ABSTRACT

DIXIT, NIKHIL NANDKUMAR. Morphological changes to the Glenohumeral Musculoskeletal system following Brachial Plexus Birth Injury (BPBI). (Under the direction of Dr. Katherine Saul).

This dissertation evaluates musculoskeletal abnormalities occurring due to preganglionic and postganglionic BPBI. BPBI paralyzes muscles responsible for shoulder and elbow movement with severe bone deformities at the glenohumeral joint. Currently, there is a lack of understanding about the influence of nerve injury location on muscle changes and osseous deformities.

Aim 1 was to understand the contributions of altered loading due to static loading and muscle growth restriction to joint deformity after BPBI. This study used a computational framework to examine the mechanical factors contributing to changes in bone growth and morphology. Simulations predicted that static loading is primarily responsible for joint deformation. Under dynamic loading, glenoid version angle (GVA), glenoid inclination angle (GIA), and glenoid radius of curvature (GRC) (-1.3°, 38.2°, 2.5mm) were similar to the baseline values (-1.8°, -38°, 2.1mm) but increased in magnitude in the static case (5.2°, -48°, 3.5mm). More severe joint deformations were observed in GIA and GRC when muscle growth was restricted (GVA: 3.6°, GIA: -55°, GRC: 4.0mm).

Aim 2 sought to identify the changes to underlying muscle structure following preganglionic and postganglionic BPBI. Twenty-six Sprague Dawley rat pups received preganglionic or postganglionic neurectomy on a single limb at postnatal days 3-4. Range of motion was more restricted in the postganglionic group (4-weeks: -40.7 ± 21.8%; 8-weeks: -29.6 ± 49.1%) compared to preganglionic (4-weeks: 21.1 ± 41.0%; 8-weeks: 26.6 ± 45.8%), but muscles following preganglionic injury were more degenerated relative to the unaffected limb than following postganglionic injury. Compared to unaffected limb, optimal muscle length was shorter in the affected limb remarkably more for the preganglionic group (-18.2 ± 9.2%) than the

postganglionic group (-5.1 ± 6.2%), and muscle mass was on average lower in the affected limb for the preganglionic group (-57.2 ± 24.1%) compared to the postganglionic group (-28.1 ± 17.7%).

While the extent of bone deformity after postganglionic injury is well characterized, the nature of glenohumeral deformation following preganglionic BPBI is unclear. Aim 3 characterized bone changes in the same animals studied in Aim 2. MicroCT scans of the joint showed after postganglionic injury, the glenoid was significantly more declined on the affected limb compared to the unaffected limb (-17.7 ± 16.9°, $p = 0.021$). Compared to the preganglionic group, the affected shoulder in the postganglionic group exhibited significantly higher declination ($p = 0.0010$) and higher glenoid radius of curvature ($p = 0.0070$). After preganglionic injury, humeral head radius of curvature, thickness, and width were significantly smaller on the affected side compared to the unaffected side (curvature: -0.5 ± 0.5 mm, $p = 0.0085$; thickness: -0.5 ± 0.5 mm, $p = 0.011$; width: -0.4 ± 0.4 mm, $p = 0.0076$) and tended to be linearly correlatedinterlinked.

Finally, Aim 4 investigated whether changes to muscle mass and optimal muscle length over time following BPBI provide a mechanically-driven explanation for observed differences in postural and bone deformity between preganglionic and postganglionic BPBI. A computational framework was developed to simulate bone growth in response over time. The simulations predicted the net glenohumeral joint loads in the postganglionic injury case were nearly 11% greater than the preganglionic. Bone deformations were more severe in the postganglionic case, with the glenoid more declined (pre: -43.8°, post: -48.9°), flatter with higher radius of curvature (pre:3.0mm, post:3.7mm), and retroverted (pre: -13.7°, post: -16.1°) than the preganglionic case.

This dissertation provides new information about the influential mechanical factors driving glenohumeral joint deformity. The work found that the reduced muscle mass and restriction in longitudinal muscle growth were more severe due to preganglionic injuries than postganglionic injuries. However, the restriction in ROM and the osseous deformities were more severe in postganglionic injuries than preganglionic injuries. The alterations in bone morphology can be predicted from the mechanical contributions of the altered muscle structure.

Morphological changes to the Glenohumeral Musculoskeletal system following Brachial Plexus Birth Injury (BPBI)

by
Nikhil Nandkumar Dixit

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Mechanical Engineering

Raleigh, North Carolina
2019

APPROVED BY:

Dr. Katherine Saul Committee Chair	Dr. Jacqueline Cole
Dr. Scott Ferguson	Dr. Hsiao Ying Shadow Huang

DEDICATION

To my parents, Nandkumar W. Dixit and Varsha N. Dixit for their continual love and support over the years.

BIOGRAPHY

Nikhil N. Dixit was born in Chandrapur, Maharashtra in India and grew up in Nagpur, Maharashtra. He graduated from YCCE, Nagpur in 2011 with a B.E. in Mechanical Engineering. Subsequently, he started his Masters studies from IIT Kharagpur and graduated with M.Tech. in Aerospace Engineering in 2013. Upon graduation, he accepted a role of an Engineer in Eaton Technologies, Pune, India. He worked there for several years and left that position to pursue Doctoral studies from the Department of Mechanical and Aerospace Engineering at North Carolina State University in Fall 2015. He will complete his academic requirements for a Ph.D. in Mechanical Engineering in May 2019.

ACKNOWLEDGMENTS

First and foremost, I would like to thank Dr. Katherine Saul for her continual guidance and support throughout my doctoral studies. I have learned so much from you in the last four years to become the researcher I am today. I couldn't have wished for any other advisor than you.

I would like to thank my committee members for critically reviewing my work and helping me enhance it. Thank you very much, Dr. Jacqueline Cole for sharing your knowledge and experience in guiding my research. You are the other person from whom I have learnt a lot during my studies. I thank Dr. Scott Ferguson for your analytical take at my work. I have to say that your course on Engineering Design Optimization was one of the best courses I have taken. I thank Dr. Hsiao Ying Shadow Huang for evaluating my work. I especially enjoyed our discussions on finite element analysis.

I thank Dr. Matthew Fisher. It was great to work with you and discuss how we can improve the growth models. The was fortunate to work with exceptional surgeons like Dr. Roger Cornwall, Dr. Kerry Danelson and Dr. Tom Smith in performing preganglionic and postganglionic neurectomies on neonatal rats. It was great to work with you and help you out with the surgeries. I would also like to thank all my funding sources from the National Center for Simulation in Rehabilitation Research (NIH2 P2CHD065690-06) and National Institutes of Health (NIH 1 R21 HD088893-01). I thank Schneider Electric for giving me the opportunity to test my work for real world mechanical engineering applications.

A big thank you to all my lab mates. Without you, this research would always be incomplete. Daniel McFarland, it was a pleasure working with you. The conference trips were fun. And yes, thanks for the countless rides. Emily McCain, I loved the discussions we had on so many topics other than studies. At least, now I can have boxes on my figures wherever I want. Michael

Poppo, you definitely kept the lab atmosphere a fun. Matt Berno, you are definitely one of a kind. It was great to know you. Carolyn McCormick, I can never forget sharing the experiences we had with handling all those rats for the first time. I would also like to thank Emily Fawcett, Danielle Howe, Maggie Tamburro, Eric Warren, Kristen Lo, and Eric Levingston for all your help.

I am fortunate to have the company of my friends who have stayed in touch all the way from India during all these years. Thanks a lot, Anshuman Sinha, Ameya Padhye, Kunal Moharil, Shantanu Ligde, and Sumedh Bhalerao. You guys are awesome.

I could never thank enough to my family members with such limited space. Thank you Aai and Baba for putting your faith in me. Without your support, and blessing, I would not have not even begun my doctoral studies. We took great risk and it paid off for sure. My brother Omkar Dixit, thanks for your lame jokes which kept me cheerful throughout the day. Thank you Swarangi Joshi, my girlfriend then and fiancé now for always being there with me with all your love and support. And lastly, thanks a lot to Spud, my dog, who wanted to make his contributions to my research by sitting on my laptop keyboard when I was writing this dissertation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: Background

Brachial plexus birth injury

The brachial plexus is a network of nerves formed by the lower four cervical nerves and first thoracic nerve (C5, C6, C7, C8, and T1). This plexus extends from the spinal cord, through the cervicoaxillary canal in the neck, over the first rib, and into the armpit. It supplies afferent and efferent nerve fibers to the chest, shoulder, arm and hand. The brachial plexus is divided into five roots, three trunks, six divisions (three anterior and three posterior), three cords, and five branches.

Brachial plexus structure:

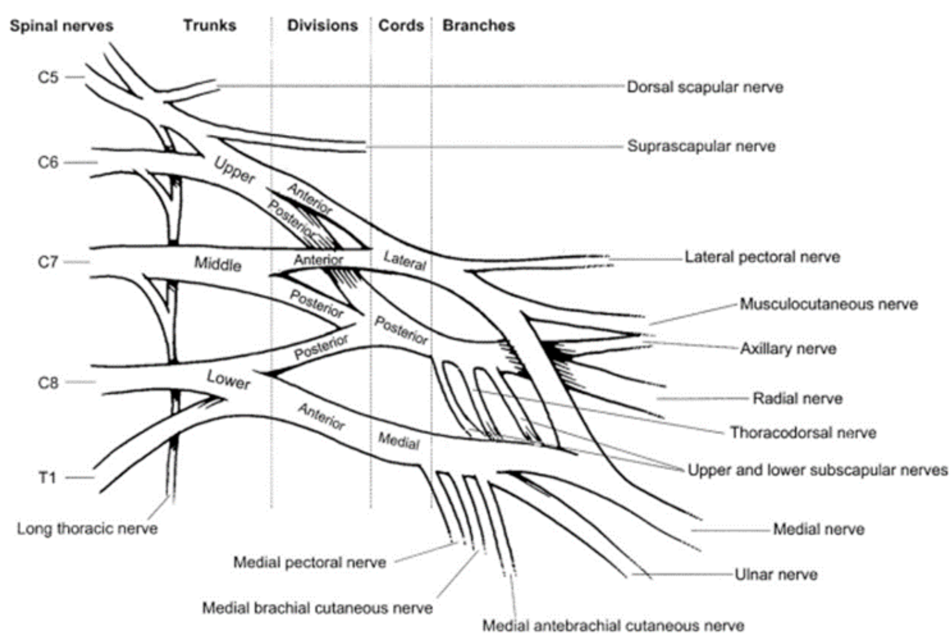

Figure 1.1: Anatomical illustrations of the brachial plexus with areas of roots, trunks, divisions, and cords marked.

The five roots are the five anterior rami of the spinal nerves. The brachial plexus emerges at the C5, C6, C7, C8, and T1 levels (Figure 1.1). C5 and C6 merge to establish the upper trunk, C7 continuously forms the middle trunk, and C8 and T1 merge to establish the lower trunk. Each trunk then splits in two, to form six divisions. These six divisions regroup to become three cords.

The dorsal scapular nerve derives directly from the superior trunk and innervates the rhomboid muscles, which retract the scapula. The subclavian nerve originates in both C5 and C6 and innervates the subclavius, a muscle that contributes to lifting the first ribs during respiration. The long thoracic nerve arises from C5, C6, and C7. This nerve innervates the serratus anterior, which draws the scapula laterally and is the prime mover in all forward-reaching and pushing actions. The nerves from the brachial plexus are responsible for nerve supply of the whole arm. A detailed table of the nerves, roots, and activated muscles is shown in Table 1.1.

