to the right side of the image
% % %                               Yinit(i)=[m_nodes_new(1,i)+bou];
% increasing order from the top to the bottom of the image
% % %                               Anew3((Yinit(i)-
(15)):(Yinit(i)+(15)),Xinit(i))=255;
% % %                               Anew3(Yinit(i),(Xinit(i)-
(15)):(Xinit(i)+(15)))=255;
% % %                               xcir = Xinit(i) + (diam)*cos(teta);
% % %                               ycir = Yinit(i) + (diam)*sin (teta);
% % %                               Xr = round(xcir);
% % %                               Yr = round(ycir);
% % %                               N=length(Xr);
% % %                               for j=1:N
% % %                                  Anew3(Yr(j),Xr(j))=255;
% % %                               end
% % %             %         end % end if
% % %               end % end for
```

```
% % %              %
% % %              %   figure(10)
% % %              %   imshow(Anew3)



            save('or_connect_new','friends_new2')

            %   toc
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%





%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% Visualisation data

% value2 are the non-zero values in the angle matrix
[pos_raw pos_column value2]=find(B2);
value2=round(value2);
% valueC are the non-zero values in the distance matrix
% [pos_rawC pos_columnC valueC]=find(C2);
% valueC=round(valueC);

% weights for the weighted fiber alignment
% must be estracted where the non-zero values for the fiber angle
% are (pos_raw(i),pos_column(i)) instead of
(pos_rawC(i),pos_columnC(i))

f=length(pos_raw);
weight=zeros(1,f);
for i=1:f
    weight(i)=C2(pos_raw(i),pos_column(i));
end
%              counttime=counttime+1;
%              toc;
%              timeperf(counttime)=toc;
%              linetemp(counttime)=926;

g=length(value2);
% START=0;
% STOP=180;
```

```
% STEP=5;
interval3=(0:5:180);
k=length(interval3);
counter3=zeros(k,1);
counter2=zeros(k,1);
for i=1:(g)
    for j=1:(k-1)
        if (value2(i)<=interval3(j+1))&&(value2(i)>interval3(j))
            counter3(j)=counter3(j)+1;
            counter2(j)=weight(i)+ counter2(j);
            %counter2(j)=counter(j)*weight(i);

        end
    end
end

% counter3= number of fibers at the specific angle
% counter2= fiber length at the specific angle

counter_weighted3=counter3.*counter2;
% counter_weighted_norm3=counter_weighted3/length(value2);

% clear the big matrixes to reduce workspace memory, matrixes no
longer needed
clear('B2','C','C2')

x_hist2=(0:5:180);
[n2,xout2]=hist(value2,x_hist2);
% n12=n2/length(value2);
% ylimit=max(n12);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%     n2TOTinterval=0:5:180;
%     ylimit=max(n2)+0.01;
%     figure(16)
%     bar(n2TOTinterval,n2,'k')
%     axis([0 180 0 ylimit])
%     title ('MODEL normalized fiber count VS angle')
%     xlabel('angle')
%     ylabel('normalized fiber count')



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%


%CONNECTIVITY
```

```
% new scaffold network

% working on the correct connectivity


clear friends_new2
load('or_connect_new')
% friends_new2=[friends_new2+friends_new2'];

% now a full matrix
% Connections_new=sum(sum(friends_new2))/2;
% total number of connection in the new matrix
% they corrispond to the number of bar elements
connectivity_new=zeros(1,Nnodes_new);
for f=1:Nnodes_new
    connectivity_new(f)=sum(friends_new2(f,1:Nnodes_new));
end
% connectivity_new is a vector prepresenting number of connection per
nodes
% it has to work with the full matrix to correctlyt represents the
nodes
% connectivity

% min_con=min(connectivity_new);
% max_con=max(connectivity_new);

min_con=0;
max_con=50;
% to make sure to compare the same vectos in different networks

conn_interval_new=(min_con:max_con);
k=1;
for conn=min_con:max_con
    [conn_vect]=find(connectivity_new==conn);
     Conn_count_new(k)=length(conn_vect);
     k=k+1;
end
% Conn_count:how many nodes with that specific connectivity

%normalized connectivity count new network
% Conn_count_norm_new=Conn_count_new/(Connections_new*2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% FEM model data production on Anew3 final scaffold
load('data_step1')
% Nnodes_new=length(n_nodes_new);
nodes_identifier=(1+Nodes_counter):(Nnodes_new+Nodes_counter);
nodes_identifier=nodes_identifier';
nodes(:,1)=nodes_identifier;                              % node
identifier
```

```
nodes(:,2)=(n_nodes_new + Xoffset)';                    % node X
coordinates
nodes(:,3)=(m_nodes_new + Yoffset)';                    % node Y
coordinates

Nodes_counter_done=max(nodes_identifier);

% figure(7) % visual check
% axis equal
% plot(nodes(:,2),-nodes(:,3),'xr')
clear friends_new2
load('or_connect_new')

% make sure at this line that friends_new2 is = 0 when i<j
% otherwise the bar elements will be counted twice


[a b]=size(friends_new2);
good_friends=zeros(a,b);
for i=1:a
    for j=1:b
        if i>j
            good_friends(i,j)=friends_new2(i,j);
        end
    end
end


% good_friends=friends_new2;
[a b]=size(good_friends);
counterE=0;
% counterK=0;
for i = 1:a
    [E1]=find(good_friends(i,:));
    LE1=length(E1);
       for k=1:LE1
           E=[i,E1(k)];
           counterE=counterE+1;
           Etot(counterE,1)=i+Nodes_counter;
           Etot(counterE,2)=E1(k)+Nodes_counter;
       end
end
clear good_friends
good_friends=friends_new2;
a1=gen_pos_m; % row
b1=gen_pos_n; % column
str1 = int2str(a1);
str2 = int2str(b1);
filename=('FEM_nodesdata_');
str3='_';
S1 = (strcat(filename,str1,str3,str2));
save(S1,'nodes','-ascii');

filename='FEM_elementsdata_';
S2 = (strcat(filename,str1,str3,str2));
```

```
save(S2,'Etot','-ascii');

load('anglelastsaving');
filename='angledata_';
S3 = (strcat(filename,str1,str3,str2));
% good_friends is the final friendship matrix after the alignment
control
% and after the cutting of the fiber s over the XD direction
% friens angle is in radiant
save(S3,'friends_angle','good_friends');


% %%%%%%%%%%%%%%%%%%%%%%geometric
data%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% number of nodes                                          --->
'Nnodes_new'
% connectivity after the fiber alignment control    --->
'conn_interval_new','Conn_count_new
% fiber alignment weighted   --->
'interval3','counter_weighted_norm3'
% fiber alignment simple     --->
'xout2','n2','value2'

filename='geodata_';
S4 = (strcat(filename,str1,str3,str2));
save(S4,'Nnodes_new','conn_interval_new','Conn_count_new','interval3',
'counter_weighted3','xout2','n2','value2');


%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=1066;
%             FX = diff(timeperf);
%             figure(7)
%             grid on
%             plot(linetemp,timeperf,'xb');hold on
%             plot(linetemp,timeperf,'b');
% %           plot(linetemp(2:length(linetemp)),FX,'xr');
% %           plot(linetemp(2:length(linetemp)),FX,'r');hold off
%             plot(linetemp(1:length(linetemp)-1),FX,'xr');
%             plot(linetemp(1:length(linetemp)-1),FX,'r');hold off
%
%             title ('algorithm performance evaluation, code vs time
and its gradient')
%             xlabel('code line')
%             ylabel('time [seconds]')


% clear
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            % debug completed April 27 2009
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX C)