Table 1.1: Brachial plexus nerves and their activation patterns (Essential Clinical Anatomy 3rd. edition)

Nerve	Roots	Muscles
Dorsal scapular nerve	C4, C5	Rhomboid muscles and levator scapulae
Long thoracic nerve	C5, C6, C7	Serratus anterior
Nerve to the subclavius	C5, C6	Subclavius muscle
Suprascapular nerve	C5, C6	Supraspinatus and infraspinatus
Lateral pectoral nerve	C5, C6, C7	Pectoralis major and pectoralis minor (by communicating with the medial pectoral nerve)
Musculocutaneous nerve	C5, C6, C7	Coracobrachialis, brachialis and biceps brachii
Upper subscapular nerve	C5, C6	Subscapularis (upper part)
Thoracodorsal nerve(middle subscapular nerve)	C6, C7, C8	Latissimus dorsi
Lower subscapular nerve	C5, C6	Subscapularis (lower part) and teres major
Axillary nerve	C5, C6	Anterior branch: deltoid and a small area of overlying skin. Posterior branch: teres minor and deltoid muscles posterior branch: teres minor and deltoid muscles
Radial nerve	C5, C6, C7, C8, T1	Triceps brachii, supinator, anconeus, the extensor muscles of the forearm, and brachioradialis
Medial pectoral nerve	C8, T1	Pectoralis major and pectoralis minor
Medial root of the median nerve	C8, T1	All of the flexors in the forearm except flexor carpi ulnaris and that part of flexor digitorum profundus that supplies the 4th and 5th digits
Ulnar nerve	C8, T1	Flexor carpi ulnaris, the medial two bellies of flexor digitorum profundus, the intrinsic hand muscles, except the thenar muscles and the two lateral lumbricals of the hand which are served by the median nerve

Brachial plexus birth injury causes and effects

Brachial plexus injuries in newborns usually occur during a difficult delivery, such as with a large baby, a breech presentation, or a prolonged labor. They may also happen when a birth becomes complicated and the person assisting the delivery must deliver the baby quickly and exert some force to pull the baby from the birth canal. If one side of the baby's neck is stretched, the nerves may also be stretched, and injury may result. Such an injury typically has symptoms including an internally rotated and adducted arm posture (Al-Qattan, 2003). In adults, brachial plexus injuries typically occur due to accidents, especially motor vehicle accidents. Injuries to the lower trunk may be specifically due to forceful anterior compression of the arms in situations like compression from a car seatbelt during an accident (Soldado et al., 2016).

Brachial plexus injuries are often categorized according to the type of the nerve injury and the pattern of the nerves involved. There are 4 types of nerve injuries: stretch (neurapraxia), neuroma, rupture (postganglionic), and avulsion (preganglionic). A stretch injury is caused when the nerve has been stretched but not torn. The injury occurs distal to the junction with the spinal cord where the nerve originates. This is the most common form of the injury and usually the affected nerves recover on their own within 3 months postnatal (Zafeiriou and Psychogiou, 2008). A neuroma occurs when the nerve has tried to heal, but a scar tissue has formed and presses against the injured nerve or interferes with nerve function. Treating a neuroma may require surgical treatment with nerve reconstruction and/or tendon transfers. Nerve rupture causes a tear in the midsubstance of a nerve. This type of injury does not typically heal on its own and requires surgical intervention such as nerve graft or nerve transfer. Avulsion injury causes a tear in the nerve at the spinal cord. This severe injury is less common and may require the nerve to be surgically replaced (nerve transfer) if function is to be restored. BPBI presentation in patients can vary significantly according to the location of the nerve injury relative to the nerve root ganglion.

While many children who experience BPBI spontaneously recover nerve and muscle function, 20% to 30% of affected children do not have total neurological recovery (Pondaag et al., 2004). Up to 33% of children affected with BPBI sustain permanent postural and osseous deformities (Pearl, Michael L., 2009), which can have lifelong consequences for upper limb function and quality of life (Bae et al., 2008). Damage to the C5-C6 nerves typically occurring in BPBI results in complete denervation of subclavius, supraspinatus, infraspinatus, subscapularis, teres minor, teres major, deltoid and partial denervation of rhomboid muscles, levator scapulae, serratus anterior, pectoralis major and minor, coracobrachialis, brachialis, biceps brachii, latissimus dorsi, triceps brachii, supinator, anconeus, extensor muscles of forearms, and brachioradialis. When these muscles are affected, internal rotation and abduction contractures occur at the shoulder (Al-Qattan, 2003). This restriction in the shoulder range of motion (ROM) and the muscle paralysis results in disuse of the shoulder making it more static (Sheehan et al., 2014). Further, presentation in patients can vary significantly according to the location of the nerve injury relative to the nerve root ganglion. Nerve root ruptures which occur distal to the ganglion (postganglionic) frequently result in shoulder internal rotation and elbow flexion contractures (Waters, Peter, 2005). Contractures and restriction in ROM in these patients significantly correlate with osseous deformities at the glenohumeral joint, including retroversion, declination and flattening of the glenoid fossa (Bhardwaj et al., 2013; Kozin, 2004; Pearl, Michael L. and Edgerton, 1998). In contrast, avulsion injuries proximal to the ganglion (preganglionic) typically result in paralysis without contractures at the shoulder and elbow or significant osseous deformity (Al-Qattan, 2003).

BPBI results in abnormal bone morphology at the glenohumeral joint, including higher retroversion and declination(Eismann et al., 2015; van Gelein Vitringa, Valerie M et al., 2013), as well as to the humeral head with increased flattening (Sibinski et al., 2010). Glenohumeral dysplasia is a 3D deformity and not limited to the axial plane; thus, considerations of dysplasia in other anatomical planes may have implications for evaluation and treatment planning (Eismann et al., 2016). The orientation and positioning of the glenohumeral joint is also affected, with posterior humeral head subluxation (van Gelein Vitringa, Valerie M et al., 2013). Further, the variable position of the scapula on the thorax can result in confounding interpretations of glenohumeral dysplasia, and thus requires more attention in assessing the severity of the deformity (Stein et al., 2017). These severe osseous deformities due to BPBI affect function of the glenohumeral joint, making it unstable (Di Mascio et al., 2011; Kambhampati et al., 2006). 3D MRI techniques have shown that patients with persistent shoulder weakness following BPBI have glenohumeral abduction deformities with severe contractures (Eismann et al., 2015). The origin of the postural and osseous deformities is incompletely understood (Nikolaou et al., 2011). Some studies using magnetic resonance imaging (MRI) in patients with BPBI found atrophy in internal and external rotator muscles(Waters, Peter M. et al., 2009) or internal rotators alone (Hogendoorn et al., 2010) and suggest that a muscle imbalance at the shoulder could cause the contracture. Contractures and osseous deformities in patients have been seen to coexist (Al Qattan 2003) and therefore development of contractures could result in osseous deformities. Animal studies in mice have found that functional shortening of the impaired denervated muscles after BPBI may contribute towards generation of contractures (Nikolaou et al. 2011). Muscle spindle degeneration and loss of ErB signaling was observed in biceps in cases of postganglionic injuries but not in preganglionic injury cases (Nikolaou et al., 2015). Studies in postganglionic BPBI affected rats have shown muscle mass loss and restricted longitudinal muscle growth, potentially responsible for the development of contractures (Crouch et al., 2015).

BPBI treatment strategies and challenges

The primary aim of treatment for BPBI is restoration of abduction and external rotation function. How this is achieved depends upon the type of the injury, degree of deformity of the glenohumeral joint over time, and which functioning muscles are available around the shoulder for transfer.

Nonsurgical treatment:

The first step in treatment management is non-operative. Because most newborns with BPBI recover on their own (Zafeiriou and Psychogiou, 2008), the doctor will re-examine the child frequently to see if the nerves are recovering. Nerves grow and recover very slowly; it may take up to 2 years for a complete recovery. As the shoulder develops an internal rotation contracture, it may respond to physiotherapy and splinting, or botulinum toxin injection (Nath and Paizi, 2007). Physical therapy is the main treatment method for BPBI (Jellicose and Parsons, 2008). Because a baby cannot move the affected arm independently, an active role of parents in keeping the joints limber and the functioning muscles fit is critical. Daily physical therapy and range-of-motion exercises, done as often as possible during the day, begin when a baby is about 3 weeks old. The exercises will maintain the ROM in the shoulder, elbow, wrist, and hand. This prevents the joint from becoming permanently stiff, a condition called joint contracture.

Surgical intervention

If non-operative procedures do not result in relatively rapid recovery, several surgical intervention procedures can be considered. During this period of rapid musculoskeletal development in child's life, inability to recover from the injury quickly can lead to permanent musculoskeletal deformities, and therefore timing and type of the surgical procedure play a key role towards achieving recovery.

If there is no change over the first 3 to 6 months, the clinician may suggest surgery on the nerves to improve the potential outcome. There are two types of nerve surgeries; nerve graft and nerve transfer. In nerve graft surgeries, depending upon the nerve injury, it may be possible to repair a rupture by "splicing" a donor nerve graft from another intact nerve site in the patient. In nerve transfer surgeries, it may be possible to restore some function in the arm by using an intact nerve from another nearby muscle as a donor. Ulnar and/or median nerve fascicle transfer is implemented for functional restoration of elbow flexion (Little et al., 2014). Typically nerve graft or transfer surgeries do not restore full, normal function, and are usually not effective for older infants (Pichon and Carlioz, 1979).

It may take several months or years for nerves to recover, and hence many children with brachial plexus injuries continue to have some weakness in the shoulder, arm, or hand (Sever, 1920). Other surgical procedures are sometimes performed subsequently to improve function in the longer term. For example, tendon transfer surgeries can be employed to improve shoulder abduction and external rotation e.g. open or arthroscopic subscapularis release, anterior or arthroscopic glenohumeral capsulotomy, and tendon transfer (Pearl, Michael et al., 2003). For more severe deformities, where the shoulder is deemed unreconstructable, a humeral derotation osteotomy may be used to correct a severely internally rotated humerus into a more functional position (Zafeiriou and Psychogiou, 2008).

Most surgeons agree that infants presenting total plexus palsies (affecting all roots C5, C6, C7, C8, and T1) should undergo microsurgical nerve reconstruction at 3 months of age

(Kirjavainen et al., 2007). However, there is no consensus among surgeons for the patients with partial BPBI (typically C5-C6 injury), with some suggesting intervention at 3 months (Haerle and Gilbert, 2004; Kirjavainen et al., 2007) , some between 4 to 6 months (Waters, 1999; Al-Qattan et al., 2000), and some suggesting to wait at least 9 months to maximize the possibility of spontaneous recovery before indicating microsurgical treatment (Clarke and Curtis, 1995).

Muscle anatomy and physiology

Perinatal nerve injuries can alter the muscle architectural properties affecting their force development capacity (Crouch et al., 2015). To understand the potential contributions of altered muscle to glenohumeral loading, it is critical to examine the anatomy of the shoulder and how muscle architecture relates to muscle force production.