REAL – ARTIFICAL NETWORK ERROR MINIMIZER CODE (MATLAB)

```
% MICROARCHITECTURE DATA ON THE DESIGN SPACE
% Error mapping script  Wtot=W1+W2+W3



clear
close all
clc
load('FEMmodeldata.mat')
% they can be transformed in questions, example
% magn = input('please provide magnification adopted in the original
image source  ');

magn=3500;
fiber_diameter=DTOTM(2);
node_density=M(2);
Element_size=3000;
TotalAreaModel=81000000;
tic
counter=0;
for DEF=0.25:0.02:0.35;
    for ConnIndex=0:0.02:0.1

toc


[Nodes_densityS,OIS,counter_weighted_normS,nS,Conn_count_normS,Diamete
rS,Real_areaS,model_sizeS]=
fem_generator_9(TotalAreaModel,Element_size,DEF,ConnIndex,node_density
,fiber_diameter,magn);


        counter=counter+1

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % primary data for error function evaluation
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


        % for overlaps density error
%         Er1(counter)=abs(Nodes_density_real-
Nodes_densityS)/Nodes_density_real;
%         Er1(counter)=abs(Nodes_density_real-Nodes_densityS);
```

```
        Er1(counter)=Nodes_densityS;


        % fiber alignment weighted
%         Er2mom=sum(abs(counter_weighted_normS-
counter_weighted_norm'));
%         Er2(counter)=Er2mom/sum(counter_weighted_norm');
%         Er2(counter)=sum(abs(counter_weighted_normS-
counter_weighted_norm'));

        Er2(counter,1:19)=counter_weighted_normS;

        % connectivity
%         Er3mom=sum(abs(Conn_count_normS-Conn_count_norm));
%         Er3(counter)=Er3mom/sum(Conn_count_norm);
%         Er3(counter)=sum(abs(Conn_count_normS-Conn_count_norm));

        Er3(counter,1:51)=Conn_count_normS;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % secondary data

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        Er4(counter,1:37)=nS;
        Er5(counter)=OIS;
        Er6(counter)=model_sizeS;
        Er7(counter)=Real_areaS;
        Er8(counter)=DiameterS;


        ErrorTOT(counter,2)=DEF;
        ErrorTOT(counter,1)=ConnIndex;
    end
end


save('Mesh_4_5_final_fine','Er1','Er2','Er3','Er4','Er5','Er6','Er7','
Er8','ErrorTOT')

clear
load('Mesh_4_5_final_fine')
```

```
% MICROARCHITECTURE DATA ON THE DESIGN SPACE
% Error mapping script  Wtot=W1+W2+W3



clear
close all
clc
load('FEMmodeldata.mat')
% they can be transformed in questions, example
% magn = input('please provide magnification adopted in the original
image source  ');

magn=3500;
fiber_diameter=DTOTM(2);
node_density=M(2);
Element_size=3000;
TotalAreaModel=81000000;
tic
counter=0;
for DEF=0.25:0.02:0.35;
    for ConnIndex=0:0.02:0.1

toc


[Nodes_densityS,OIS,counter_weighted_normS,nS,Conn_count_normS,Diamete
rS,Real_areaS,model_sizeS]=
fem_generator_9(TotalAreaModel,Element_size,DEF,ConnIndex,node_density
,fiber_diameter,magn);


        counter=counter+1

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % primary data for error function evaluation
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


        % for overlaps density error
%         Er1(counter)=abs(Nodes_density_real-
Nodes_densityS)/Nodes_density_real;
%         Er1(counter)=abs(Nodes_density_real-Nodes_densityS);

        Er1(counter)=Nodes_densityS;


        % fiber alignment weighted
%         Er2mom=sum(abs(counter_weighted_normS-
counter_weighted_norm'));
%         Er2(counter)=Er2mom/sum(counter_weighted_norm');
%         Er2(counter)=sum(abs(counter_weighted_normS-
counter_weighted_norm'));
```

```
        Er2(counter,1:19)=counter_weighted_normS;

        % connectivity
%         Er3mom=sum(abs(Conn_count_normS-Conn_count_norm));
%         Er3(counter)=Er3mom/sum(Conn_count_norm);
%         Er3(counter)=sum(abs(Conn_count_normS-Conn_count_norm));

        Er3(counter,1:51)=Conn_count_normS;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % secondary data

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        Er4(counter,1:37)=nS;
        Er5(counter)=OIS;
        Er6(counter)=model_sizeS;
        Er7(counter)=Real_areaS;
        Er8(counter)=DiameterS;


        ErrorTOT(counter,2)=DEF;
        ErrorTOT(counter,1)=ConnIndex;
    end
end


save('Mesh_4_5_final_fine','Er1','Er2','Er3','Er4','Er5','Er6','Er7','
Er8','ErrorTOT')

clear
load('Mesh_4_5_final_fine')
```

```
function
[Nodes_densityS,OIS,counter_weighted_normS,nS,Conn_count_normS,Diameter
rS,Real_areaS,model_sizeS]=
fem_generator_9(TotalAreaModel,Element_size,DEF,ConnIndex,node_density
,fiber_diameter,magn);

%
[Nodes_densityS,OIS,counter_weighted_normS,nS,Conn_count_normS,Diameter
rS,Real_areaS,model_sizeS]=
fem_generator_9(1000000,1000,0,0,0.46,0.32,3500);

%
[Nodes_densityS,OIS,counter_weighted_normS,nS,Conn_count_normS,Diameter
rS,Real_areaS,model_sizeS]=
fem_generator_9(9000000,3000,0,0,0.46,0.32,3500);

% tic

% % function inputs typical for isotropic scaffold 1.5

% magn=2500;
% node_density=0.1842;
% fiber_diameter=0.4790;
% DEF=0.59;
% ConnIndex=0;
% TotalAreaModel=36000000;
% Element_size=3000;


% function outputs
% %
% % Nodes_densityS------------OK
% % OIS-----------------------OK
% % counter_weighted_normS----OK
% % nS------------------------OK
% % Conn_count_normS----------OK
% % DiameterS-----------------OK
% % Real_areaS ---------------OK
% % model_sizeS --------------OK


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

%MODEL GENERATOR MAIN BODY

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% MODEL INPUTS
```

```
save('data_magn','magn')
% clear
% load('data_magn')

DiameterS=fiber_diameter;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
    %  NETWORK SIMULATION MAIN SETTINGS

     %input 1)
     % magn


     %input 2)

     % converted node_density from N/microns^2 into N/pixels^2

     node_density=node_density/(magn/100*magn/100);

     %input 3)

     % converted fiber diameter from microns into pixels length

     fiber_diameter =round(fiber_diameter*magn/100);

     % external borders size in pixels required for visualization
purposes only
     bou=500;

     % CONTROL PARAMETERS

     %input 4)
     % (I a) LEVEL OF FIBER ALIGNMENT DESIDERED
     %      DEF;
            % patches dimensions to feel holes due to subareas network
definition
            % originally patchstep 250 and patchstep/5 X ---
patchstep/3 Y

            patchstep=150;  % 150 is the minimum to get a random
configuration
            %within the patch
            %patch x direction is reduced when 90 is the main
direction of
```

```
            %alignment
            patchx=patchstep-DEF*patchstep/2;
            %patch y direction is icreased when 90 is the main
direction of
            %alignment
            patchy=patchstep+DEF*patchstep/2;

            patchstepMAX=max(patchx,patchy);

            % (I b)
            % correction index, deformation greater than 5% along the
cross preferred dir
            % will be erased   originally 5% (0.05) it compensates the
creation of
            % alignment in the XD
%             counttime=0;
%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=100;
%
            % set a constant value if required
%             CI=0.01;
            % function of DEF if the alignment goes up the CI goes
down
            CI=0.05*DEF/0.7;
            % the 0.7 constant was selected on a trial and error base
            % improve this choice if any time left

            % (I c)
            % correction angle for CI , originally was 25
            % (eliminate the fibers with a CI% stretch on the
direction perpendiular
            % to the main alignment +- an angle of range)
            %range=60;
            % function of DEF if the alignment goes up the range goes
up too
            range=30*DEF/0.7;   %20 and 0.7 was selected improve this
choice if any time left

     % LEVEL OF CONNECTIVITY DESIDERED
     % Connectivity control index from 0 (0% eliminated)to 1 (100%
eliminated)
     % fibers to be eliminated are randomly selected
     %(II)

     %input 5)
     % ConnIndex;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
    % MODEL AREA
    %(III)

        % MACRO target= 3 X 3 mm
```

```
        % for 9.0 magnification of 2500 this means 75000 x 75000
pixels
        % for 1.5 and 4.5 magnification of 3500 this means 95000 x
95000 pixels
        % MESO target= 0.5 X 0.5 mm
        % for 9.0 magnification of 2500 this means 12500 x 12500
pixels
        % for 1.5 and 4.5 magnification of 3500 this means 17500 x
17500 pixels

        % Total Model Area in pixels
        %  TotalAreaModel
        %  Element_size

        % ASSUMING TO GENERATE SQAURE ONLY Element_sizeX=Element_sizeY
        save('model_size','TotalAreaModel','Element_size','bou');
        %number of sub areas
        Nmatrix=TotalAreaModel/(Element_size*Element_size);
        % number of sub areas per side
        SNmatrix=sqrt(Nmatrix);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
        % Real area in square micrometers
        Area_new=Element_size*Element_size*Nmatrix;
        %magnification factor used
        magn_fact=magn/100;
        % real model area and size
%         disp('model area in micrometers^2 ')
        model_area=Area_new/(magn_fact^2);
%         disp(model_area)
        Real_areaS=model_area;

%         disp('model size in micrometers')
        model_size=(Element_size*SNmatrix)/magn_fact;
%         disp(model_size)
        model_sizeS=model_size;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%


% BOUNDARIES
Ln_new=Element_size;
Lm_new=Element_size;

% subareas definition for fiber alignment procedure
```

```
% X EXTENSION
% 15 subareas every 1500 pixels
xsubareas=Ln_new*(15/1500);
 stepx=round(Ln_new/xsubareas);
% dispacement of DEF% subareas x length (originally was 70%  0.7)
 deg_alx=round(DEF*(stepx/2));

% Y EXTENSION
% 7 subareas every 1500 pixels
 ysubareas=Lm_new*(15/1500);
 stepy=round(Lm_new/ysubareas);
  % dispacement of DEF% subareas x length (originally was 70%  0.7)
 deg_aly=round(DEF*(stepy/2));

 %creates a matrix to identify areas boundaries and neighbours
borders=zeros(SNmatrix+2,SNmatrix+2);
borders(2:SNmatrix+1,2:SNmatrix+1)=1;
% disp(borders(2:SNmatrix+1,2:SNmatrix+1));

fiber_diameter_pixels=fiber_diameter;
save('SIMULATIONDATA','magn','node_density','fiber_diameter_pixels','b
ou','DEF','CI','range','ConnIndex','TotalAreaModel','Element_size','bo
rders');

%borders for points at the borders identification
borders_conn_hor_right=borders;
borders_conn_hor_left=borders;
borders_conn_vert_up=borders;
borders_conn_vert_down=borders;

%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=203;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%STARTS SUB-AREAS ---- NETWORKS GENERATION 1/2
% sub areas generation
% total nodes counter
Nodes_counter=0;
% subareas counter
Mcounter=1;
for i=1:SNmatrix+2
    for j=1:SNmatrix+2
        %generates areas only if inside the borders matrix
        if borders(i,j)==1
%             i
%             j
            %areas offsets, offsets depend also on applied deformation
(deg_alx and deg_aly)
            Xoffset=(j-2)*Element_size - round((j-2)*deg_alx);
%             disp(Xoffset)
            Yoffset=(i-2)*Element_size + round((i-2)*deg_aly);
%             disp(Yoffset)
```

```
            [Nodes_counter_done] =
orientationadaptation_comp29_novis1(node_density,Xoffset,Yoffset,i,j,E
lement_size,Mcounter,Nodes_counter,DEF,fiber_diameter,CI,bou,ConnIndex
,range,patchstepMAX);
            % Nodes_counter counts the total number of nodes
            Nodes_counter=Nodes_counter_done;
            % Mcounter counts the total number of areas
            Mcounter=Mcounter+1;
        end
    end
end
%ENDS SUB-AREAS  -----   NETWORKS GENERATION 1/2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=239;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%STARTS SUB-AREAS CONNECTING  -----  NETWORKS GENERATION 2/2
new_conn_count=1;
conn_subareas_count=0;


for i=1:SNmatrix+2
    for j=1:SNmatrix+2
        % operates on original sub networks only
        if borders(i,j)==1
            % UP AND
DOWN%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            for k=i-1:i+1   % check up and down the analyzed element
(i,j)
                if k~=i     % to eliminate the element position i

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                    % DOWN
                    if
((k>i)&&(borders_conn_vert_down(i,j)==1)&&(borders(k,j)==1)) % looking
down with respect to subarea (i,j)
                        % load (i,j) data analyzed subarea
                        str1c = int2str(i);
                        str2c = int2str(j);
                        filename1=('FEM_nodesdata_');
                        filename2=('FEM_conn_nodesdata_');
                        str3c='_';
                        S1 = (strcat(filename1,str1c,str3c,str2c));
                         % load (k,j) data subarea below (i,j)
                        str1v = int2str(k);
                        str2v = int2str(j);
                        str3v='_';
```

```
                            S2 = (strcat(filename1,str1v,str3v,str2v));
                            Acenter=load (S1);
                            Avert=load(S2);
                            % subareas overlapping areas definition
                            conn_subareas_count=conn_subareas_count+1;
                            Acent2=max(Acenter(:,3));
                            Acent1=Acent2-patchy;

                            Avert2=min(Avert(:,3));
                            Avert1=Avert2+patchy;
                            [ac,bc]=size(Acenter);
                           % nodes of (i,j) at the boundary with (k,j)
                           for p=1:ac
                               if
(Acenter(p,3)<=Acent2)&&(Acenter(p,3)>=Acent1)

New_conn_net(new_conn_count,1)=Acenter(p,1);

New_conn_net(new_conn_count,2)=Acenter(p,2);

New_conn_net(new_conn_count,3)=Acenter(p,3);
                                    new_conn_count=new_conn_count+1;
                               end
                           end
                           [av,bv]=size(Avert);
                           % nodes of (k,j) at the boundary with (i,j)
                           for p=1:av
                               if
(Avert(p,3)<=Avert1)&&(Avert(p,3)>=Avert2)

New_conn_net(new_conn_count,1)=Avert(p,1);

New_conn_net(new_conn_count,2)=Avert(p,2);

New_conn_net(new_conn_count,3)=Avert(p,3);
                                    new_conn_count=new_conn_count+1;
                               end
                           end
                       id = int2str(conn_subareas_count);
                       Sstore = (strcat(filename2,str3c,id));
                       save(Sstore,'New_conn_net');     % save sub-
areas connecting zones points network creation
                       new_conn_count=1;
                       clear New_conn_net
                       borders_conn_vert_down(i,j)=0; % this avoid to
connect twice the same set
                       borders_conn_vert_up(k,j)=0;
                       end  % if k > i

                       % UP
                       if
((k<i)&&(borders_conn_vert_up(i,j)==1)&&(borders(k,j)==1))  % looking
up with respect to subarea (i,j)
                            str1c = int2str(i);
                            str2c = int2str(j);
                            filename1=('FEM_nodesdata_');
```

```
                                filename2=('FEM_conn_nodesdata_');
                                str3c='_';
                                S1 = (strcat(filename1,str1c,str3c,str2c));
                                str1v = int2str(k);
                                str2v = int2str(j);
                                str3v='_';
                                S2 = (strcat(filename1,str1v,str3v,str2v));
                                Acenter=load (S1);
                                Avert=load(S2);
                               conn_subareas_count=conn_subareas_count+1;
                               Acent2=min(Acenter(:,3));
                               Acent1=Acent2+patchy;
                               Avert2=max(Avert(:,3));
                               Avert1=Avert2-patchy;
                               [ac,bc]=size(Acenter);
                               % nodes of (i,j) at the boundary with (k,j)
                               for p=1:ac
                                   if
(Acenter(p,3)>=Acent2)&&(Acenter(p,3)<=Acent1)

New_conn_net(new_conn_count,1)=Acenter(p,1);

New_conn_net(new_conn_count,2)=Acenter(p,2);

New_conn_net(new_conn_count,3)=Acenter(p,3);
                                       new_conn_count=new_conn_count+1;
                                   end
                               end
                               [av,bv]=size(Avert);
                               % nodes of (k,j) at the boundary with (i,j)
                               for p=1:av
                                   if
(Avert(p,3)>=Avert1)&&(Avert(p,3)<=Avert2)

New_conn_net(new_conn_count,1)=Avert(p,1);

New_conn_net(new_conn_count,2)=Avert(p,2);

New_conn_net(new_conn_count,3)=Avert(p,3);
                                       new_conn_count=new_conn_count+1;
                                   end
                               end
                           id = int2str(conn_subareas_count);
                           Sstore = (strcat(filename2,str3c,id));
                           save(Sstore,'New_conn_net');     % save sub-
areas connecting zones points network creation
                           new_conn_count=1;
                           clear New_conn_net
                           borders_conn_vert_up(i,j)=0; % this avoid to
connect twice the same set
                           borders_conn_vert_down(k,j)=0;
                           end % if k < i
%                         end
                    end
                end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
            % LEFT AND 
RIGHT%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            for k=j-1:j+1   % check left and right the analyzed 
elelment
                if k~=j
                       %RIGHT
                       if 
((k>j)&&(borders_conn_hor_right(i,j)==1)&&(borders(i,k)==1))      % 
looking right with respect to subarea (i,j)
                           str1c = int2str(i);
                           str2c = int2str(j);
                           filename1=('FEM_nodesdata_');
                           filename2=('FEM_conn_nodesdata_');
                           str3c='_';
                           S1 = (strcat(filename1,str1c,str3c,str2c));
                           str1v = int2str(i);
                           str2v = int2str(k);
                           str3v='_';
                           S2 = (strcat(filename1,str1v,str3v,str2v));
                           Acenter=load(S1);
                           Ahor=load(S2);
                           conn_subareas_count=conn_subareas_count+1;
                          Acent2=max(Acenter(:,2));
                          Acent1=Acent2-patchx;
                          Ahor2=min(Ahor(:,2));
                          Ahor1=Ahor2+patchx;
                          [ac,bc]=size(Acenter);
                          % nodes of (i,j) at the boundary with (i,k)
                          for p=1:ac
                              if 
(Acenter(p,2)<=Acent2)&&(Acenter(p,2)>=Acent1)

New_conn_net(new_conn_count,1)=Acenter(p,1);

New_conn_net(new_conn_count,2)=Acenter(p,2);

New_conn_net(new_conn_count,3)=Acenter(p,3);
                                  new_conn_count=new_conn_count+1;
                              end
                          end
                          [av,bv]=size(Ahor);
                          % nodes of (i,k) at the boundary with (i,j)
                          for p=1:av
                              if 
(Ahor(p,2)<=Ahor1)&&(Ahor(p,2)>=Ahor2)

New_conn_net(new_conn_count,1)=Ahor(p,1);

New_conn_net(new_conn_count,2)=Ahor(p,2);

New_conn_net(new_conn_count,3)=Ahor(p,3);
                                  new_conn_count=new_conn_count+1;
                              end
                          end
                       id = int2str(conn_subareas_count);
                       Sstore = (strcat(filename2,str3c,id));
```

```
                        save(Sstore,'New_conn_net');     % save sub-
areas connecting zones points network creation
                        new_conn_count=1;
                        clear New_conn_net
                        borders_conn_hor_right(i,j)=0; % this avoid to
connect twice the same set
                        borders_conn_hor_left(i,k)=0;
                        end  % if k > i

                        %LEFT
                        if
((k<j)&&(borders_conn_hor_left(i,j)==1)&&(borders(i,k)==1))        %
looking left with respect to subarea (i,j)
                            str1c = int2str(i);
                            str2c = int2str(j);
                            filename1=('FEM_nodesdata_');
                            filename2=('FEM_conn_nodesdata_');
                            str3c='_';
                            S1 = (strcat(filename1,str1c,str3c,str2c));
                            str1v = int2str(i);
                            str2v = int2str(k);
                            str3v='_';
                            S2 = (strcat(filename1,str1v,str3v,str2v));
                            Acenter=load(S1);
                            Ahor=load(S2);
                            conn_subareas_count=conn_subareas_count+1;
                           Acent2=min(Acenter(:,2));
                           Acent1=Acent2+patchx;
                           Ahor2=max(Ahor(:,2));
                           Ahor1=Ahor2-patchx;
                           [ac,bc]=size(Acenter);
                           % nodes of (i,j) at the boundary with (i,k)
                           for p=1:ac
                               if
(Acenter(p,2)>=Acent2)&&(Acenter(p,2)<=Acent1)

New_conn_net(new_conn_count,1)=Acenter(p,1);

New_conn_net(new_conn_count,2)=Acenter(p,2);

New_conn_net(new_conn_count,3)=Acenter(p,3);
                                    new_conn_count=new_conn_count+1;
                               end
                           end
                           [av,bv]=size(Ahor);
                           % nodes of (i,k) at the boundary with (i,j)
                           for p=1:av
                               if
(Ahor(p,2)>=Ahor1)&&(Ahor(p,2)<=Ahor2)

New_conn_net(new_conn_count,1)=Ahor(p,1);

New_conn_net(new_conn_count,2)=Ahor(p,2);

New_conn_net(new_conn_count,3)=Ahor(p,3);
                                    new_conn_count=new_conn_count+1;
                               end
```

```
                        end
                        id = int2str(conn_subareas_count);
                        Sstore = (strcat(filename2,str3c,id));
                        save(Sstore,'New_conn_net');     % save sub-
areas connecting zones points network creation
                        new_conn_count=1;
                        clear New_conn_net
                        borders_conn_hor_left(i,j)=0; % this avoid to
connect twice the same set
                        borders_conn_hor_right(i,k)=0;
                        end % if k < i
%                       end %
                  end % end if k
              end % end for k
          end % end if border
      end % end main for j
end % end main for i

% save('New_conn_net','New_conn_net');


%
%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=467;

%ENDS SUB-AREAS ---- NETWORKS GENERATION 2/2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% VISUALIZATION & GEOMETRY DATA COLLECTION
% Visualize all the created struts and nodes to verify correctivness
load('model_size');
%number of sub areas
Nmatrix=TotalAreaModel/(Element_size*Element_size);
% number of sub areas per side
SNmatrix=sqrt(Nmatrix);


%
finalmatrix=zeros(Element_size*SNmatrix+2*bou,Element_size*SNmatrix+2*
bou);


startvalue2=1;
storagecounter=0;
```

```
for i=1:SNmatrix+2
    for j=1:SNmatrix+2
        if borders(i,j)==1
        storagecounter=storagecounter+1;
        str1 = int2str(i);
        str2 = int2str(j);
        filename1=('FEM_nodesdata_');
        filename2=('angledata_');
        filename3=('geodata_');
        str3='_';
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %load
        % FEM_nodesdata
        S1 = (strcat(filename1,str1,str3,str2));
        Anodes=load (S1);
        % angledata
        S2 = (strcat(filename2,str1,str3,str2));
        load (S2);
        % geodata
        S3 = (strcat(filename3,str1,str3,str2));
        load (S3);
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %Visualizer

%         [finalmatrixdone] =
networkvisualizer_single5_novis1(good_friends,Anodes,Element_size,SNma
trix,borders,finalmatrix,Mcounter,fiber_diameter,bou);
%         finalmatrix=finalmatrixdone;
%
        Mcounter=Mcounter+1;
        % Angle distribution
        n2TOT(storagecounter,1:length(n2))=n2;
        n2TOTinterval=xout2;
        % Angle values
%         value2TOT(storagecounter,1:length(value2))=value2;