The stability of the glenohumeral joint is crucial for effective shoulder function. Failure to attain joint stability can lead to dislocation, impingement, and rotator cuff injuries (Hawkins and Janda, 1996). Joint stability is controlled by rotator cuff muscles, labrum, ligaments, and the biceps tendon. The rotator cuff consists of subscapularis, supraspinatus, infraspinatus, and teres minor (Figure 1.2). As a group, the rotator cuff keeps the head of the humerus in the small glenoid fossa of the scapula in order to provide stable movement through the ROM and avoid mechanical obstruction. The labrum is a fibrocartilaginous ridge surrounding the glenoid cavity. It deepens the cavity and creates a seal with the head of humerus, reducing the risk of dislocation. The glenohumeral joint is primarily supported by 5 ligaments; superior glenohumeral ligament, middle glenohumeral ligament, inferior glenohumeral ligament, coracohumeral ligament, transverse humeral ligament. The superior glenohumeral ligament arises from the glenoid and inserts on the anatomical neck of the humerus. This ligament limits the external rotation and inferior translation of the humeral head. The middle glenohumeral ligament also arises from the glenoid and inserts on the anatomical neck of the humerus limits and external rotation and anterior translation of the humeral head. The inferior glenohumeral ligament arises from the glenoid and inserts on the humerus just beyond the lesser tuberosity. The anterior portion of this ligament limits external rotation and superior and anterior translation of the humeral head whereas the posterior portion of the ligament limits internal rotation and anterior translation. The coracohumeral ligament runs laterally from the coracoid process to the humerus, covering the superior glenohumeral ligament and blending with the superior joint capusle and supraspinatus tendon superiorly. The ligament splits into anterior and posterior divisions by the biceps tendon. The anterior portion limits extension while the posterior portion limits flexion. Both divisions limit inferior and posterior translation of the humeral head. The ligament helps to support the weight of the resting arm against gravity. The transverse humeral ligament serves to keep the tendon of the long head of the biceps in the bicipital groove. Lastly, the biceps tendon acts as a minor humeral head depressor, thereby contributing to stability.

Figure 1.2: Rotator cuff muscles (adapted from Teachmeanatomy.info)

Muscle architecture and force production

The force a muscle produces depends, in part, on the length of the muscle fibers and the characteristic architecture of the muscle-tendon. The length-tension relationship (Figure 1.3) describes the relationship between the isometric force exerted by a muscle and its length and has active and passive contributions. The active length-tension relationship reflects the degree of overlap between the actin and myosin filaments in a sarcomere. Short lengths with too much overlap between actin and myosin results in suboptimal force development due to interference from actin overlap and the Z disks. Long lengths with too little overlap between actin and myosin leads to fewer binding sites available for the myosin head to form crossbridges and therefore effectively develop force. At the optimal overlap, maximal tension is developed. Sarcomeres at this length would effectively offer maximum force development capacity for the muscle at the corresponding muscle length. This muscle length is called optimal muscle length or resting length of the muscle. The passive length-tension relationship reflects the presence of elastic elements called titin within a sarcomere, which connects the Z line with the thick myosin filament. Titin, as well as other connective tissues within the muscle, stretch and produce force with increasing length. Thus, when the sarcomere stretches past its optimal length, the stretch in titin generates passive forces. Finally, the peak muscle force developed is proportional to the physiological cross-sectional area of a muscle, which is related to the total mass of the muscle. Thus, muscles with an altered optimal fiber length or total muscle mass will develop force differently within the context of an unaltered muscle-tendon in situ path.

$$PCSA = \frac{(M * \cos\theta)}{(\rho * L)} \qquad (3)$$

Where PCSA is the physiological cross-sectional area, M is the muscle mass, θ is the pennation angle, ρ is the muscle density, and L is the optimal muscle fiber length.

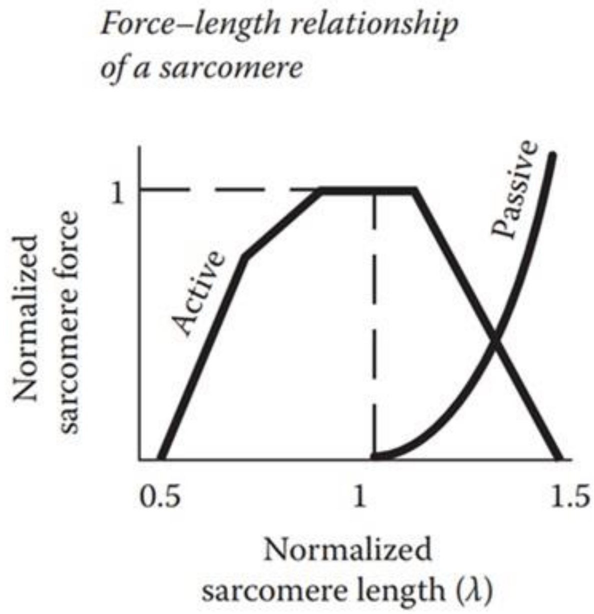

Figure 1.3: Muscle force-length relationship. The overall force developed by a muscle depends on length and is a combination of active and passive components

Multibody dynamic musculoskeletal modeling

Computational musculoskeletal analysis enables researchers to study musculoskeletal changes and the effects on human movement that cannot be studied experimentally. It can also be used to develop predictive models that can be employed to understand effects of pathological changes, test treatment strategies, and predict function.

Typically, the force-generating actuators in multibody dynamic musculoskeletal simulations represent the muscles. OpenSim, a commonly used modeling framework (Delp et al., 2007) , uses a Hill-type representative muscle model to capture the length- and velocity-dependent force-generating behavior of a muscle-tendon unit (Zajac, 1989). Individual muscles in the musculoskeletal model can be defined by assigning muscle parameters to characterize each muscle: maximum isometric force, optimal muscle fiber length, tendon slack length, maximum contraction velocity, and pennation angle. The muscle-tendon unit consists of three components;

a contractile element (CE), a parallel element (PE), and a series element (SE) (Figure 1.4). The resultant force generated by this unit is a function of the activation value, peak force-generating capacity, the normalized length of the muscle unit, and the normalized velocity of the muscle unit. The contractile element (CE) is an activation-dependent force producing element. The force developed by this element represents the active curve on the force-length and force-velocity relationships. The muscle activations in the model vary between 0 (unactivated) to 1 (fully activated). For a given activation value, the force development is highest at its optimal length (normalized length = 1). Isometric force generation occurs when the muscle contraction velocity is 0. As the muscle contracts, the length shortens and shortening velocity increases (becomes more negative) and the force development capacity of the muscle reduces. The passive muscle forces are represented by the parallel element. When the muscle elongates and its length becomes greater than the optimal fiber length, the parallel element generates a nonlinear passive force. Further, in series with the muscle element, the tendon force also increases nonlinearly upon muscle-tendon unit elongation. The tendon is more compliant for small elongations and stiffer for greater muscle elongations (Maganaris and Paul, 2002).

Previous computational muscle models of BPBI (Cheng et al., 2015; Crouch et al., 2014) have identified that the supraspinatus, infraspinatus, teres minor, subscapularis, deltoid, long head of the biceps, short head of the biceps, brachialis, brachialis, coracobrachialis, pectoralis major, and long head of the triceps being most capable of producing shoulder forces that may contribute to shoulder deformity following BPBI. These studies investigated possible scenarios following BPBI including potential active muscle imbalance at the glenohumeral joint and impaired longitudinal growth of affected muscles, and their effect on shoulder ROM and joint loading.

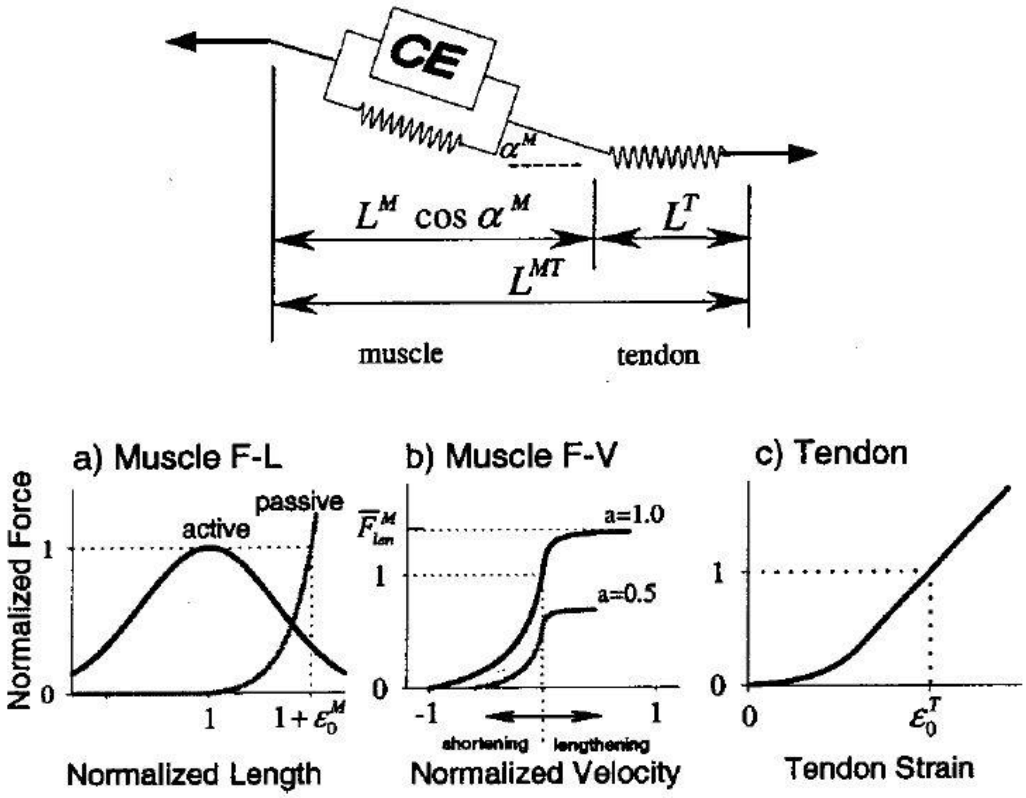

Figure 1.4: Hill-type muscle model used in OpenSim musculoskeletal dynamics

Simulations predicted that impaired longitudinal muscle growth restricted shoulder ROM and caused posteriorly-directed shoulder joint force consistent with clinically observed BPBI deformities; in contrast, active muscle imbalance was not predicted to lead to restricted ROM or loading consistent with glenoid deformities. These models were limited by including idealized changes to muscle architecture to investigate possible mechanical influence and do not directly predict changes in bone morphology. Thus, alternate modeling methods, including animal models or direct modeling of bone growth, are necessary modeling improvements needed to understand the musculoskeletal changes following preganglionic and postganglionic BPBI.

Bone development

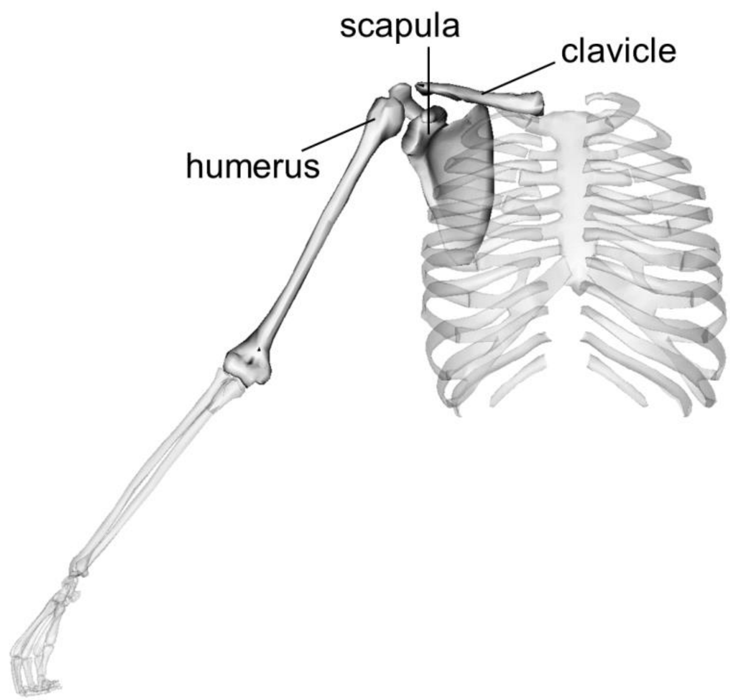

Figure 1.5: Glenohumeral joint skeletal anatomy consisting of humerus, scapula, and clavicle

BPBI affects development of glenohumeral joint morphology (Brochard et al., 2016; Sheehan et al., 2014). The glenohumeral joint (Figure 1.5) is composed of three bones: humerus, scapula, and clavicle. The humeral head fits into a shallow socket in the scapula, called the glenoid. The clavicle provides the structural support to the humeral head superiorly and articulates with the scapula and the sternum.