        value2TOT(startvalue2:startvalue2+length(value2)-1)=value2;
        startvalue2=startvalue2+length(value2);
        value2TOTinterval=xout2;
        % Weighted angle distribution

n3TOT(storagecounter,1:length(counter_weighted3))=counter_weighted3;
        n3TOTinterval=interval3;
        % Connectivity distribution

Conn_count_newTOT(storagecounter,1:length(Conn_count_new))=Conn_count_
new;
        conn_interval_newTOT=conn_interval_new;
        end
    end
end
% figure(1)
% imshow(finalmatrixdone);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% CONNECTING NETWORKS CREATION
```

```
% additional connection creation and storing
aa=0;
for vi=1:conn_subareas_count
    storagecounter=storagecounter+1;

    filename2=('FEM_conn_nodesdata_');
    str3c='_';
    id = int2str(vi);
    Sstore = (strcat(filename2,str3c,id));
    filename3=('FEMelements');

    Sstorelements =(strcat(filename3,str3c,id));
    clear New_conn_net
    load (Sstore)

[FEMelements,n,xout,value,Conn_count,conn_interval,counter_weighted] =
orientationadaptation_con_net15_novis1(New_conn_net,fiber_diameter,bou
,patchstepMAX,ConnIndex,Ln_new,Lm_new);
    a=length(FEMelements);
    ba(vi)=a;
    aa=aa+a;
%     finalmatrixdone=finalmatrix;
    save (Sstorelements,'FEMelements');
    % storage of connetting subareas elements
%     clear finalmatrix
    % Angle distribution
    n2TOT(storagecounter,1:length(n))=n;
    n2TOTinterval=xout;
    % Angle values
%     value2TOT(storagecounter,1:length(value))=value;
    value2TOT(startvalue2:startvalue2+length(value)-1)=value;
    startvalue2=startvalue2+length(value);
    value2TOTinterval=xout;
    % Angle weighted distribution
    n3TOT(storagecounter,1:length(counter_weighted))=counter_weighted;
    n3TOTinterval=xout;
    % Connectivity distribution
    Conn_count_newTOT(storagecounter,1:length(Conn_count))=Conn_count;
    conn_interval_newTOT=conn_interval_new;
end
%     figure(2)
%     imshow(finalmatrixdone);



%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=586;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
```

```
% FEM DATA
% FEM data generator in ASCII format
kA=1;
kB=1;
for i=1:SNmatrix+2
    for j=1:SNmatrix+2
        if borders(i,j)==1
        str1 = int2str(i);
        str2 = int2str(j);
        filename1=('FEM_nodesdata_');
        filename2=('FEM_elementsdata_');
        str3='_';
        S1 = (strcat(filename1,str1,str3,str2));

        % nodes FEM data formation
        A=load (S1);
        idA=A(length(A),1);
        Anodes(kA:idA,1:3)=A;
        kA=idA+1;

        % elements FEM data formation original subareas
        S2 = (strcat(filename2,str1,str3,str2));
        B=load (S2);
        [idB b]=size(B);
        Aelements(kB:kB+idB-1,1:2)=B;
        kB=idB+kB;

        end
    end
end
save('FEM_nodesdata_TOT','Anodes','-ascii');
load('FEM_nodesdata_TOT');

% save('FEM_elementsdata_TOT','Aelements','-ascii');
% clear
% load('FEM_elementsdata_TOT');



 % elements FEM data formation connecting subareas
kB=1;
for vi=1:conn_subareas_count
    str3c='_';
    id = int2str(vi);
    filename3=('FEMelements');
    Sstorelements =(strcat(filename3,str3c,id));
    load (Sstorelements)
    BO=FEMelements;

    [idB b]=size(BO);
    Aelements2(kB:kB+idB-1,1:2)=BO;
    kB=idB+kB;
    clear FEMelements
    clear BO
end
```

```
if sum(sum(borders))==1
    % if to verify if subareas have been generated
   AelementsTOT=Aelements;
else
   AelementsTOT=[Aelements;Aelements2];
end
 save('FEM_elementsdata_TOT','AelementsTOT','-ascii');
 load('FEM_elementsdata_TOT');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%   MODEL GENERATED --> FROM NOW ON DATA COLLECTION AND STORAGE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% ARCHITECTURE DESCRIPTION
% get the global geometry  info
% disp('MODEL ARCHITECTURAL DESCRIPTORS')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% Angular Stiffness data
clear FEM_angular_stiffness_TOT
FEM_angular_stiffness_TOT(:,1)=FEM_nodesdata_TOT(:,1);
FEM_angular_stiffness_TOT(:,2)=0;

for i=1:length(FEM_nodesdata_TOT)
    for j=1:length(FEM_elementsdata_TOT)
          if
(FEM_nodesdata_TOT(i,1)==FEM_elementsdata_TOT(j,1))||(FEM_nodesdata_TO
T(i,1)==FEM_elementsdata_TOT(j,2))

FEM_angular_stiffness_TOT(i,2)=FEM_angular_stiffness_TOT(i,2)+1;
          end
    end
end
save('FEM_angular_stiffness_TOT','FEM_angular_stiffness_TOT','-
ascii');


%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=685;

%%%%%%%%%%%%%%
% connectivity

% total nuber of overlaps
```

```
total_number_overlaps=length(FEM_nodesdata_TOT);

%connectivity plot
Conn_count_newTOT=zeros(1,51);
for i=1:length(FEM_angular_stiffness_TOT)
    for j=1:51
        conncheck=j-1;
        if FEM_angular_stiffness_TOT(i,2)==conncheck;
            Conn_count_newTOT(1,j)=Conn_count_newTOT(1,j)+1;
        end
    end
end

% total number of lost overlaps
overlaps_lost=Conn_count_newTOT(1);
% check they must be the same
overlap_lost1=0;
for i=1:length(FEM_angular_stiffness_TOT)
    if FEM_angular_stiffness_TOT(i,2)==0
        overlap_lost1=overlap_lost1+1;
    end
end

% total number of bar elements
Number_bar_elements=sum(Conn_count_newTOT.*(0:(length(Conn_count_newTO
T)-1)))/2;
% check they must be the same
Number_bar_elements1=length(FEM_elementsdata_TOT);

% total number of overlaps including the lost overlaps
Number_nodes_conn=sum(Conn_count_newTOT);
% check they must be the same
Number_nodes_conn1=length(FEM_nodesdata_TOT);

%connectiviy plot
% Conn_count_newTOT=Conn_count_newTOT/(sum(Conn_count_newTOT));
[acon bcon]=find(Conn_count_newTOT>0);
xlimitmin=conn_interval_newTOT(min(bcon));
xlimitmax=conn_interval_newTOT(max(bcon));
ylimit=max(Conn_count_newTOT);
% ylimit=1;
% figure(3)
% bar(conn_interval_newTOT,Conn_count_newTOT,'k')
% axis([xlimitmin-1 xlimitmax+1 0 ylimit])
% title ('MODEL normalized  number of fiber overlaps VS connections')
% xlabel('connections')
% ylabel('normalized number of fiber overlaps')
Conn_count_normS=Conn_count_newTOT;
%%%%%%%%%%%%%%
% overlaps/area

% disp('overlaps density [N overlaps/square micrometers]')
Noverlaps=Number_nodes_conn;
% overlaps number needs to be reduced because some connections could
be
% lost during the processing
```

```
Eff_Noverlaps=Noverlaps-overlaps_lost;
overlaps_density=Eff_Noverlaps/model_area;
% disp(overlaps_density);
Nodes_densityS=overlaps_density;
%%%%%%%%%%%%%%%%%%%%%%%%%%
% orientation index from fiber anlge values

    %  fiber angles values
%     [a b]=size(value2TOT);
%     value2TOT=sum(value2TOT((1:a),:));
    value2TOT=value2TOT(:);
%     tetadefault=input('please provide the supposed orientation angle
');

    tetadefault=90;
    deltateta=(value2TOT-tetadefault);
    deltatetaradiant = (2*pi/360)*deltateta;
    oi=sum((cos(deltatetaradiant).^2));
    OI=oi/length(value2TOT);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% fiber angle distribution

    % normalized fiber alignment
    [a b]=size(n2TOT);
    if a>1
    n2TOT=sum(n2TOT((1:a),:));

    end
%normalization
    n2TOT=n2TOT/(sum(n2TOT));
    ylimit=max(n2TOT)+0.01;
%     ylimit=0.5;
%     figure(4)
%     bar(n2TOTinterval,n2TOT,'k')
%     axis([0 180 0 ylimit])
%     title ('MODEL normalized fiber count VS angle')
%     xlabel('angle')
%     ylabel('normalized fiber count')
    nS=n2TOT;

%
%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
%             linetemp(counttime)=788;


%     amid=round(length(n2TOT)/2);
%     text(n2TOT(amid),n2TOTinterval(amid),[num2str(OI),' Orientation
Index'],'FontSize',10);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% fiber angle weighted distribution

    % normalized fiber alignment
    [a b]=size(n3TOT);
```

```
    n3TOT=sum(n3TOT((1:a),:));
    % normalization
        n3TOT=n3TOT/(sum(n3TOT));
%         ylimit=max(n3TOT)+0.1;

%         figure(5)
%         bar(n3TOTinterval,n3TOT,'k')
%         axis([0 180 0 ylimit])
%         title ('MODEL normalized weighted fiber count VS angle')
%         xlabel('angle')
%         ylabel('normalized weighted fiber count')


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%

        n2TOTNEW(1)=sum(n2TOT(1:2));
        n2TOTNEW(2)=sum(n2TOT(3:4));
        n2TOTNEW(3)=sum(n2TOT(5:6));
        n2TOTNEW(4)=sum(n2TOT(7:8));
        n2TOTNEW(5)=sum(n2TOT(9:10));
        n2TOTNEW(6)=sum(n2TOT(11:12));
        n2TOTNEW(7)=sum(n2TOT(13:14));
        n2TOTNEW(8)=sum(n2TOT(15:16));
        n2TOTNEW(9)=sum(n2TOT(17:18));
        n2TOTNEW(10)=sum(n2TOT(19:20));
        n2TOTNEW(11)=sum(n2TOT(21:22));
        n2TOTNEW(12)=sum(n2TOT(23:24));
        n2TOTNEW(13)=sum(n2TOT(25:26));
        n2TOTNEW(14)=sum(n2TOT(27:28));
        n2TOTNEW(15)=sum(n2TOT(29:30));
        n2TOTNEW(16)=sum(n2TOT(31:32));
        n2TOTNEW(17)=sum(n2TOT(33:34));
        n2TOTNEW(18)=sum(n2TOT(35:36));
        n2TOTNEW(19)=sum(n2TOT(37));



        n2TOTintervalNEW=0:10:180;

        n2TOTNEW=n2TOTNEW/(sum(n2TOTNEW));
%         ylimit=max(n3TOTNEW);
%         ylimit=0.4;

%         figure(6)
%         ylimit=max(n2TOTNEW)+0.01;
%         bar(n2TOTintervalNEW,n2TOTNEW,'k')
%         axis([0 180 0 ylimit])
%         title ('MODEL normalized fiber count VS angle')
```

```
%         xlabel('angle')
%         ylabel('normalized fiber count')
        counter_weighted_normS=n2TOTNEW;

%%%%%%%%%%%%%%%%
% model geometry

% disp('model area in micrometers^2 ')

model_area=Area_new/(magn_fact^2);
% disp(model_area)

% disp('model size in micrometers')

model_size=(Element_size*SNmatrix)/magn_fact;
% disp(model_size)
%
% disp('Orientation Index')
% disp(OI)
OIS=OI;

% fiber diameter again converted in micrometers before being stored

fiber_diameter=fiber_diameter*100/magn;

% store final geometry information
save('MODELDATA','model_sizeS','Real_areaS','Nodes_densityS','Noverlap
s','Eff_Noverlaps','overlaps_lost','Conn_count_normS','conn_interval_n
ewTOT','n2TOTinterval','counter_weighted_normS','nS','OIS','Number_bar
_elements','DiameterS')


% % Nodes_densityS------------OK
% % OIS-----------------------OK
% % counter_weighted_normS----OK
% % nS------------------------OK
% % Conn_count_normS----------OK
% % DiameterS-----------------OK
% % Real_areaS ---------------OK
% % model_sizeS --------------OK


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% ASCII

for i=1:length(FEM_nodesdata_TOT)
    A=[FEM_nodesdata_TOT(i,1) FEM_nodesdata_TOT(i,2)
FEM_nodesdata_TOT(i,3)];
    FEM_ndata_TOT_ST(i,1:3)=A;

end
```

```
% convert to real size data, all in micrometers
FEM_nodesdata_TOT(:,2)=FEM_nodesdata_TOT(:,2)*100/magn;
FEM_nodesdata_TOT(:,3)=FEM_nodesdata_TOT(:,3)*100/magn;

FEM_ndata_TOT_ST(:,2)=FEM_ndata_TOT_ST(:,2)*100/magn;
FEM_ndata_TOT_ST(:,3)=FEM_ndata_TOT_ST(:,3)*100/magn;


save('FEM_nodesdata_TOT','FEM_nodesdata_TOT','-ascii');
dlmwrite('FEM_ndata_TOT_ST.txt',FEM_ndata_TOT_ST, 'delimiter', ',')




for i=1:length(FEM_elementsdata_TOT)
    A=[FEM_elementsdata_TOT(i,1) FEM_elementsdata_TOT(i,2)];
    FEM_edata_TOT_ST(i,1:2)=A;

end
dlmwrite('FEM_edata_TOT_ST.txt',FEM_edata_TOT_ST, 'delimiter', ',')



for i=1:length(FEM_angular_stiffness_TOT)
    A=[FEM_angular_stiffness_TOT(i,1) FEM_angular_stiffness_TOT(i,2)];
    FEM_angular_stiffness_TOT_ST(i,1:2)=A;

end
dlmwrite('FEM_angular_stiffness_TOT_ST.txt',FEM_angular_stiffness_TOT_
ST, 'delimiter', ',')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%

% check variables

% Number_nodes_conn
% Number_nodes_conn1
%
% overlaps_lost
% overlap_lost1
%
% Eff_Noverlaps
%
% Number_bar_elements
% Number_bar_elements1
%

%             counttime=counttime+1;
%             toc;
%             timeperf(counttime)=toc;
```

```
%             linetemp(counttime)=945;
%             FX = diff(timeperf);
%             figure(7)
%             grid on
%             plot(linetemp,timeperf,'xb');hold on
%             plot(linetemp,timeperf,'b');
%             plot(linetemp(1:length(linetemp)-1),FX,'xr');
%             plot(linetemp(1:length(linetemp)-1),FX,'r');hold off
% %
%             title ('algorithm performance evaluation, code vs time
and its gradient')
%             xlabel('code line')
%             ylabel('time [seconds]')
clear
load('FEM_nodesdata_TOT');
load('FEM_ndata_TOT_ST.txt');

load('FEM_elementsdata_TOT');
load('FEM_edata_TOT_ST.txt');

load('FEM_angular_stiffness_TOT');
load('FEM_angular_stiffness_TOT_ST.txt');

load('MODELDATA')
load('SIMULATIONDATA')



            %%%%%%%%%%%%%%%%%%%%%%%%
            % debug completed May 1st 2009
            %%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX D)

FEM ANALYSIS CODE (ANSYS)

```
/clear,nostart              ! clear the database, nostart do restart the default file
!*************************************************************************
! ***** PRE-PROCESSING *****
!*************************************************************************
/prep7                          ! start preprocessor
 /title,80 micron 9.0          ! title
/nerr,5,99999999,,0           ! increase number erros allowed
                              !***constants definition***
X1=82                           ! width of the model
Y1=82                          ! height of the model
poisson=0.3                   ! Poisson ratio
ElasticityModuli=45                ! Ex Modulus      (45 --> 45 MPa first guess)
ShearModuli=ElasticityModuli/((1+poisson)*2)       ! shear moduli
fiberdiameter=0.4790                                ! fiber diameter from the image analysis
xfiberarea=3.14*(fiberdiameter)*(fiberdiameter)/4   ! cross section area
                              !***mesh definition***
                              !*** import nodes ***
nread,nodes.txt                 !import nodes, define nodes, (command,filename.txt),
!nodes must respect a specific format see the Matlab file
                               !**** material real properties defini
```

```
!r,NSET, R1, R2, R3, R4, R5, R6          ! Defines the element real constants.

!interpreted as area, moment of inertia, thickness, etc., as required for the particular
!element type using this set

! in the case of link10 R2 represents the initial strain

real,1

r,1,xfiberarea,1E-6,0,0,0,0

!cross section area, pi*(D^2)/4

!-1E-6 negative pre-strain to simulate the slack --> tortuosity




                                        ! *** define the element type ***

et,1,link10,0,2,0                       !keyopt(1) no info,

                                !keyopt(2) = 2 Small stiffness assigned to slack cable for both
!longitudinal and perpendicular motions (applicable only with stress stiffening)
!keyopt(3) = 0 Tension-only (cable) option

eread,elements.txt                      !*** import elements ***

!edele,383                               ! erase elements  that are accidentaly coincident in this
!mesh (numbers vary from mesh to mesh)

!edele,6214



                                !  *** set element material properties


mp,ex,1,ElasticityModuli                ! Ex Modulus       (45 MPa first guess)

mp,prxy,1,poisson                       ! Poisson's ratio

mp,dens,1,940                           ! material density (940 Kg/m3 from Nick
!Amoroso data first guess)
```

```
!mp,gxy,1,ShearModuli                    ! shear moduli

Finish                                   ! stop preprocessor

!*************************************************************************

! stop preprocessor

!*************************************************************************

!*************************************************************************

!********** SOLUTION *****

!*************************************************************************

/solu                                    ! start solution phase

/GST                                     ! graphical solution tracking

antype,static,new                        ! static analysis, new re-set all the analysis
!variables (clean previous variables)

                                         !*** large deformation and newton raphson method ***

nlgeom,on                                ! turn on large deflection effects in static or full
!transient analysis

solcontrol,on,on                         !Specifies whether to use optimized nonlinear solution
!defaults and some enhanced internal solution algorithms

                                         !key (1) on Activates optimized defaults for a set of commands
!applicable to nonlinear solutions

                                         !key (2) on ANSYS ensures the time step is small enough to
!account for changes in nonlinear element status

                                         !key (3) and key (4) are not necessary for link10

!nropt,full,,off                         ! Specifies the Newton-Raphson options in a static or full
!transient analysis.

                                         ! Use full Newton-Raphson

                                         ! Adaptive descent key off because of solcontrol on
```

```
!lnsrch,on                    ! Activates a line search to be used with Newton-Raphson.

pred,off,,off                 ! Activates a predictor in a nonlinear analysis, key(1) no
!prediction occurs on the substeps key(2) no predictions occures on the load steps.

                              ! Rotation DOFs are present so it must be turned off according
!to the manual

cnvtol,f,,,0,4                !Sets convergence values for nonlinear analyses ,
!CNVTOL, Lab, VALUE, TOLER, NORM, MINREF

                              ! lab key(1)=f the convergence is checked on the forces

                              ! value=  Defaults to the maximum of a program calculated
!reference or MINREF (key(4))

                              ! toler=  When SOLCONTROL,ON, tolerance about
!VALUE. Defaults to 0.005 (0.5%) for force and moment,

                              ! norm=0  0 Infinite norm (check each DOF separately).

                              ! minref=4  The minimum value allowed for the program
!calculated reference value. Used only if VALUE is blank

                              !        default value is 0.01

                              !        tryal and errore base from the !force value where all
the fibers are recruited which appers to be 2000 --> 2000*0.005=4

nsubst,100,10000,10           !number of substeps, minimum 10, max 10000,
!suggested 100

!*****************************************************************************
************************************

outres,all,last               !Controls the solution data written to the database.

                              !RSOL  —  Nodal reaction loads, ALL get all the information.

                              !Writes the specified solution results item only for the last
!substep of each load step



last_strain=9                 ! last-strain x 0.05 gives the maximun strain level and
!strainstep gives the strain steps
```

```
strainstep=0.05

                    ! ***array definition for storing data***

nsel,all

                    ! Defines an array parameter and its dimensions.

*DIM,FXP_dir_FX,ARRAY,1,last_strain,1    ! 1xlast_strain dimension array  resultant !FX, FY

*DIM,FXP_dir_FY,ARRAY,1,last_strain,1


!**********************START THE DO LOOP

nsel,all

*do,t,1,last_strain,1

        /gopr                       !Reactivates suppressed printout.

    sigma=t*strainstep              !sigma imposed

                                !***set the boundary conditions ***


    nsele,all                       ! boundary condition 1) neglecting additional degree of freedom if using  beam188, rotx and roty

    !d,all,rotx,0

    !d,all,roty,0

    d,all,uz,0

    nsel,s,loc,x,0,X1/15            ! select nodes to apply boundary conditions !on left edge

    nsel,r,loc,y,Y1*0.2,Y1*0.8

    d,all,ux,0                  ! boundary condition  2) no ux for left edge

    !d,all,uy,0                 ! uncomment for uniaxial
```

```
nsel,all

nsel,s,loc,y,-Y1/15,Y1/15                    !  boundary condition 3) select nodes to
!apply boundary conditions on bottom edge

nsel,r,loc,x,X1*0.2,X1*0.8

d,all,uy,0                                   !  comment for uniaxial



nsel,s,loc,x,(X1-(X1/15)),X1+(X1/15)         ! boundary condition 4) select
!nodes to apply boundary conditions on right edge

nsel,r,loc,y,Y1*0.2,Y1*0.8

d,all,ux,sigma*X1                            ! move along ux

nsel,s,loc,y,(Y1-(Y1/15)),Y1+(Y1/15)         ! boundary condition 5) select
!nodes to apply boundary conditions on on top edge

nsel,r,loc,x,X1*0.2,X1*0.8

d,all,uy,sigma*0.31*Y1                       ! move along uy

                                             ! IMPORTANT multiplied by 0.3 in order to simulate
!equistress conditions applying different stretch over the two axis

                                             ! when the XP direction is at its maximum e.g 0.45
!stretch the PD is at 0.135

nsel,all

solve

Finish


!*********************************************************************

! stop solution

!*********************************************************************
```

```
!*************************************************************************

     !*********** POST PROCESSING *****

!*************************************************************************

          /POST1                         ! Enter POST1


       nsel,s,loc,x,0,X1/15               ! select nodes on the left edge they will provide the response over the XD (horizontal) axis

          nsel,r,loc,y,Y1*0.2,Y1*0.8

          ESLN,S                          ! select elements on the left edge they will provide the response over the XP axis

          FSUM,,                          ! sums the nodal force and moment contributions of elements

          *GET,FSUMX,FSUM,0,ITEM,FX       ! storing the force sum in the array FXP_dir_FX

          FXP_dir_FX(1,t)=FSUMX

          nsel,s,loc,y,0,Y1/15            ! select nodes on the left edge they will provide the response over the PD axis

          nsel,r,loc,x,X1*0.2,X1*0.8

          ESLN,S                          ! select elements on the left edge they will provide the response over the PD axis

          FSUM,,                          ! sums the nodal force and moment contributions of elements

          *GET,FSUMY,FSUM,0,ITEM,FY       ! storing the force sum in the array FXP_dir_FY

          FXP_dir_FY(1,t)=FSUMY

          parsave,all,morehope            ! saves parameters before antype restart !(antype,,rest erase everything--> not possible top store results in a txt)
```

```
FINISH

!*************************************************************************

! stop post processing

!*************************************************************

!*************************************************************************

!********** Re-enter SOLUTION *****

!*************************************************************

/SOLU

*if, t,lt,last_strain, then

antype,,rest                    !restart each step from the previous 1-2 , 2-3 , 3-4 !otherwise it does 1-2, 1-3, 1-4

*endif

parresu,,morehope    ! loa parameters before antype restart (antype,,rest erase !everything--> not possible top store results in a txt)



*enddo

!*************************************************************************

!*********************END THE DO LOOP

!*************************************************************************
```

```
!*********************************************************************

!********** POST PROCESSING *****

!*********************************************************************


*MWRITE,FXP_DIR_FX,XDprefe_forces,txt,,IJK,1,last_strain,1        ! store the XD force

%G

*MWRITE,FXP_DIR_FY,PDprefe_forces,txt,,IJK,1,last_strain,1        ! store the PD force