Normal bone growth is essential for proper musculoskeletal development and functioning. Bones serve a variety of mechanical functions in human body. Together the bones in the body form the skeleton, which provides a frame to keep the body supported. It also serves as an attachment point for skeletal muscles, tendons and ligaments, which function together to generate and transfer forces so that the movement of the individual body parts can be controlled in three-dimensional space. Bone morphological growth is dependent on its environment and is characterized by two primary drivers that combine to govern overall bone shape over time: direct biological and mechanical environment factors (Heegaard et al., 1999; Giorgi et al., 2014).

Biological growth

Biological growth captures the component of growth which is independent of the mechanical environment acting on the bones and is associated with overall increase in size with age until maturity. Overall biological growth is associated with the spatial arrangement of cell density. For a specific bone, the number of ossification centers and the chondrocyte density pattern defines the morphological changes over time. For long bones such as the humerus or femur, the equivalent biological growth has been described to be proportional to the chondrocyte density. The equation for local chondrocyte density along the long axis of a rudiment was calculated (Heegaard et al., 1999) by fitting a polynomial curve to the grey level distribution on a sagittal micrograph of a joint, where darker areas indicated higher chondrocyte density. The chondrocyte density is greater towards the ends of the rudiments and lower towards the diaphysis.

$$\dot{\varepsilon}_{\mathrm{b}} = \mathrm{C_d} = k(0.14 - 0.87\xi + 4.40\xi^2 - 2.66\,\xi^3) \qquad (1)$$

where $\dot{\boldsymbol{\varepsilon}}_{\mathbf{b}}$ is the biological growth rate, proportional by constant k to the chondrocyte density $\mathbf{C_d}$; ξ is the distance of each node along the proximal-distal axis from the distal end of the rudiment.

Mechanical growth

Mechanically-induced growth is proportional to stresses encountered within the bone tissue. The nature of the loading also influences the growth; static forces hinders the bone growth whereas dynamic loads promote the growth. This phenomenon is captured using the following equations for static and dynamic loads (Giorgi et al., 2015; Giorgi et al., 2014).

$$\dot{\varepsilon}_m = C_d \left(\frac{\sum_{i=1}^{N} \sigma_{hi}}{N} \right), \quad \text{for static loads}$$

$$\dot{\varepsilon}_m = -C_d \left(\frac{\sum_{i=1}^{N} \sigma_{hi}}{N} \right), \quad \text{for dynamic loads} \tag{2}$$

Finite element modeling of bone growth

Computational modeling of bone growth over time is an important tool to understand how altered loading, such as that experienced after BPBI, drives altered bone growth, and may allow a predictive scheme for morphological changes after surgical and other interventions. Finite element modeling divides the entire domain into very small elements which are connected through nodes, and changes in shape can be modeled using thermal expansion proportional to experimentally derived growth rates.

This technique has been previously applied to model bone growth in an idealized planar biomechanical model of the proximal interphalangeal (finger) joint (Heegaard et al., 1999); in this work, epiphyseal growth was simulated using a modified version of endochondral ossification theory (Carter, D. R. et al., 1987), in which growth and shape depends on the biological growth and mechanical growth. The model predicted the development of congruent surfaces within the joint region and was the first mechanobiological simulation of any aspect of prenatal joint development. This work has since been extended to a number of applications and using variations on the implementation for mechanical growth. For example, researchers have explored the effect of mechanical loading on endochondral ossification and growth plate progression (Shefelbine, S. J. et al., 2002), have predicted changes in the hip joint load direction due to cerebral palsy over time (Shefelbine, Sandra and Carter, 2004), and characterized the morphological effects on the femoral head from altered gait patterns as a result of cerebral palsy (Carriero et al., 2011). The latter two studies implemented mechanical growth two ways: proportional to the overall cyclic hydrostatic stress and octahedral shear stress (Shefelbine and Carter, 2004), and dependent on hydrostatic and shear stress (Carriero et al. 2011). Some recent work by this group has introduced an advanced biological growth model based on prior experimental findings (Heegaard et al. 1999) for long bones, to integrate a more spatially complex biological growth stimulus with the mechanical growth to predict bone morphogenesis (Giorgi et al., 2014). They used a thermal expansion analog to model the bone growth using multiple loading conditions and longer iteration times to obtain a range of realistic joint shapes. This has been applied to predict the effects of normal and abnormal loading on bone development following hip dysplasia (Giorgi et al., 2015), and the endochondral ossification of long bones to predict the changes in bone shape during the embryonic stages (Vaca-González et al., 2018).

Current FEA bone growth modeling work predicts morphological changes in 2D. However, BPBI results in 3D osseous deformation at the glenohumeral joint (Sheehan et al. 2014; Brochard et al. 2016). Currently there is limited knowledge regarding whether changes in the mechanical loading as a result of BPBI directly contributes towards development of osseous deformities at the glenohumeral joint. Musculoskeletal multibody dynamic simulations suggest that muscle force changes following BPBI would mechanically result in altered glenohumeral joint reaction loads, directing them more posterior than normal (Cheng et al. 2015). Such a change is consistent with the observed osseous deformities at the glenohumeral joint (Crouch et al. 2013); however, existing work allows merely for correlative relationships between predicted forces and observed deformity; a direct mechanical link to the development of bone deformity in three dimensions is not yet demonstrated.

Animal models of BPBI

Animal models provide an excellent, practical, and ethical framework by which one can replicate BPBI injury mechanisms and analyze their effects. Choosing a suitable animal model is critical to effectively replicate the injury scenario observed in humans. Rats develop more rapidly than humans, with every 10.5 days in a rat's life equivalent to approximately one human year based on musculoskeletal (Quinn, 2005). Thus, the length of a rat study is shorter than a human study for the same age span (Quinn, 2005). Examination of the shoulder and direct measurement of musculoskeletal properties that are difficult or impossible to measure from medical images or human subjects are possible following sacrifice. Rat and mouse shoulder anatomy is similar to that of humans (Norlin et al., 1994). Hence it is a relevant animal model to consider for BPBI studies.

Substantial work has examined whether murine models of BPBI capture the essential postural and osseous deformities characteristic of BPBI in human patients. Mouse (Kim et al., 2010) and rat (Li et al., 2008) models were initially developed to examine whether brachial plexus nerve injuries could be inflicted that result in presentation similar to that of humans. Since then, both mice and rat have been successfully used to study microstructural and macrostructural changes to muscles, bones, and posture post BPBI (Swan et al., 2014; Li et al., 2010; Crouch et al., 2015; Nikolaou et al., 2014; Nikolaou et al., 2015; Nikolaou et al., 2011).

Rats with simulated postganglionic BPBI after sacrifice at 4 weeks postnatal (Li et al. 2010) developed gross architectural glenohumeral joint distortions characterized by increased glenoid retroversion and inclination with a smaller humeral head. After sacrifice at 8 weeks, postganglionic BPBI rats (Crouch et al. 2015) exhibited significant muscle atrophy and impaired muscle longitudinal growth on the affected limb compared to the unaffected limb. The affected rats developed restriction in external rotation ROM, an indication of internal rotation contracture, with higher glenoid declination and retroversion; this is consistent with clinical outcomes (Brochard et al. 2016). Postganglionic BPBI in mice led to impaired longitudinal muscle growth of biceps and brachialis with associated elbow flexion contractures, and impaired longitudinal growth of subscapularis which may be associated with internal rotation contractures (Nikolaou et al., 2011). Biceps and brachialis muscles at 4 week postnatal revealed higher atrophy after to postganglionic BPBI and greater internal rotation and elbow flexion contracture when compared to preganglionic BPBI (Nikolaou et al. 2015). However, there is limited knowledge about the changes in all the other muscles crossing the shoulder that are affected due to postganglionic and preganglionic BPBI and its impact on altered glenohumeral joint loads and subsequent osseous deformities.

Summary, significance, and motivation of this study

Brachial plexus birth injury (BPBI) is the most common nerve injury among children (Foad et al., 2008). Abnormal growth patterns in the glenoid and humeral head are common sequelae, with retroversion and declination of the glenoid (Brochard et al., 2016; Hogendoorn et al., 2010) and flattened and smaller humeral heads (Sibinski et al., 2010; Kozin, 2004). These glenohumeral osseous deformities severely impact shoulder function, and lead to lifelong disability. The factors driving progression to deformity are unclear, but bone growth is known to depend on both biological and mechanical environment (Shefelbine and Carter, 2004). The mechanical environment of the shoulder is substantially altered postinjury. For example, glenohumeral deformity is known to directly correlate with contractures and restriction in shoulder ROM also observed post injury (Pearl and Edgerton, 1998; Kozin, 2004; Bhardwaj et al., 2013), in particular internal rotation and adduction contractures. The restriction in ROM and muscle paralysis causes disuse of the arm resulting in a static shoulder (Sheehan et al., 2014). Recent evidence from rodent models of BPBI suggests that contracture may be a result of restricted longitudinal growth of affected muscles in addition to the paralysis (Nikolaou et al., 2015), with marked morphological changes to the glenoid in affected animals (Crouch et al., 2015). However, the severity of the contracture varies depending the location of injury: nerve root avulsion or nerve rupture (Al-Qattan, 2003), After a nerve root avulsion (proximal to the nerve root ganglion), patients typically have muscle weakness or paralysis without severe contractures. However, nerve root rupture (distal to the ganglion) results in more severe contractures at the shoulder and elbow (Al-Qattan, 2003). Thus, understanding not only the causes of osseous deformity but the differential mechanical environment in different types of BPBI is of clinical importance.

BPBI results in muscle paralysis due to lack of activation, reduced limb use due to restriction in ROM, and probable restricted muscle growth in affected muscles. Each of these factors following BPBI alter the mechanical environment of the developing glenohumeral joint. Currently, the direct relationship between altered joint loading at the shoulder and resulting morphology of the joint remains unclear. There is limited literature available on the effects of preganglionic injury on the musculoskeletal system. This study aims to develop a novel computational framework predicting the morphological changes due to varying loads and presents a new insight on muscle architectural and morphological changes at the shoulder joint due to the two variants of the BPBI viz. preganglionic and postganglionic injury using rat models.

Specific Aim 1: **Evaluate the effect of static and dynamic loading and paralysis with and without restriction in longitudinal muscle growth on glenohumeral joint morphology after BPBI. Hypothesis:** The static nature of joint loads post BPBI are a primary driver of abnormal osseous deformations at the glenohumeral joint and glenohumeral joint deformity with restricted muscle growth will be more severe than with unrestricted muscle growth. **Motivation:** Due to the restricted ROM under BPBI, disuse and contracture may reduce the dynamic loads on the glenohumeral joint and static forces will predominate. Static loads on a skeletal joint tend to hinder bone growth, while dynamic loads tend to promote bone growth (Giorgi et al., 2014); thus altered modality of loading may be a driver of the abnormal glenohumeral growth observed postnatally (Hogendoorn et al., 2010). Prediction of this alternate joint growth pattern is critical to understanding the root cause of abnormal osseous deformities. The interactions of muscle and bone growth during development of osseous deformity are not well understood. Prior computational simulations suggest that restricted muscle growth observed in animal models of BPBI would impart excessive passive loading on the joints, oriented in directions consistent with observed deformity. However, such studies could not directly relate this loading to changes in morphology. Further, since clinical scenarios suggest that preganglionic nerve root avulsion may not experience significant contractures, understanding the effects of muscle shortening on bone loading and growth is of clinical importance. **Method:** A finite element (FE) model including mechanical and biological contributions to bone growth will be developed in ABAQUS. The bone growth model will be validated using existing data describing glenohumeral morphology in a rat model of BPBI.

Specific Aim 2: **Examine the effect of nerve injury location (preganglionic vs postganglionic) on muscle architecture and shoulder ROM. Hypothesis:** Affected muscles with preganglionic injury will have longer optimal muscle length and higher ROM with lower contractures compared to postganglionic injuries. **Motivation:** Location of the nerve injury relative to the ganglion is known to influence the presentation of contractures (Nikolaou et al., 2015; Al-Qattan, 2003). However, the specific relationship between changes in the affected muscle and shoulder ROM is not clear. **Method:** An existing rat model of BPBI, using both postganglionic (Crouch et al., 2015) and preganglionic (Nikolaou et al., 2015) neurectomy interventions, will be used to evaluate the changes to muscle architecture and joint ROM 8 weeks post injury. Rat models have been demonstrated to have similar shoulder musculoskeletal anatomy to humans (Norlin et al., 1994).

Specific Aim 3: **Examine the effect of nerve injury location (preganglionic vs postganglionic) on bone morphology. Hypothesis:** Preganglionic injury will cause less severe osseous deformities compared to postganglionic injuries. **Motivation:** Contractures are reported to be significantly more severe following nerve ruptures (postganglionic injury) compared to nerve root avulsions (preganglionic injury) (Al-Qattan, 2003). Contractures and restriction in ROM are known to directly correlate with osseous deformities at the glenohumeral joint (Pearl and Edgerton, 1998; Kozin, 2004; Bhardwaj et al., 2013). However, the specific relationship between the injury location and changes in the bone morphology is not clear. **Method:** Rat shoulder anatomy is very similar to humans (Norlin et al., 1994). Previous studies have successfully investigated the osseous deformities at the shoulder joint in postganglionic neurectomy affected rat models (Li et al., 2008; Li et al., 2010; Crouch et al., 2015). We will use both preganglionic and postganglionic injury rat models to assess morphological features of the glenoid and the humeral head.

Specific Aim 4: **Develop a novel computational platform for modeling musculoskeletal growth and loading. Hypothesis:** Inclusion of continual soft tissue development in the computational simulation improves joint morphology prediction. **Motivation:** Musculoskeletal simulations often examine a single time point and can predict functional changes (e.g. reduced ROM, altered joint reaction forces) consistent with bony deformities observed experimentally (Crouch et al., 2014; Cheng et al., 2015). However, these approaches cannot examine the time course of changes nor examine influence of rehabilitation on progression or reversal of deformity. Multi-scale interactions between time dependent tissue changes (restricted muscle growth, muscle atrophy, changes in PCSA etc.) and joint-level function are understudied, with little attention paid to the implications of altered joint morphology over time for muscle and other soft tissue function. **Method:** A novel integrated iterative musculoskeletal modeling tool will be developed to predict the effect of changes in muscle growths over time. The muscle loads will be predicted in OpenSim and will be applied as a displacement/load boundary condition in ABAQUS.

Overall impact of the study

Following this work, the mechanics of the shoulder that contribute to shoulder deformity after BPBI will be more clearly understood. In particular, the effects of preganglionic injuries on musculoskeletal development are currently undescribed in either the clinical or basic science literature. This work can have immediate clinical impact in identifying the factors that play the most significant role in altering bone growth as potential targets of intervention. In addition, I will develop a novel computational framework to predict bone growth over time and the interactions of bone and muscle growth. This has broad implications for future investigation, including predicting the effects of interventions (e.g. timing or type of nerve, muscle, or bony surgical intervention), or predicting joint growth for other joints of the body following injury or disease.

CHAPTER 2: Computational analysis of glenohumeral joint growth and morphology following a brachial plexus birth injury *(Published in Journal of Biomechanics)*

Nikhil N. Dixit, M.Tech, Daniel C. McFarland, BS, Katherine R. Saul, PhD

Abstract

Children affected with brachial plexus birth injury (BPBI) undergo muscle paralysis. About 33% of affected children experience permanent osseous deformities of the glenohumeral joint. Recent evidence suggests that some cases experience restricted muscle longitudinal growth in addition to paralysis and reduced range of motion at the shoulder and elbow. It is unknown whether altered loading due to paralysis, muscle growth restriction and contracture, or static loading due to disuse is the primary driver of joint deformity after BPBI. This study uses a computational framework integrating finite element analysis and musculoskeletal modeling to examine the mechanical factors contributing to changes in bone growth and morphometry following BPBI. Simulations of 8 weeks of glenohumeral growth in a rat model of BPBI predicted that static loading of the joint is primarily responsible for joint deformation consistent with experimental measures of bone morphology, whereas dynamic loads resulted in normal bone growth. Under dynamic loading, glenoid version angle (GVA), glenoid inclination angle (GIA), and glenoid radius of curvature (GRC) (-1.3°, 38.2°, 2.5mm respectively) were similar to baseline values (-1.8°, -38°, 2.1mm respectively). In the static case with unrestricted muscle growth, these measures increased in magnitude (5.2°, -48°, 3.5mm respectively). More severe joint deformations were observed in GIA and GRC when muscle growth was restricted (GVA: 3.6°, GIA: -55°, GRC: 4.0mm). Predicted morphology was consistent with literature reports of in vivo glenoid morphology following postganglionic BPBI. This growth model provides a framework for understanding the most influential mechanical factors driving glenohumeral deformity following BPBI.

Introduction

BPBI is the most common nerve injury among children (Foad et al., 2008), affecting 0.4 to 4 per 1000 newborns (Cheng et al., 2015). It primarily occurs due to neck hyperextension during a difficult childbirth, resulting in nerve root avulsion or rupture of the plexus; C5-C6 roots of the brachial plexus are most commonly damaged, affecting the shoulder and elbow flexion function (Al-Qattan, 2003). Abnormal growth patterns of the glenoid and humeral head are frequent sequelae of BPBI, observed in humans (Kozin, 2004; Pearl and Edgerton, 1998) as well as in a rat model of BPBI employing neurectomy (Crouch et al., 2015). Retroversion, declination, and flattening of the glenoid (Brochard et al., 2016; Hogendoorn et al., 2010) with flattened and smaller humeral heads (Sibinski et al., 2010; Kozin, 2004) are most typical.

The mechanical environment of the developing glenohumeral joint following BPBI is altered due to reduced limb use associated with muscle paralysis, restricted range of motion, and restricted muscle growth. Individuals affected with BPBI experience postural deformities, including an internally rotated and adducted arm with restricted range of motion (ROM) (Kozin, 2004). The restriction in ROM leads to disuse of the arm and a static shoulder (Sheehan et al., 2014). Recent evidence from rodent models of BPBI suggests that the severe contractures associated with BPBI may be a result of restricted longitudinal growth of affected muscles. Restricted growth of the biceps brachii was observed only after a postganglionic injury (injury distal to the dorsal root ganglion) with corresponding restriction of elbow ROM, whereas there was no observed length difference in a preganglionic injury, and no contracture reported (Nikolaou et al., 2015). Therefore, it is possible that static loading of the paralyzed shoulder and altered loading from disrupted muscle growth may play independent roles after BPBI.

It is known that bone development is greatly affected by its environment with both biological stimuli and the mechanical environment (Shefelbine and Carter, 2004) playing pivotal roles in bone morphology (Heegaard et al., 1999). For example, biological growth for long bones is proportional to the cellular activity, whereas mechanical environment modulates growth in proportion to the stress state, with static loads hindering and dynamic loads promoting bone growth (Giorgi et al., 2014). Clinical reports suggest that the mechanical loading of the shoulder in BPBI may be an important driver of bone deformity. For example, glenoid deformity is reported to be positively correlated to shoulder contracture (Al-Qattan, 2003). Further, in the case of nerve root avulsion, with injury proximal to the nerve root ganglion, patients typically have muscle weakness or paralysis without severe contractures or bone deformity (Pearl and Edgerton, 1998; Kozin, 2004; Bhardwaj et al., 2013). Prior biomechanical analyses using a computational musculoskeletal model of BPBI revealed that restriction of muscle growth would result in glenohumeral loading that is more compressive and posteriorly directed compared to an unimpaired limb, consistent with observed glenoid changes (Crouch et al., 2014; Cheng et al., 2015). However, this study did not consider the effects of static loading on the growth of bone following injury, nor did it directly link altered loading to the magnitude of morphological change.

It is unclear whether altered mechanical loading is sufficient to explain glenoid morphology after BPBI, and whether static loading or restricted muscle growth is the primary driver of changes. Therefore, our goal was to develop a computational simulation method and evaluate its feasibility to capture the effects of static loading and paralysis with and without restricted muscle growth on developing glenohumeral joint morphology after BPBI.

Methods

We used an iterative computational framework (Figure 2.1) to analyze the influence of altered loading on bone morphology following BPBI. The overall approach was to estimate joint loads under 3 clinically-relevant conditions using a musculoskeletal modeling platform (OpenSim 3.3, Stanford University) and apply these loads in a finite element analysis (Abaqus 6.13.3, Dassault Systemes) to predict bone growth in response to biological and mechanical stimuli.

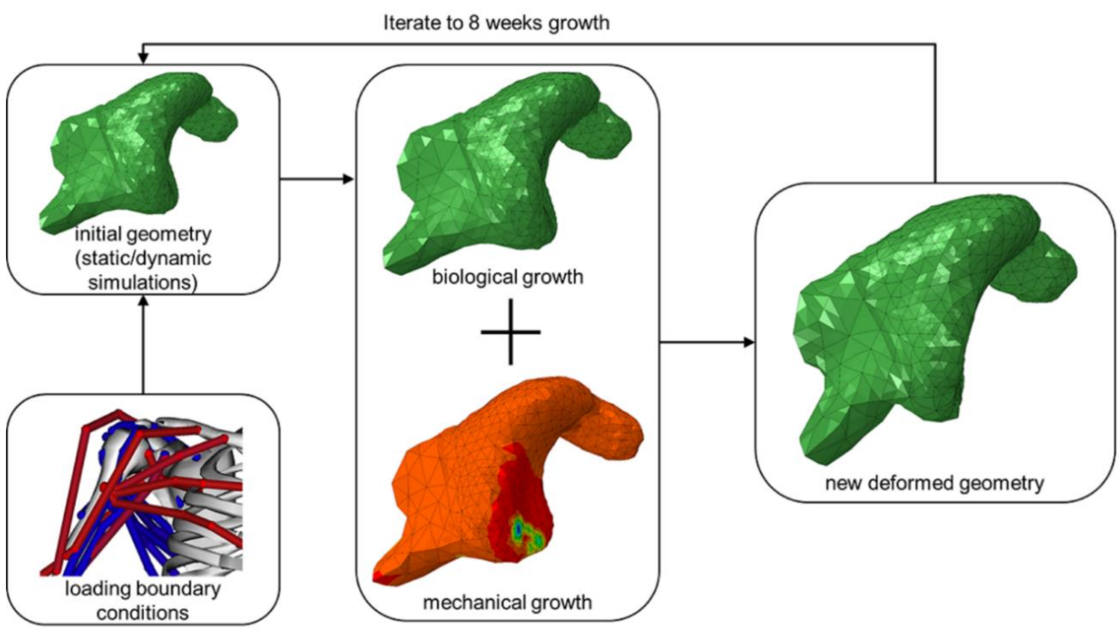

Figure 2.1: Iterative computational framework to predict glenohumeral growth. Boundary conditions derived from multibody dynamic modeling of the shoulder musculoskeletal system informed finite element models of biological and mechanical stimulus for bone growth using a thermal expansion analog.