%G

ALLSEL,ALL

finish

!*********************************************************************

! stop post processing

!*********************************************************************

 ! ***** DE-BUGGING COMPLETED ALL COMMANDS VERIFIED UP TO THIS
! 09/02/2010 *****
```

BIBLIOGRAPHY

1. Langer, R. and J.P. Vacanti, *Tissue Engineering.* Science, 1993. **260**: p. 920-926.
2. Merryman, W.D., et al., *Synergistic effects of cyclic tension and transforming growth factor-beta1 on the aortic valve myofibroblast.* Cardiovasc Pathol, 2007. **16**(5): p. 268-76.
3. Shelton, J.C., D.L. Bader, and D.A. Lee, *Mechanical conditioning influences the metabolic response of cell-seeded constructs.* Cells Tissues Organs, 2003. **175**(3): p. 140-50.
4. van der Meulen, M.C. and R. Huiskes, *Why mechanobiology? A survey article.* J Biomech, 2002. **35**(4): p. 401-14.
5. Sarkadi, B. and J.C. Parker, *Activation of ion transport pathways by changes in cell volume.* Biochim Biophys Acta, 1991. **1071**(4): p. 407-27.
6. Billiar, K.L. and M.S. Sacks, *Biaxial mechanical properties of the natural and glutaraldehyde treated aortic valve cusp--Part I: Experimental results.* Journal of Biomechanical Engineering, 2000a. **122**(1): p. 23-30.
7. Sacks, M.S., *Biomechanics of native and engineered heart valve tissues*, in *Functional Tissue Engineering*, F. Guilak, et al., Editors. 2003, Spring-Verlag: New York.
8. Vesely, I. and A. Lozon, *Natural preload of aortic valve leaflet components during glutaraldehyde fixation: Effects on tissue mechanics.* Journal of Biomechanics, 1993. **26**(2): p. 121-131.

9. Stella, J.A., J. Liao, and M.S. Sacks, *Time-dependent biaxial mechanical behavior of the aortic heart valve leaflet.* J Biomech, 2007.

10. Sacks, M.S. *Biaxial mechanical behavior of bovine pericardium as a bioprosthetic material*. in *Proceedings of the 11th Conference Engineering, Engineering Mechanics Division/ASCE*. 1996. Fort Lauderdale, FL.

11. Smith, D.B., et al., *Fatigue-induced changes in bioprosthetic heart valve three-dimensional geometry and the relation to tissue damage.* J Heart Valve Dis, 1999. **8**(1): p. 25-33.

12. Sacks, M.S. and F.J. Schoen, *Calcification-independent damage to collagen in explanted bioprosthetic heart valves.* Journal of Thoracic and Cardiovascular Surgery, 2000. **submitted**.

13. Vesely, I., J.E. Barber, and N.B. Ratliff, *Tissue damage and calcification may be independent mechanisms of bioprosthetic heart valve failure.* J Heart Valve Dis, 2001. **10**(4): p. 471-7.

14. Sung, H.W., et al., *In vitro evaluation of cytotoxicity of a naturally occurring cross-linking reagent for biological tissue fixation.* J Biomater Sci Polym Ed, 1999. **10**(1): p. 63-78.

15. Schoen, F.J., *Future directions in tissue heart valves: impact of recent insights from biology and pathology.* J Heart Valve Dis, 1999. **8**(4): p. 350-8.

16. M. H. Zheng, J.C.Y.K.C.W.J.X.D.W., *Porcine small intestine submucosa (SIS) is not an acellular collagenous matrix and contains porcine DNA: Possible implications in human implantation.* Journal of Biomedical Materials Research Part B: Applied Biomaterials, 2005. **73B**(1): p. 61-67.

17. Badylak, S., *The extracellular matrix as a biologic scaffold material.* Biomaterials, 2007. **28**: p. 3587-3593.

18. Badylak, S.F. and T.W. Gilbert, *Immune response to biologic scaffold materials.* Semin Immunol, 2008. **20**(2): p. 109-16.

19. Gilbert, T.W., et al., *Fiber kinematics of small intestinal submucosa under biaxial and uniaxial stretch.* J Biomech Eng, 2006. **128**(6): p. 890-8.

20. Gilbert, T.W., et al., *Degradation and remodeling of small intestinal submucosa in canine Achilles tendon repair.* J Bone Joint Surg Am, 2007. **89**(3): p. 621-30.

21. Liao, J., E.M. Joyce, and M.S. Sacks, *Effects of decellularization on the mechanical and structural properties of the porcine aortic valve leaflet.* Biomaterials, 2008. **29**(8): p. 1065-74.

22. Donald O. Freytes, R.M.S.S.F.B., *Uniaxial and biaxial properties of terminally sterilized porcine urinary bladder matrix scaffolds.* Journal of Biomedical Materials Research Part B: Applied Biomaterials, 2008. **84B**(2): p. 408-414.

23. Bidanda, B. and P. Bartolo, *Virtual prototyping & bio manufacturing in medical applications*. 2008, New York: Springer. xv, 299.

24. Lips, P., et al., *Gas foaming of segmented poly(ester amide) films* Polymer, 2005. **46**(22): p. 9396-9403.

25. Temenoff, J.S. and A.G. Mikos, *Review: tissue engineering for regeneration of articular cartilage.* Biomaterials, 2000. **21**(5): p. 431-40.

26. Guan, J., K.L. Fujimoto, and W.R. Wagner, *Elastase-sensitive elastomeric scaffolds with variable anisotropy for soft tissue engineering.* Pharm Res, 2008. **25**(10): p. 2400-12.

27. Atkinson, P.M. and D.R. Lloyd, *Anisotropic flat sheet membrane formation via TIPS: atmospheric convection and polymer molecular weight effects.* Journal of Membrane Science, 2000. **175**(2): p. 225-238.

28. Barnes, C.P., et al., *Nanofiber technology: Designing the next generation of tissue engineering scaffolds.* Adv Drug Deliv Rev, 2007.

29. Zhang, S., *Fabrication of novel biomaterials though molecular self-assembly.* Nature Biotechnology, 2003. **21**(10): p. 1171-1178.

30. Meyers, R.A., *Molecular biology and biotechnology : a comprehensive desk reference*. 1995, New York: VCH. xxxviii, 1034.

31. Nedoviâc, V. and R. Willaert, *Applications of cell immobilisation biotechnology*. Focus on biotechnology ; v. 8B. 2005, Dordrecht ; New York: Springer. vi, 573.

32. Jen, A.C., M.C. Wake, and A.G. Mikos, *Review: Hydrogels for cell immobilization.* Biotechnol Bioeng, 1996. **50**(4): p. 357-64.

33. Anseth, K., C. Bowman, and L. Brannon-Peppas, *Mechanical properties of hydrogels and their experimental determination* Biomaterials, 1996. **17**: p. 1647-1657.

34. Roeder, B.A., et al., *Local, Three-Dimensional Strain Measurements Within Largely Deformed Extracellular Matrix Constructs.* Journal of Biomechanical Engineering, 2004. **126**(6): p. 699-708.

35. Preethi, L.C. and H.B. Victor, *Deterministic Material-Based Averaging Theory Model of Collagen Gel Micromechanics.* Journal of Biomechanical Engineering, 2007. **129**(2): p. 137-147.

36. Neidert, M.R. and R.T. Tranquillo, *Tissue-engineered valves with commissural alignment.* Tissue Eng, 2006. **12**(4): p. 891-903.

37. Robinson, P.S., et al., *Functional Tissue-Engineered Valves from Cell-Remodeled Fibrin with Commissural Alignment of Cell-Produced Collagen.* Tissue Eng, 2007.

38. Thomopoulos, S., G.M. Fomovsky, and J.W. Holmes, *The development of structural and mechanical anisotropy in fibroblast populated collagen gels.* J Biomech Eng, 2005. **127**(5): p. 742-50.

39. Isenberg, B.C., C. Williams, and R.T. Tranquillo, *Endothelialization and flow conditioning of fibrin-based media-equivalents.* Ann Biomed Eng, 2006. **34**(6): p. 971-85.

40. Isenberg, B.C., C. Williams, and R.T. Tranquillo, *Small-diameter artificial arteries engineered in vitro.* Circ Res, 2006. **98**(1): p. 25-35.

41. Williams, C., et al., *Cell sourcing and culture conditions for fibrin-based valve constructs.* Tissue Eng, 2006. **12**(6): p. 1489-502.

42. Grassl, E.D., T.R. Oegema, and R.T. Tranquillo, *A fibrin-based arterial media equivalent.* J Biomed Mater Res A, 2003. **66**(3): p. 550-61.

43. Rosner, B.I., T. Hang, and R.T. Tranquillo, *Schwann cell behavior in three-dimensional collagen gels: evidence for differential mechano-transduction and the influence of TGF-beta 1 in morphological polarization and differentiation.* Exp Neurol, 2005. **195**(1): p. 81-91.

44. Karamichos, D., R.A. Brown, and V. Mudera, *Collagen stiffness regulates cellular contraction and matrix remodeling gene expression.* J Biomed Mater Res A, 2007. **83**(3): p. 887-94.

45. Thomopoulos, S., Fomovsky, Gregory M., Chandran Preethi L., Holmes, Jeffry W. , *Collagen fiber alignment does not explain mechanical anisotropy in fibroblasr populated collagen gels.* JBME, 2007. **129**(5): p. 642-650.

46. Lanir, Y., *Rheological behavior of skin.* The rheology of blood, blood vessels and associated tissues, 1980: p. 276-294.

47. Sacks, M.S., *Incorporation of experimentally-derived fiber orientation into a structural constitutive model for planar collagenous tissues.* J Biomech Eng, 2003. **125**(2): p. 280-7.

48. Sacks, M.S., *A structural constitutive model for chemically treated planar tissues under biaxial loading.* Computational Mechanics, 2000. **26**: p. 243-249.

49. Freed, L.E., et al., *Biodegradable Polymer Scaffolds for Tissue Engineering.* Nature Biotechnology, 1994. **12**: p. 689-693.

50. Engelmayr, G.C. and M.S. Sacks, *A structural model for the flexural mechanics of nonwoven tissue engineering scaffolds.* J Biomech Eng, 2006. **128**: p. 610-622.

51. Sutherland, F.W., et al., *From stem cells to viable autologous semilunar heart valve.* Circulation, 2005. **111**(21): p. 2783-91.

52. Engelmayr, G.C., Jr. and M.S. Sacks, *Prediction of extracellular matrix stiffness in engineered heart valve tissues based on nonwoven scaffolds.* Biomech Model Mechanobiol, 2007.

53. Venugopal, J., et al., *Interaction of cells and nanofiber scaffolds in tissue engineering.* J Biomed Mater Res B Appl Biomater, 2007.

54. Chen, M., et al., *Role of fiber diameter in adhesion and proliferation of NIH 3T3 fibroblast on electrospun polycaprolactone scaffolds.* Tissue Eng, 2007. **13**(3): p. 579-87.

55. Courtney, T., et al., *Design and analysis of tissue engineering scaffolds that mimic soft tissue mechanical anisotropy.* Biomaterials, 2006. **27**(19): p. 3631-8.

56. Pham, Q.P., U. Sharma, and A.G. Mikos, *Electrospinning of polymeric nanofibers for tissue engineering applications: a review.* Tissue Eng, 2006. **12**(5): p. 1197-211.

57. Teo, W.E., W. He, and S. Ramakrishna, *Electrospun scaffold tailored for tissue-specific extracellular matrix.* Biotechnol J, 2006. **1**(9): p. 918-29.

58. Nerurkar, N.L., D.M. Elliott, and R.L. Mauck, *Mechanics of oriented electrospun nanofibrous scaffolds for annulus fibrosus tissue engineering.* J Orthop Res, 2007. **25**(8): p. 1018-28.

59. Yoshimoto, H., et al., *A biodegradable nanofiber scaffold by electrospinning and its potential for bone tissue engineering.* Biomaterials, 2003. **24**(12): p. 2077-82.

60. Stankus, J.J., et al., *Microintegrating smooth muscle cells into a biodegradable, elastomeric fiber matrix.* Biomaterials, 2006. **27**(5): p. 735-44.

61. Zhong, S., et al., *An aligned nanofibrous collagen scaffold by electrospinning and its effects on in vitro fibroblast culture.* J Biomed Mater Res A, 2006. **79**(3): p. 456-63.

62. Tian, F., et al., *Quantitative analysis of cell adhesion on aligned micro- and nanofibers.* J Biomed Mater Res A, 2007.

63. Li, W.J., et al., *Engineering controllable anisotropy in electrospun biodegradable nanofibrous scaffolds for musculoskeletal tissue engineering.* J Biomech, 2007. **40**(8): p. 1686-93.

64. Stella, J.A., et al., *Tissue-to-cellular level deformation coupling in cell micro-integrated elastomeric scaffolds.* Biomaterials, 2008. **29**(22): p. 3228-36.

65. Riboldi, S.A., et al., *Electrospun degradable polyesterurethane membranes: potential scaffolds for skeletal muscle tissue engineering.* Biomaterials, 2005. **26**(22): p. 4606-15.

66. Katarzyna, G., Sylwester, G., *Biodegradable porous polyurethane scaffolds for tissue repair and regeneration.* Journal of Biomedical Materials Research, 2005. **79A**(1): p. 128-38.

67. Rockwood, D.N., et al., *Characterization of biodegradable polyurethane microfibers for tissue engineering.* J Biomater Sci Polym Ed, 2007. **18**(6): p. 743-58.