Specifically, we simulated three conditions to examine the effects of reduced motion of the shoulder and restricted muscle growth: 1) dynamic loading (intact shoulder); 2) static with unrestricted muscle growth; 3) and static with restricted muscle growth. This analysis was applied to the geometry of a neonatal rat shoulder, derived from an existing rat model of BPBI for which quantitative descriptions of bone deformities are available (Li et al., 2010; Crouch et al., 2015); rat and human shoulder anatomy are known to be similar (Norlin et al., 1994). We simulated the growth of the humerus and scapula over 8 weeks of rat growth (equivalent to 5.8 human years) (Crouch et al., 2015) from a baseline undeformed initial condition representing 0 days postnatal.

Finite element model of bone growth

An iterative finite element (FE) modeling approach was used to predict the osseous morphological growth in response to biological and mechanical stimuli (Abaqus 6.13.3, Dassault Systemes). We extend earlier work (Giorgi et al., 2015; Giorgi et al., 2014) by incorporating realistic 3D geometries in our simulations. Initial geometry for all simulations was defined using an existing micro CT scan of an uninjured neonatal rat (Crouch et al., 2015). The geometry was reduced to include only the glenoid and humeral head regions to improve computational efficiency of the analysis (Figure 2.2). Because these neonatal bones are mostly cartilaginous in the region of interest, a cartilage material model was applied, with a Young's modulus of 1.1 MPa and Poisson's ratio of 0.45 (Giorgi et al., 2014). The geometry was meshed using quadratic tetrahedral elements (type C3D10M) with smaller elements near the articulating surface. The mesh had an element size of 0.760 mm^3 for the scapula and 0.890 mm^3 for the humerus in the volume away from articulating surface. We performed a convergence study to identify the mesh density needed to converge the peak contact stress in the artic3ulating region under a constant humeral head displacement of 0.23 mm (exceeding all displacements used in the simulations). Near the contact region, we performed simulations with mesh densities determined by increasing the seed number along the circumferential edge of each volume from 10 to 70 in increments of 10, equivalent to element sizes of 1.247 mm^3 to 0.178 mm^3 on the glenoid and 1.770 mm^3 to 0.252 mm^3 on the humeral head.

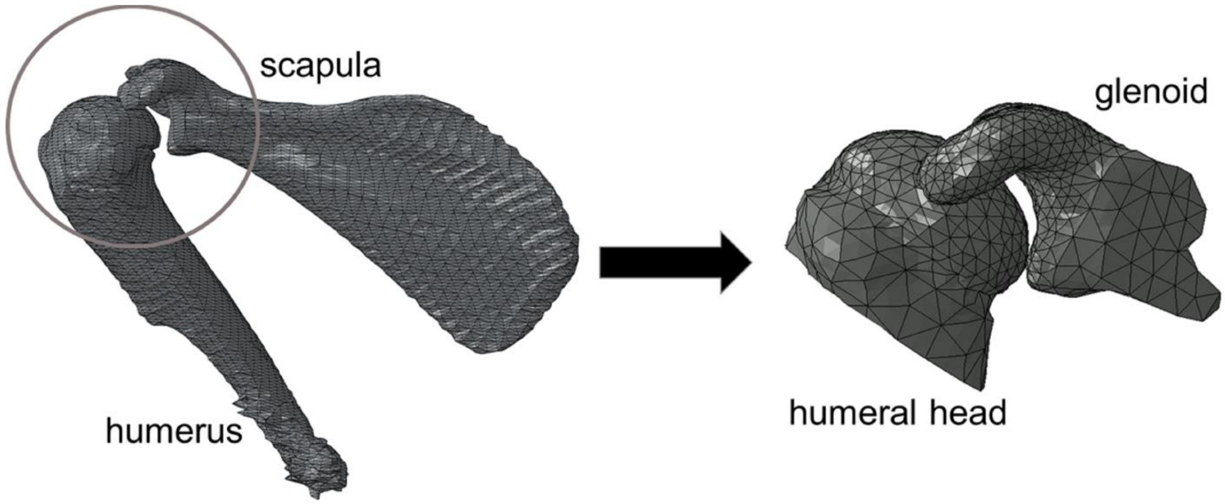

Figure 2.2: Initial glenohumeral model geometry. Meshes defining the geometry of a typically developing neonatal rat scapula and humerus derived from microCT were reduced to include only the glenoid and humeral head articulating regions to increase computational efficiency.

The location of the peak stress was monitored, and the magnitude of peak contact stress (Figure 2.3) was monitored by normalizing to contact stress predicted with a very fine mesh with a seed number of 120 (element size of 0.103 mm^3 on the glenoid and 0.147 mm^3 on the humeral head). Mesh densities over 40 seeds resulted in peak stress within 10% of the very fine mesh simulation and no change in peak stress location. Based on this result we used element size of 0.312 mm^3 on the glenoid and 0.440 mm^3 on the humeral head in the region of contact. The final glenoid mesh contained 6447 elements with 9822 nodes and the final humeral head mesh contained 4730 elements with 7109 nodes. A frictionless impenetrable surface-to-surface contact was defined between the two articulating surfaces.

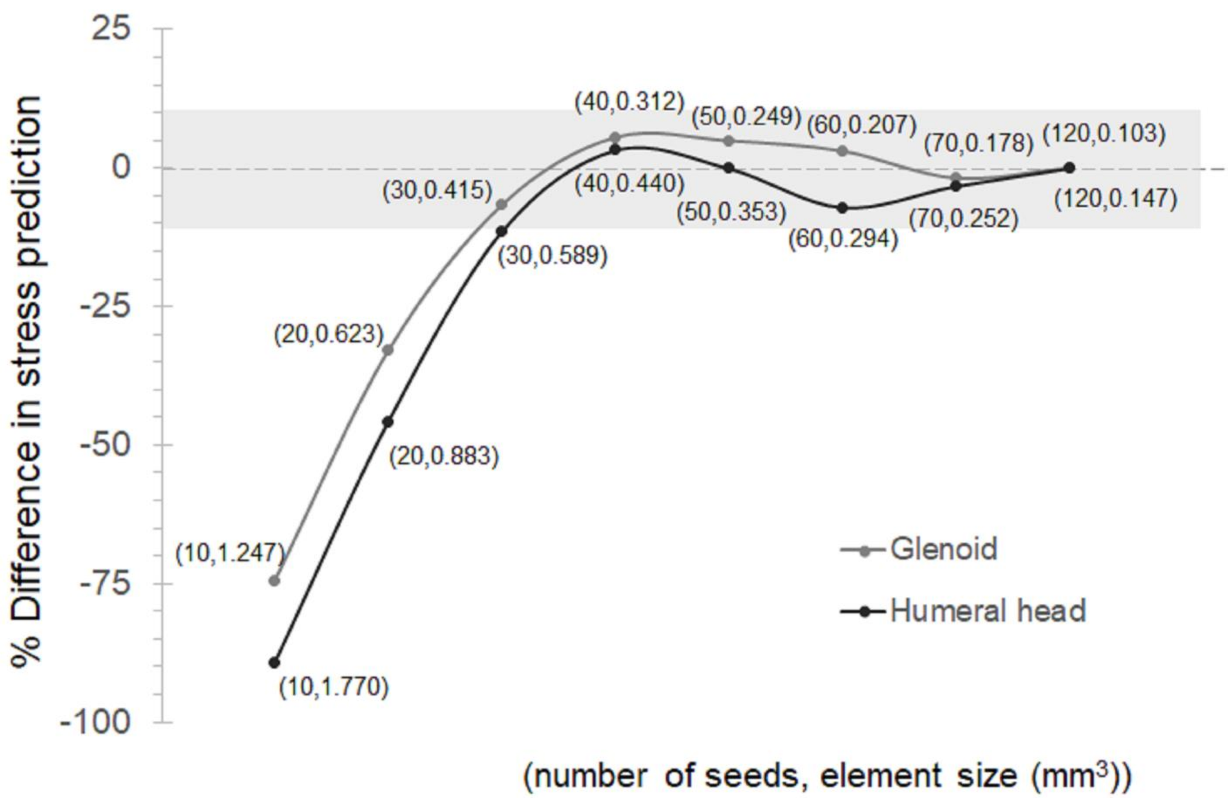

Figure 2.3: Mesh convergence for predicted peak contact stresses on the glenoid and humeral head geometries. Meshes created using greater than 40 seeds resulted in peak stress within 10% (shaded region) of peak stresses predicted using a very fine mesh created with 120 seeds for both the humeral head (black) and glenoid (grey). Each simulation point is annotated with the element size mm^3 and number of seeds

Biological growth for the humeral head was modeled using a cubic equation (Equation 1), relevant to the growth of long bones, that was a function of distance from the growth plate (Heegaard et al., 1999):

$$\dot{\varepsilon}_b = C_d = k(0.14 - 0.87\xi + 4.40\xi^2 - 2.66\,\xi^3) \tag{1}$$

where $\dot{\boldsymbol{\varepsilon}}_{\mathbf{b}}$ is the biological growth rate, proportional by constant k to the chondrocyte density $\mathbf{C_d}$; ξ is the distance of each node along the proximal-distal axis from the distal end of the rudiment. For the analysis, k was defined to be 11000; the constant determines the contribution of biological growth (Giorgi et al., 2014), and was chosen to maintain the biological growth component as 75-85% of the total growth (Germiller and Goldstein, 1997). Biological growth for the glenoid was defined to be isometric in all directions due to the multiple growth centers present within the region of interest.

Mechanical growth was defined to be proportional to the compressive hydrostatic stress at every node. Stress due to static loads restricted the growth while dynamic hydrostatic stress stimulated the growth according to the following equations:

$$\dot{\varepsilon}_m = C_d \left(\frac{\sum_{i=1}^{N} \sigma_{hi}}{N} \right), \quad \text{for static loads}$$

$$\dot{\varepsilon}_m = -C_d \left(\frac{\sum_{i=1}^{N} \sigma_{hi}}{N} \right), \quad \text{for dynamic loads} \qquad (2)$$

where $\dot{\boldsymbol{\varepsilon}}_{\mathbf{m}}$ is the mechanical growth rate, $\boldsymbol{\sigma}_{\mathbf{hi}}$ is the compressive hydrostatic stress, N is the number of iterations per time step to computational convergence, and $\mathbf{C_d}$ is the chondrocyte density as determined from the biological calculation above (Eq. 1).

Thermal expansion was used as an analog to incorporate deformations due to biological and mechanical growth. The growth rates determined from the biological and mechanical governing equations were used to define temperature gradients to identify nodal deformations, and the deformations were summed to create the resultant deformed geometry. Following each timestep, the new geometry was oriented for next cycle of simulation such that the two bodies just touched each other in their deformed state. To do so, we performed a scouting operation to identify the nearest node between the two bodies using an impact filter within an explicit dynamic analysis. The biological and mechanical computations were then repeated. In total, fifteen growth cycles were implemented to model 8 weeks of postnatal growth.

Musculoskeletal model

Boundary conditions for the loading of the finite element model were obtained from a series of musculoskeletal forward dynamic simulations. Force direction and magnitude calculated in the simulations described below were applied as displacements in the finite element growth model. The resultant joint reaction forces were converted into corresponding displacements using a joint stiffness of 0.7 N/mm (Giorgi et al., 2014).

Joint reaction forces acting at the glenoid were calculated using a musculoskeletal model of the shoulder (Saul et al., 2015) as a foundation, implemented for forward dynamic analysis in OpenSim (Delp et al., 2007) and altered to represent a neonatal rat. Twenty actuators representing muscles crossing the glenohumeral joint were included in the model. Joint forces were scaled using a force reduction factor based on a ratio of average PCSA of subscapularis and infraspinatus in neonatal rats to humans (Swan et al., 2014; Saul et al., 2015).

To capture changes in loading associated with the relative growth of the bones and muscles over time, we calculated loading in two size configurations representing the initial and ending skeletal growth after 8 weeks in a rat (Crouch et al., 2015): with the nominal model and with the skeletal geometry scaled to 1.4 times the starting geometry. To capture the influence of impaired longitudinal growth, we implemented two growth scenarios for the muscles in the model. In the first, both affected and unaffected muscles were allowed to lengthen in proportion to the skeletal growth (unrestricted muscle growth). In the second, only unaffected muscles were permitted to lengthen while affected muscles were constrained to maintain the initial optimal muscle fiber length at the initial and end skeletal sizes (restricted muscle growth). Linear interpolation was used to estimate loading at intervening timepoints.

Three loading conditions were implemented using this approach. First, we implemented a dynamic condition intended to represent an intact shoulder; thus, all muscles were considered unaffected. A cyclic motion in the scapular plane from 0° to 30° elevation was performed. The muscle activations required to perform the motion were obtained using a computed muscle control algorithm (Delp et al., 2007; Thelen and Anderson, 2006). Muscle lengths at the end timepoint were permitted to scale with skeletal size. Resulting joint reaction force was applied as a dynamic load in the finite element analysis.

Next, we simulated two static conditions representing BPBI, including muscle paralysis. We simulated denervation of muscles affected by C5-C6 level BPBI (deltoids, biceps long head, biceps short head, subscapularis, supraspinatus, infraspinatus, teres minor and teres major) (Crouch et al., 2014; Cheng et al., 2015; Hogendoorn et al., 2010; Waters et al., 2009) by restricting activation level to 0 (passive contributions only). All unaffected muscles were activating to 30% (Waters et al., 2009; Cheng et al., 2015). We simulated static loading with muscle length allowed to scale with skeletal growth, and again with affected muscles restricted to maintain the initial optimal fiber length. Joint loads were computed with the shoulder in 30° abduction and applied statically in the finite element simulation.

Analysis

Post simulation, the glenoid surfaces for each simulation were measured to assess glenoid version angle (GVA), glenoid inclination angle (GIA), and the glenoid radius of curvature (GRC). GVA is the angle complementary to the angle between the centerline of the scapula in the transverse plane and the tangent line along the cavity (Figure 2.4a). GIA is the angle complementary to the angle between the centerline of the spine of the scapula and the tangent line to the rim of the glenoid cavity (Figure 2.4b). GRC is measured by fitting a circle to the glenoid curvature (Figure 2.4c). The GVA, GIA, and the GRC predicted from the growth simulations were compared with existing geometry derived from micro CT at the same 8-week timepoint in rats. Specifically, we examined the geometry in an affected limb which had undergone postganglionic neurectomy (restriction in muscle growth) of the C5-C6 nerve roots at 5 days postnatal, and the contralateral typically developing limb (Crouch et al., 2015).

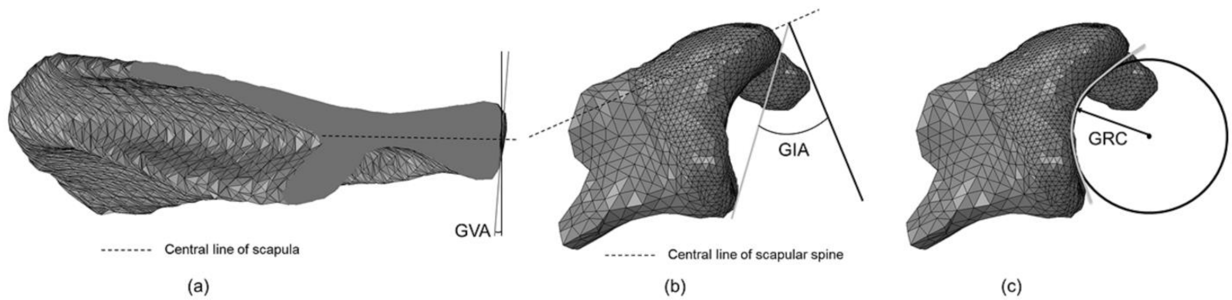

Figure 2.4: Bone deformity measures. Glenoid deformity was quantified by measuring (a) GVA, (b) GIA, and (c) GRC. GVA is the angle complementary to the angle between the centerline of the scapula in the transverse plane and the tangent line along the cavity, GIA is the angle complementary to the angle between the centerline of the spine of the scapula and the tangent line to the rim of the glenoid cavity, and GRC is measured by fitting a circle to the glenoid curvature.

Results

Deformations under the dynamic load preserved the GVA, GIA, and GRC present in the initial configuration, with no flatness on the glenoid surface (Figure 2.5). Static loading on the joint resulted in higher GVA, GIA, and GRC, with more declined glenoid fossae and prominent flatness on the glenoid. (Figure 2.5b). The change in these measures under dynamic loading to -1.2°, -38.2°, and 2.5 mm, respectively was not remarkable compared to the baseline starting point of -1.8°, -38°, and 2.1 mm, respectively. In contrast, these measures increased in magnitude to 5.2°, -48°, and 3.5 mm, respectively, in the static case with unrestricted muscle growth. More severe joint deformations were observed in GIA and GRC in the case of restricted muscle growth with GVA:3.6°, GIA:-55°, and GRC:4.0 mm in the same 8 weeks of simulated growth rat bone growth (Figure 2.5b,c).

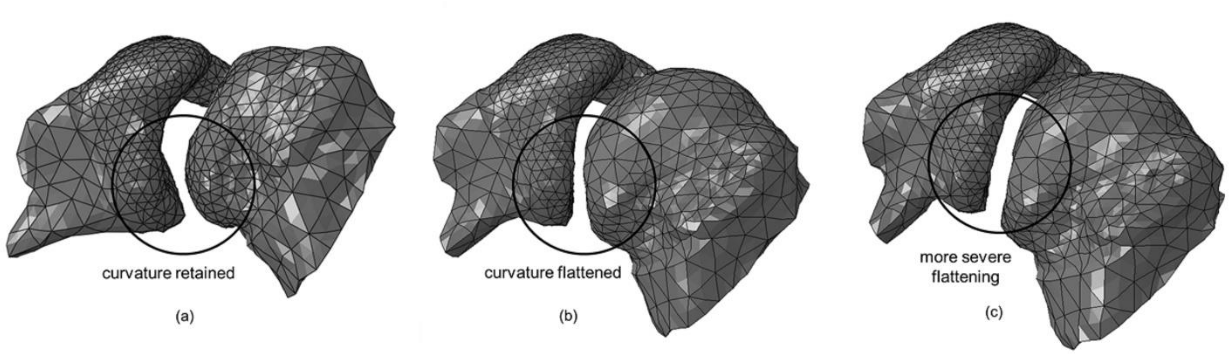

Figure 2.5: Glenoid geometry after simulated 8 weeks of growth under (a) dynamic loading and (b) static loading with unrestricted muscle growth, and (c) static loading with unrestricted muscle growth. The glenoid curvature essential for effective functioning of the joint was retained in dynamic loading, but was flattened and declined under static loading, with more severe changes when muscle growth was restricted.

The growth model predictions were consistent with the glenohumeral joint morphology previously reported for rats developing typically and after BPBI (Table 2.1). The morphology after simulated dynamic loading had similar GVA, GIA, and GRC to the typically developing rat shoulder. For example, predicted GIA (-38.2°) was similar to GIA from experimental data (-38.4° ± 3.7), as were predicted (2.5mm) and experimental (2.7mm ± 0.2) GRC. We compared the static simulation with restricted growth to measurements in rats following postganglionic neurectomy, in which restricted ROM and longitudinal muscle growth have both been reported (Crouch et al., 2015). Both computational and experimental measures displayed more declined and flattened glenoids compared to the baseline condition, with values within the experimentally measured range (GVA: simulation: 3.6°, experiment: 0.2 ± 10.1°; GIA: simulation: -55°, experiment: -56.1°± 16.6°; GRC: simulation: 4.0 mm, experiment: 5.7 mm± 3.7). No experimental data is currently available for a BPBI model in rats with reduced contracture or in the absence of restricted growth, with which the simulation results from unrestricted muscle growth could be compared.

Table 2.1: Predicted and experimentally measured glenoid morphology

	simulation				experiment[a]	
	baseline starting point	dynamic load	static: unrestricted muscle growth	static: restricted muscle growth	unaffected joint	postganglionic neurectomy (restricted muscle growth)
GVA (deg)	-1.8	-1.2	5.2	3.6	7.7 ± 4.1	0.2 ± 10.1
GIA (deg)	-38	-38.2	-48	-55	-38.4 ± 3.7	-56.1 ± 16.6
GRC (mm)	2.1	2.5	3.5	4.0	2.7 ± 0.2	5.7 ± .3.7

a: Crouch et al., 2015

Discussion

We examined whether and to what extent static joint loading and restricted muscle growth following BPBI influences glenohumeral joint morphogenesis, providing insight into the mechanical underpinnings of formation of joint deformities. The computational analysis predicted that the bone deformities were primarily related to static loads associated with paralysis and disuse. The simulations also showed notable differences in the bone morphology under restricted and unrestricted muscle growth scenarios, with glenoid deformities being more pronounced with restricted muscle growth. The overall joint shapes predicted from models in which both static loading and restricted longitudinal growth of muscle were included replicated the altered morphology of the glenoid in rats after a postganglionic neurectomy (Crouch et al., 2015), suggesting that altered mechanical loading may account for the altered glenoid morphology after BPBI.

The computational techniques employed here to model joint morphological changes with growth build upon earlier work simulating the mechanical and biological bone development over a cross section (Van Der Meulen, M C H et al., 1993), FE bone growth modeling (Heegaard et al., 1999; Giorgi et al., 2014; Giorgi et al., 2015) and stress predictions over 3D surfaces (Carriero et al., 2011; Shefelbine and Carter, 2004; Roddy et al., 2011). However, the formation of 3D osseous deformities at the shoulder, or the direct effect of muscle abnormalities on osseous deformity that we capture in the current simulations, have not been previously studied.