68. Wake, M.C., P.K. Gupta, and A.G. Mikos, *Fabrication of pliable biodegradable polymer foams to engineer soft tissues.* Cell Transplant, 1996. **5**(4): p. 465-73.

69. Guan, J., et al., *Preparation and characterization of highly porous, biodegradable polyurethane scaffolds for soft tissue applications.* Biomaterials, 2005. **26**(18): p. 3961-71.

70. Stankus, J.J., J. Guan, and W.R. Wagner, *Fabrication of biodegradable elastomeric scaffolds with sub-micron morphologies.* J Biomed Mater Res, 2004. **70A**(4): p. 603-14.

71. Vunjak-Novakovic, G., et al., *Bioreactor cultivation conditions modulate the composition and mechanical properties of tissue-engineered cartilage.* J Orthop Res, 1999. **17**(1): p. 130-8.

72. Wang, J.H. and B.P. Thampatty, *An introductory review of cell mechanobiology.* Biomech Model Mechanobiol, 2006. **5**(1): p. 1-16.

73. Butcher, J.T., C.A. Simmons, and J.N. Warnock, *Mechanobiology of the aortic heart valve.* J Heart Valve Dis, 2008. **17**(1): p. 62-73.

74. Matsumoto, T., et al., *Mechanical strain regulates endothelial cell patterning in vitro.* Tissue Eng, 2007. **13**(1): p. 207-17.

75. Yanagisawa, M., et al., *Effects of compressive force on the differentiation of pluripotent mesenchymal cells.* Life Sci, 2007. **81**(5): p. 405-12.

76. Terraciano, V., et al., *Differential response of adult and embryonic mesenchymal progenitor cells to mechanical compression in hydrogels.* Stem Cells, 2007. **25**(11): p. 2730-8.

77. Song, G., et al., *Mechanical stretch promotes proliferation of rat bone marrow mesenchymal stem cells.* Colloids Surf B Biointerfaces, 2007. **58**(2): p. 271-7.

78. Screen, H.R., et al., *Cyclic tensile strain upregulates collagen synthesis in isolated tendon fascicles.* Biochem Biophys Res Commun, 2005. **336**(2): p. 424-9.

79. MacKenna, D., S.R. Summerour, and F.J. Villarreal, *Role of mechanical factors in modulating cardiac fibroblast function and extracellular matrix synthesis.* Cardiovasc Res, 2000. **46**(2): p. 257-63.

80. Kulik, T.J. and S.P. Alvarado, *Effect of stretch on growth and collagen synthesis in cultured rat and lamb pulmonary arterial smooth muscle cells.* J Cell Physiol, 1993. **157**(3): p. 615-24.

81. Gupta, V. and K.J. Grande-Allen, *Effects of static and cyclic loading in regulating extracellular matrix synthesis by cardiovascular cells.* Cardiovasc Res, 2006. **72**(3): p. 375-83.

82. Pei, M., et al., *Bioreactors mediate the effectiveness of tissue engineering scaffolds.* Faseb J, 2002. **16**(12): p. 1691-4.

83. Gooch, K.J., et al., *Effects of mixing intensity on tissue-engineered cartilage.* Biotechnol Bioeng, 2001. **72**(4): p. 402-7.

84. Bursac, N., et al., *Cultivation in rotating bioreactors promotes maintenance of cardiac myocyte electrophysiology and molecular properties.* Tissue Eng, 2003. **9**(6): p. 1243-53.

85. Buschmann, M.D., et al., *Mechanical compression modulates matrix biosynthesis in chondrocyte/agarose culture.* J Cell Sci, 1995. **108 (Pt 4)**: p. 1497-508.

86. Seidel, J.O., et al., *Long-term culture of tissue engineered cartilage in a perfused chamber with mechanical stimulation.* Biorheology, 2004. **41**(3-4): p. 445-58.

87. Fink, C., et al., *Chronic stretch of engineered heart tissue induces hypertrophy and functional improvement.* Faseb J, 2000. **14**(5): p. 669-79.

88. Zimmermann, W.H., et al., *Tissue engineering of a differentiated cardiac muscle construct.* Circ Res, 2002. **90**(2): p. 223-30.

89. Mauck, R.L., et al., *Functional tissue engineering of articular cartilage through dynamic loading of chondrocyte-seeded agarose gels.* J Biomech Eng, 2000. **122**(3): p. 252-60.

90. Kim, B.S., et al., *Cyclic mechanical strain regulates the development of engineered smooth muscle tissue.* Nat Biotechnol, 1999. **17**(10): p. 979-83.

91. Akhyari, P., et al., *Mechanical stretch regimen enhances the formation of bioengineered autologous cardiac muscle grafts.* Circulation, 2002. **106**(12 Suppl 1): p. I137-42.

92. Long, J.L. and R.T. Tranquillo, *Elastic fiber production in cardiovascular tissue-equivalents.* Matrix Biol, 2003. **22**(4): p. 339-50.

93. Guilak, F., A. Ratcliffe, and V.C. Mow, *Chondrocyte deformation and local tissue strain in articular cartilage: a confocal microscopy study.* J Orthop Res, 1995. **13**(3): p. 410-21.

94. Ingber, D.E., *Mechanobiology and diseases of mechanotransduction.* Ann Med, 2003. **35**(8): p. 564-77.

95. Lee, D.A. and M.M. Knight, *Mechanical loading of chondrocytes embedded in 3D constructs: in vitro methods for assessment of morphological and metabolic response to compressive strain.* Methods Mol Med, 2004. **100**: p. 307-24.

96. Sheetz, M.P., *Cell control by membrane-cytoskeleton adhesion.* Nat Rev Mol Cell Biol, 2001. **2**(5): p. 392-6.

97. Discher, D.E., P. Janmey, and Y.L. Wang, *Tissue cells feel and respond to the stiffness of their substrate.* Science, 2005. **310**(5751): p. 1139-43.

98. Engler, A., et al., *Substrate compliance versus ligand density in cell on gel responses.* Biophys J, 2004. **86**(1 Pt 1): p. 617-28.

99. Jiang, G., et al., *Rigidity Sensing at the Leading Edge through ±v²3 Integrins and RPTP*±. 2006. **90**(5): p. 1804-1809.

100. Thomas, C.H., et al., *Engineering gene expression and protein synthesis by modulation of nuclear shape.* Proc Natl Acad Sci U S A, 2002. **99**(4): p. 1972-7.

101. Huang, H.Y., J. Liao, and M.S. Sacks, *In-situ deformation of the aortic valve interstitial cell nucleus under diastolic loading.* J Biomech Eng, 2007. **129**(6): p. 880.

102. Johnson, J., A. Ghosh, and J. Lannutti, *Microstructure-Property Relationships in a Tissue-Engineering Scaffold.* Journal of Applied Polymer Science, 2007. **104**: p. 2919-2927.

103. Fung, Y.C., K. Fronek, and P. Patitucci, *Pseudoelasticity of arteries and the choice of its mathematical expression.* Am J Physiol, 1979. **237**(5): p. H620-31.

104. David, G., et al., *Regional multiaxial mechanical properties of the porcine anterior lens capsule.* J Biomech Eng, 2007. **129**(1): p. 97-104.

105. Lanir, Y., *Constitutive Equations for Fibrous Connective Tissues.* journal of biomechanics, 1983. **16**: p. 1-12.

106. Driessen, N.J., C.V. Bouten, and F.P. Baaijens, *A structural constitutive model for collagenous cardiovascular tissues incorporating the angular fiber distribution.* J Biomech Eng, 2005. **127**(3): p. 494-503.

107. Choi, H.S. and R.P. Vito, *Two-dimensional stress-strain relationship for canine pericardium.* J Biomech Eng, 1990. **112**(2): p. 153-9.

108. Humphrey, J.D., D.L. Vawter, and R.P. Vito, *Pseudoelasticity of Excised Visceral Pleura.* Journal of Biomechanical Engineering, 1987. **109**: p. 115-120.

109. Fung, Y.C., *A model of the lung structure and its validation.* J Appl Physiol, 1988. **64**(5): p. 2132-41.

110. Holzapfel, G.A., *Determination of material models for arterial walls from uniaxial extension tests and histological structure.* J Theor Biol, 2006. **238**(2): p. 290-302.

111. Humphrey, J.D. and F.C.P. Yin, *A new constitutive formulation for characterizing the mechanical behavior of soft tissues.* Biophysical Journal, 1987. **52**: p. 563-570.

112. Holzapfel, G.A. and T.C. Gasser, *A new constitutive framework for arterial wall mechanics and a comparative study of material models.* Journal of Elasticity, 2000. **61**: p. 1-48.

113. Lanir, Y., *Plausibility of Structural Constitutive-Equations for Isotropic Soft-Tissues in Finite Static Deformations.* Journal of Applied Mechanics-Transactions of the Asme, 1994. **61**(3): p. 695-702.

114. Lanir, Y., *Plausibility of structural constitutive equations for swelling tissues--implications of the C-N and S-E conditions.* J Biomech Eng, 1996. **118**(1): p. 10-6.

115. Sacks, M.S., *A structural constitutive model for chemically treated planar connective tissues under biaxial loading.* Computational Mechanics, 2000. **26**(3): p. 243-249.

116. Agoram, B. and V.H. Barocas, *Coupled macroscopic and microscopic scale modeling of fibrillar tissues and tissue equivalents.* J Biomech Eng, 2001. **123**(4): p. 362-9.

117. Stylianopoulos, T. and V.H. Barocas, *Volume-averaging theory for the study of the mechanics of collagen networks.* Computer Methods in Applied Mechanics and Engineering, 2007. **196**(31-32): p. 2981-2990.

118. Zahalak, G.I., et al., *A cell-based constitutive relation for bio-artificial tissues.* Biophys J, 2000. **79**(5): p. 2369-81.

119. Marquez, J.P., et al., *The relationship between cell and tissue strain in three-dimensional bio-artificial tissues.* Biophys J, 2005. **88**(2): p. 778-89.

120. Marquez, J.P., et al., *Thin bio-artificial tissues in plane stress: the relationship between cell and tissue strain, and an improved constitutive model.* Biophys J, 2005. **88**(2): p. 765-77.

121. Stylianopoulos, A., et al., *Computational predictions of the tensile properties of electrospun fibre meshes: Effect of fibre diameter and fibre orientation.* Journal of Mechanical Behavior of Biomedical Materials, 2008. **1**: p. 326-335.

122. Newton, D., et al., *Regulation of material properties in electrospun scaffolds: Role of cross-linking and fiber tertiary structure.* Acta Biomater., 2009. **5**(1): p. 518-529.

123. Bashur, C.A., L.A. Dahlgren, and A.S. Goldstein, *Effect of fiber diameter and orientation on fibroblast morphology and proliferation on electrospun poly(d,l-lactic-co-glycolic acid) meshes.* Biomaterials, 2006. **27**(33): p. 5681-5688.

124. Baker, S.C., et al., *Characterisation of electrospun polystyrene scaffolds for three-dimensional in vitro biological studies.* Biomaterials, 2006. **27**(16): p. 3136-3146.

125. Chandran, P.L. and V.H. Barocas, *Deterministic material-based averaging theory model of collagen gel micromechanics.* J Biomech Eng., 2007. **129**(2): p. 137-147.

126. Stylianopoulos, A., et al., *Computational predictions of the tensile properties of electrospun fibre meshes: effect of fibre diameter and fibre orientation.* J Mech Behav Biomed Mater, 2008. **1**: p. 326-335.

127. Stylianopoulos, T. and V.H. Barocas, *Volume-averaging theory for the study of the mechanics of collagen networks.* Comput. Methods Appl. Mech. Engrg., 2007. **196**(31-32): p. 2981-2990.

128. Stylianopoulos, T. and V.H. Barocas, *Multiscale, structure-based modeling for the elastic mechanical behavior of arterial walls.* J Biomech Eng., 2007. **129**(4): p. 611-618.

129. Burd, H.J., *A structural constitutive model for the human lens capsule.* Biomech Model Mechanobiol, 2008. **8**(3): p. 217-231.

130. Wang, C.W., L. Berhan, and A.M. Sastry, *Structure, mechanics and failure of stochastic fibrous networks: part I—microscale considerations.* J Eng Mater-T, 2000. **122**(4): p. 450-459.

131. Wang, C.W. and A.M. Sastry, *Structure, mechanics and failure of stochastic fibrous networks : Part II-Network simulations and application.* J Eng Mater-T, 2000. **122**(4): p. 460-468.

132. Wua, X.F. and Y. Dzenisb, *Elasticity of planar fiber networks.* J Appl Phys, 2005. **98**: p. 093501-9.

133. Breuls, R.G., et al., *Predicting local cell deformations in engineered tissue constructs: a multilevel finite element approach.* J Biomech Eng, 2002. **124**(2): p. 198-207.

134. Barnes, C.P., et al., *Nanofiber technology: designing the next generation of tissue engineering scaffolds.* Adv Drug Deliv Rev, 2007. **59**(14): p. 1413-33.

135. Murugan, R. and S. Ramakrishna, *Design strategies of tissue engineering scaffolds with controlled fiber orientation.* Tissue Eng, 2007. **13**(8): p. 1845-66.

136. Mauck, R.L., et al., *Engineering on the straight and narrow: the mechanics of nanofibrous assemblies for fiber-reinforced tissue regeneration.* Tissue Eng Part B Rev, 2009. **15**(2): p. 171-193.

137. Nagatomi, J., et al., *Quantification of bladder smooth muscle orientation in normal and spinal cord injured rats.* Ann Biomed Eng, 2005. **33**(8): p. 1078-89.

138. Karlon, W.J., et al., *Automated measurement of myofiber disarray in transgenic mice with ventricular expression of ras.* Anat Rec, 1998. **252**(4): p. 612-25.

139. Sacks, M.S., *Small-angle light scattering methods for soft connective tissue structural analysis*, in *Encyclopedia of Biomaterials and Biomedical Engineering*, Gary L. Bowlin and G. Wnek, Editors. 2004, Informa Healthcare

140. Tan, W., et al., *Structural and functional optical imaging of three-dimensional engineered tissue development.* Tissue Eng, 2004. **10**(11-12): p. 1747-1756.

141. Chauduri, B.B., K. Pulak, and S. Nirupam, *Detection and gradation of oriented texture.* Pattern Recogn Lett, 1993. **14**(2): p. 147-153.

142. Pourdeyhimi, B., R. Ramanathan, and R. Dent, *Measuring fiber orientation in nonwovens: part I: simulation.* Text Res J, 1996. **66**(11): p. 713-722.

143. Pourdeyhimi, B., R. Ramanathan, and R. Dent, *Measuring fiber orientation in nonwovens: part II: direct tracking.* Text Res J, 1996. **66**(12): p. 747-753.

144. Pourdehyhimi, B., et al., *Measuring Fiber Orientation in Nonwovens Part V: Real Webs.* Textile Research Journal, 1999. **69**(3): p. 185-192.

145. Ayres, C., et al., *Modulation of anisotropy in electrospun tissue-engineering scaffolds: analysis of fiber alignment by the fast Fourier transform.* Biomaterials, 2006. **27**(32): p. 5524-5534.

146. Ayres, C.E., et al., *Incremental changes in anisotropy induce incremental changes in the material properties of electrospun scaffolds.* Acta Biomater, 2007. **3**(5): p. 651-61.

147. Ayres, C.E., et al., *Measuring fiber alignment in electrospun scaffolds: a user's guide to the 2D fast Fourier transform approach.* J Biomat Sci-Polym E 2008. **19**(5): p. 603-621.

148. Stella, J.A., et al., *On the biomechanical function of scaffolds for engineering load-bearing soft tissues.* Acta Biomater, 2010. **In Press**.

149. Hashizume, R., et al., *Morphological and mechanical characteristics of the reconstructed rat abdominal wall following use of a wet electrospun biodegradable polyurethane elastomer scaffold.* Biomaterials, 2010. **31**(12): p. 3253-3265.

150. Guan, J., et al., *Synthesis, characterization, and cytocompatibility of elastomeric, biodegradable poly(ester-urethane)ureas based on poly(caprolactone) and putrescine.* J Biomed Mater Res, 2002. **61**(3): p. 493-503.

151. Guan, J., et al., *Biodegradable poly(ether ester urethane)urea elastomers based on poly(ether ester) triblock copolymers and putrescine: synthesis, characterization and cytocompatibility.* Biomaterials, 2004. **25**(1): p. 85-96.

152. Nirmalanandhan, V.S., et al., *Effect of length of the engineered tendon construct on its structure-function relationships in culture.* J Biomech, 2007. **40**(11): p. 2523-9.

153. Williams, C., et al., *Altered structural and mechanical properties in decellularized rabbit carotid arteries.* Acta Biomater, 2009. **5**(4): p. 993-1005.

154. Jones, J.R., et al., *Non-destructive quantitative 3D analysis for the optimisation of tissue scaffolds.* Biomaterials, 2007. **28**(7): p. 1404-1413.

155. Gonzalez, R.C., R.E. Woods, and S.L. Eddins, *Digital image processing using MATLAB.* 2nd Ed ed, ed. Pearson. 2001, Upper Saddle River, New Jersey. : Prentice Hall.

156. Ziabari, M., V. Mottaghitalab, and A.K. Haghi, *Application of direct tracking method for measuring electrospun nanofiber diameter.* Braz J Chem Eng, 2009. **26**(1): p. 53-62.

157. Otsu, N., A, *Threshold selection method from gray-level histograms.* IEEE T Syst Man Cyb 1979. **9**(1): p. 62-66.

158. Lam, L., S.-W. Lee, and C.Y. Suen, *Thinning methodologies- a comprehensive survey.* IEEE Trans. Pattern Anal. Mach. Intell., 1992. **14**(9): p. 869-885.

159. Anthony, P.L., et al., *Observation of parity nonconservation in moller scattering.* Phys Rev Lett, 2004. **92**(18): p. 181602.

160. Pourdeyhimi, B. and R. Dent, *Measuring fiber diameter distribution in nonwovens.* Text Res J, 1999. **69**(4): p. 233-236.

161. Bugao Xu and Y.-L. Ting, *Measuring structural characteristics of fiber segments in nonwoven fabrics.* Text Res J, 1995. **65**(1): p. 41-48.

162. Sacks, M.S. and C.J. Chuong, *Characterization of collagen fiber architecture in the canine central tendon.* J Biomech Eng, 1992. **114**(2): p. 183-190.

163. Sacks, M.S., D.B. Smith, and E.D. Hiester, *A small angle light scattering device for planar connective tissue microstructural analysis.* Ann Biomed Eng, 1997. **25**(4): p. 678-89.

164. Sacks, M.S., C.J. Chuong, and R. More, *Collagen fiber architecture of bovine pericardium.* Asaio J, 1994. **40**(3): p. 632-637.

165. Stankus, J.J., J. Guan, and W.R. Wagner, *Fabrication of biodegradable elastomeric scaffolds with sub-micron morphologies.* J Biomed Mater Res-A, 2004. **70**(4): p. 603-614.

166. Engelmayr, G.C., et al., *Accordion-like honeycombs for tissue engineering of cardiac anisotropy.* Nat Mater, 2008. **7**(12): p. 1003-1010.

167. Ji, Y., et al., *Electrospun three-dimensional hyaluronic acid nanofibrous scaffolds.* Biomaterials, 2006. **27**(20): p. 3782-3792.

168. Gitman, I.M., H. Askes, and L.J. Sluys, *Representative volume: existence and size determination.* Eng Fract Mech, 2007. **74**(16): p. 2518-2534.

169. Liu, C., *On the minimum size of representative volume element: an experimental investigation.* Exp Mech, 2005. **45**(3): p. 238-243.

170. Romero, P. and E. Masad, *Relationship between the representative volume element and mechanical properties of asphalt concrete.* J Mater Civil Eng, 2001. **13**(1): p. 77-84.

171. Suresh L. Shenoya, W.D.B., Harry L. Frisch, Gary E. Wnek, *Role of chain entanglements on fiber formation during electrospinning of*

polymer solutions: good solvent, non-specific polymer–polymer

interaction limit. polymer, 2005. **46**: p. 3372-3384.

172. Fridrikh, S.V., et al., *Controlling the fiber diameter during electrospinning.* Physical Review Letters, 2003. **90**(14): p. -.

173. Xu, C.Y., et al., *Aligned biodegradable nanofibrous structure: a potential scaffold for blood vessel engineering.* Biomaterials, 2004. **25**(5): p. 877-86.

174. Thomas, V., et al., *Mechano-morphological studies of aligned nanofibrous scaffolds of polycaprolactone fabricated by electrospinning.* J Biomater Sci Polym Ed, 2006. **17**(9): p. 969-84.

175. Wan-Ju Li, R.L.M., James A. Cooper, James A. Cooper, Rocky S. Tuan, *Engineering controllable anisotropy in electrospun biodegradable nanofibrous scaffolds for musculoskeletal tissue engineering.* J Biomech., 2007. **40**(8): p. 1686-1693.

176. Meghana V. Kakade, S.G., Kenncorwin Gardner, Keun Hyung Lee, D. Bruce Chase, and John F. Rabolt*,, *Electric Field Induced Orientation of Polymer Chains in Macroscopically Aligned Electrospun Polymer Nanofibers.* J. AM. CHEM. SOC, 2007. **129**: p. 2777-2782.

177. Zhenwen Ding, A.S., Babak Ziaie, *Selective Nanofiber Deposition through Field-Enhanced Electrospinning.* Langmuir, 2009. **25**(17): p. 9648–9652.

178. Dan Li, G.O., Jesse T. McCann, Younan Xia, *Collecting Electrospun Nanofibers with Patterned Electrodes.* Nano Letters, 2005. **5**(5): p. 913-916.

179. D'Amore, A., et al., *Characterization of the Complete Fiber Network Topology of Planar Fibrous Tissues and Scaffolds* Biomaterials, 2010. **In Press**.

180. Ray JL, L.R., Herbert JM, Benson M., *Isolation of vascular smooth muscle cells from a single murine aorta.* Methods Cell Sci, 2002. **23**: p. 185-188.

181. Ryotaro Hashizume, K.L.F., Yi Hong, Nicholas J. Amoroso, Kimimasa Tobita, and B.B.K. Toshio Miki, Michael S. Sacks , William R. Wagner, *Morphological and mechanical characteristics of the reconstructed rat abdominal wall following use of a wet electrospun biodegradable polyurethane elastomer scaffold.* Biomaterials, 2010. **31**: p. 3253-3265.

182. Sacks, M.S. and C.J. Chuong, *Characterization of Collagen Fiber Architecture in the Canine Central Tendon.* Journal of Biomechanical Engineering, 1992. **114**: p. 183-190.

183. Sacks, M.S., *Biaxial mechanical evaluation of planar biological materials.* Journal of Elasticity, 2000. **61**: p. 199-246.

184. Billiar, K.L. and M.S. Sacks, *Biaxial mechanical properties of the natural and glutaraldehyde treated aortic valve cusp--Part I: Experimental results.* J Biomech Eng, 2000. **122**(1): p. 23-30.

185. Grashow, J.S., et al., *Planar biaxial creep and stress relaxation of the mitral valve anterior leaflet.* Ann Biomed Eng, 2006. **34**(10): p. 1509-18.

186. Chauduri, B.B., K. Pulak, and S. Nirupam, *Detection and gradation of oriented texture.* Pattern Recognition Letters, 1993. **14**: p. 6.

187. Pourdeyhimi, B., R. Ramanathan, and R. Dent, *Measuring Fiber Orientation in Nonwovens: Part I: Simulation.* Textile Research Journal, 1996. **66**(11): p. 713-722.

188. Pourdeyhimi, B., R. Ramanathan, and R. Dent, *Measuring Fiber Orientation in Nonwovens: Part II: Direct Tracking.* Textile Research Journal, 1996. **66**(12): p. 747-753.

189. Ayres, C.E., et al., *Measuring fiber alignment in electrospun scaffolds: a user's guide to the 2D fast Fourier transform approach.* Journal of Biomaterials Science, Polymer Edition, 2008. **19**: p. 603-621.

190. Chandran, P.L. and V.H. Barocas, *Deterministic Material-Based Averaging Theory Model of Collagen Gel Micromechanics.* Journal of Biomechanical Engineering, 2007. **129**(2): p. 137-147.

191. Stylianopoulos, T. and V.H. Barocas, *Multiscale, Structure-Based Modeling for the Elastic Mechanical Behavior of Arterial Walls.* Journal of Biomechanical Engineering, 2007. **129**(4): p. 611-618.

192. Burd, H.J., *A structural constitutive model for the human lens capsule.* Biomech Model Mechanobiol, 2008.

193. Wang, C.W., L. Berhan, and A.M. Sastry, *Structure, Mechanics and Failure of Stochastic Fibrous Networks: Part I—Microscale Considerations.* Journal of Engineering Materials and Technology, 2000. **122**(4): p. 450-459.

194. Wang, C.W. and A.M. Sastry, *Structure, mechanics and failure of stochastic fibrous networks : Part II-Network simulations and application.* Journal of Engineering Materials and Technology, 2000. **122**(4): p. 460-468.

195. Wua, X.F. and Y. Dzenisb, *Elasticity of planar fiber networks.* A. Journal of Applied Physics, 2005. **98**: p. 093501-9.

196. Ziabari, M., V. Mottaghitalab, and A.K. Haghi, *Application of direct tracking method for measuring electrospun nanofiber diameter*. Brazilian Journal of Chemical Engineering, 2009. **26**: p. 53-62.

197. Otsu, N., A. , *Threshold Selection Method from Gray-Level Histograms.* IEEE Transactions on Systems, Man, and Cybernetics, 1979. **9**.

198. Lam, L., S.-W. Lee, and C.Y. Suen, *Thinning Methodologies-A Comprehensive Survey.* IEEE Trans. Pattern Anal. Mach. Intell., 1992. **14**(9): p. 869-885.

199. Pourdeyhimi, B. and R. Dent, *Measuring Fiber Diameter Distribution in Nonwovens.* Textile Research Journal, 1999. **69**(4): p. 233-236.

200. Murugan, R. and S. Ramakrishna, *Design Strategies of Tissue Engineering Scaffolds with Controlled Fiber Orientation.* Tissue Eng, 2007.

201. Stankus, J.J., Soletti, L., Fujimoto, K., Hong, Y., Vorp, D.A., Wagner, W.R., *Fabrication of cell microintegrated blood vessel constructs through electrohydrodynamic atomization.* Biomaterials, 2007. **28**(17): p. 2738-46.