The simulations predicted osseous changes to the glenohumeral joint characterized by glenoid declination and increased radius of curvature, as validated against existing measurements in rats (Li et al., 2010; Crouch et al., 2015). Analogous declination deformity has been reported in human patients (Pearl and Edgerton, 1998; Sheehan et al., 2014), although retroversion is generally accepted as the most common clinical presentation (Bhardwaj et al., 2013; Eismann et al., 2016; Sibinski et al., 2010). In rat models of BPBI, retroversion is not as pronounced, and variable across animals (Li et al., 2010; Li et al., 2008), which is reflected in the relatively minor changes to GVA in the current simulations. While osseous deformity has been quantified only in postganglionic conditions in a rat model, evidence in human patients suggests that shoulder osseous and postural deformities may be more severe in postganglionic injury compared to preganglionic injury. Contractures are reported to be significantly more severe following nerve ruptures (postganglionic) compared to nerve root avulsions (preganglionic) (Al-Qattan, 2003), and glenohumeral bone deformities are known to be significantly correlated to the extent of shoulder contracture (Pearl and Edgerton, 1998; Kozin, 2004; Bhardwaj et al., 2013). In prior simulations of joint loading after

BPBI using the approach described here for determining the joint loading boundary conditions (Cheng et al., 2015), external rotation ROM was predicted to be greatly reduced when muscle growth was restricted and force components in the compressive and posterior direction, force magnitudes were higher when muscle growth was restricted. In the current study, we extend these results to demonstrate that these altered forces can be directly linked to altered growth patterns that lead to more severe morphological changes under loading conditions that restrict muscle growth. Therefore, the current simulations provide a mechanical explanation for the clinical observation that restricted ROM and contracture correlate with increased osseous deformity.

Limitations to the current study should be considered. A single plane of motion was used for simulation of dynamic conditions; three-dimensional motion would likely contribute to morphological development in other planes. Simulations with dynamic loading did not consider contributions from intermittent static loads. A typically functioning joint would experience a combination of static and dynamic forces governing the overall evolution of the joint (Giorgi et al., 2014). Joint stiffness, which was used to convert joint forces to equivalent displacements for application in the finite element models, was kept constant in all simulations and over time. It is possible that the effective stiffness may vary with condition or time. We estimated the simulated finite element model stiffness for the glenohumeral joint assemblies at the half way and end points of the simulations by measuring the peak deformation at the joint interface obtained when a constant force was applied to the humeral head. Stiffnesses at both timepoints for all static and dynamic simulations were within 10% and 15%, respectively, of the stiffness of the initial geometry. The model accounted for the effect of restriction in longitudinal growth following BPBI (Nikolaou et al., 2011), but not other potential changes to muscle properties. Muscle fibrosis (Nikolaou et al., 2014) and atrophy (Eismann et al., 2015) following BPBI have also been reported.

Future studies should explore the contributions of these changes on bone deformation for both preganglionic and postganglionic injuries in which muscle response and patient presentation may differ (Nikolaou et al., 2015). We defined material properties of the cartilage to be linear elastic, isotropic and homogeneous. Although cartilage is a biphasic material (Roddy et al., 2011), we modelled cartilage as a single phase nearly incompressible material (Poisson's ratio 0.49) (Giorgi et al., 2014). Previous studies (Carter, Dennis and Wong, 2003; Shefelbine and Carter, 2004) have shown that the fluid pressure in biphasic models and hydrostatic stress in single phase models are similar when loaded at 1 Hz, which is approximately the frequency of muscle contraction (Vaal et al., 2000). In this study, static simulations did not have applied loading at 1 Hz. However, prior work in the simulation of joint morphogenesis using a single phase model were successfully extended to predict bone development under hip dysplasia under both static and dynamic loading (Giorgi et al., 2015; Giorgi et al., 2014). Additional future work is needed to capture the effects of incorporating the single phase material in this framework when loading occurs under varied loading frequencies. The effect of synovial fluid at the joint does not play a crucial role in defining joint morphology (Giorgi et al., 2014); hence we did not model the synovial fluid interface in our simulations. We used a cartilaginous material model over the entire 8 weeks of growth simulation, however some ossification occurs during this period; future simulations could incorporate evolving material models over time to address this limitation. We modeled isotropic biological growth for the glenoid of the scapula, but there are multiple ossification centers at this stage of development (Nougarolis et al., 2017), and each may have different biological growth rates. Differing growth rates along the glenoid cavity may contribute to the differences observed between predicted and experimental GRC. We examined only changes in the glenoid. We do not report changes to the humeral head, however, the flatness at the humeral head surface due to the injury has been reported in previous studies (Sibinski et al., 2010). The biological growth was held constant across all the simulations, but the nerve injury may affect the biological growth directly. We used a linear interpolation of joint reaction forces over the iteration period, but loading may not change in a linear fashion over time. The loading pattern on the joints was obtained from a scaled musculoskeletal model using the human glenohumeral joint as a foundation; while muscle and joint geometry and osseous morphological response to BPBI is very similar in rat and human shoulders (Norlin et al., 1994), they are not identical. Development of a rat forelimb musculoskeletal model should be pursued to improve representation of muscle and joint forces in future work. The current simulations have been validated for the 8-week time point. However, additional experimental data is needed to validate and refine intermediate time points.

Our iterative model predicted that static loading of the joint is primarily responsible for joint deformation due to BPBI, whereas dynamic loading will result in typical joint growth. The model captured the effects of altered muscle forces due to paralysis and restricted longitudinal growth on the glenoid and predicted that restricted longitudinal results in more severe changes to the glenoid. The computational analyses predicted morphological changes to glenoid that were consistent with experimental data describing glenoid morphology after BPBI in a rat model. This approach offers an avenue to analyze effects of mechanical and biological growth together and in isolation and can be extended in future work to explore the effects of surgical and other interventions on osseous growth over time.

CHAPTER 3: Preganglionic and postganglionic brachial plexus birth injury effects on shoulder muscle growth

Nikhil N. Dixit, M.Tech, Carolyn M. McCormick, MS, Eric Warren, BS, Jacqueline H. Cole, PhD, Katherine R. Saul, PhD

Abstract

BPBI can differ in presentation depending on whether the nerve ruptures distal to or avulses proximal to the dorsal root ganglion, with more significant contracture and bone deformity at the shoulder typical in postganglionic injuries. However, the changes to the underlying muscle structure that drive these differences in presentation are unclear. Seventeen Sprague Dawley rat pups received preganglionic or postganglionic neurectomy on a single limb at postnatal days 3-4. Muscles crossing the shoulder were retrieved once the rats were sacrificed at eight weeks postnatally. External rotation range of motion, muscle mass, muscle sarcomere length, and calculated optimal muscle lengths were measured bilaterally. Average range of motion in the postganglionic group was 61.8% and 56.2% more restricted at 4 and 8 weeks, respectively, compared to preganglionic, but the affected muscles following preganglionic injury were altered more severely relative to the unaffected limb than following postganglionic injury. Relative to unaffected limb, optimal muscle length in the affected limb was remarkably shorter for the preganglionic group (-18.2 ± 9.2%) than the postganglionic group (-5.1 ± 6.2%). In addition, muscle mass was lower in the affected limb for the preganglionic group (-57.2 ± 24.1%) compared to the postganglionic group (-28.1 ± 17.7%). This work suggests that the presence of contracture derives not only from restricted longitudinal muscle growth, but also depends on the extent of muscle mass loss occurring simultaneously following the injury.

Introduction

Brachial plexus birth injury (BPBI) is the most common nerve injury among children (Foad et al., 2008), most frequently affecting the C5-C6 nerve roots of the brachial plexus (Lagerkvist et al., 2010). While many children spontaneously recover nerve and muscle function, 20% to 30% of the children do not have total neurological recovery (Pondaag et al., 2004). Up to 33% of children affected by BPBI sustain permanent postural and osseous deformities (Pearl, 2009), which can have lifelong consequences for upper limb function and quality of life (Bae et al., 2008). Presentation in patients, however, can vary significantly according to the location of the nerve injury relative to the nerve root ganglion. Nerve root ruptures which occur distal to the ganglion (postganglionic) frequently result in paralysis with shoulder internal rotation and elbow flexion contractures (Waters, 2005) associated with restricted passive external rotation and elbow extension. Moreover, contractures and restriction in range of motion in these patients significantly correlate with osseous deformities at the glenohumeral joint, including retroversion, declination, and flattening of the glenoid fossa (Pearl and Edgerton, 1998; Kozin, 2004; Bhardwaj et al., 2013). In contrast, avulsion injuries proximal to the ganglion (preganglionic) typically result in paralysis without contractures at the shoulder and elbow or significant osseous deformity (Al-Qattan, 2003).

The underlying mechanisms leading to these different presentations are not clear, but recent work in murine models of BPBI provide initial evidence that postganglionic contractures are associated with restricted growth of paralyzed muscles (Nikolaou et al., 2011), with 10-35% shorter optimal muscle lengths in all the shoulder internal and external rotator muscles as well as biceps short head and triceps long head muscles on the affected limb compared to the unaffected limb. Optimal muscle length is the length at which the underlying sarcomeres exhibit optimal actin-myosin overlap and produce peak force (Zajac, 1989). Postganglionic nerve injury was also associated with degenerative changes to muscle spindles and absence of important cell signals (e.g., ErbB) necessary for myogenesis and muscle regeneration (Nikolaou et al., 2015). In contrast, preganglionic injury retains ErbB signaling and spindle formation and was not associated with marked joint contracture (Nikolaou et al., 2015). A prior study of muscle architecture in a preganglionic model in mice was limited to biceps and brachialis muscles and reported that reduced muscle cross-sectional area, reduced volume, and muscle fiber atrophy were similar in magnitude to those after a postganglionic injury; the authors conclude that it is unlikely that the difference in contractures resulted from a difference in motor denervation (Nikolaou et al., 2015). However, at the shoulder, the changes to other muscles or surrounding tissues that may contribute to contracture formation are unknown; prior simulation work suggested that subscapularis, infraspinatus, latissimus dorsi, long head of biceps, anterior deltoid, pectoralis major, and long head of triceps are critical contributors to force development capacity and shoulder external rotation range of motion (Crouch et al., 2014). In addition, it is possible that the intact afferent signaling observed in the preganglionic model may not correlate with muscle architectural changes in this model throughout the limb.

We conducted a prospective study in a BPBI rat model to examine the effect of nerve injury location (preganglionic vs postganglionic) on passive external rotation range of motion (ROM) and shoulder muscle architecture, specifically optimal muscle length and muscle mass. We hypothesized that paralyzed muscles after postganglionic injury have more restriction on the ROM, muscle longitudinal growth (i.e. a shorter optimal muscle length), and lower muscle mass compared to preganglionic injuries. We also used musculoskeletal simulation to examine the direct mechanical influence of the measured architectural changes on predicted internal and external rotation ROM.

Methods

All procedures were approved by the Institutional Animal Care and Use Committee at North Carolina State University. In this study, 23 neonatal Sprague Dawley rats were divided into preganglionic and postganglionic injury groups. Preganglionic neurectomy was performed on 13 rats on the left forelimb at postnatal day 3-4, wherein the C5 and C6 nerve roots were excised proximal to the dorsal root ganglion via supraclavicular incision (Nikolaou et al., 2015). Due to the complex nature of the preganglionic surgeries, in which visual access to the nerve is limited, success of the procedure was defined as observation of the typical phenotype of an internally rotated and adducted arm during the 24 hours following surgery. On that basis, 7 out of the 13 rats were analyzed in the study. Postganglionic neurectomy was performed on the right forelimb of 10 rat pups at postnatal day 4, in which C5 and C6 nerve roots underwent transverse infraclavicular incision distal to the dorsal root ganglion through the pectoralis major muscle (Crouch et al., 2015); the nerve roots are able to be clearly visualized in this procedure. The nerve surgeries were performed on pups under isoflurane anesthesia. All pups were kept with their respective dams for the first three weeks and then weaned at postnatal day 21. Thereafter they were housed 1-3 rats per cage with sex-matched littermates. The animals were kept at room temperature (70°F) and were provided standard rat chow (Purina, Woodstock, Ontario, Canada) and water ad libitum. Body mass was measured every other day for 2 weeks following neurectomy and biweekly thereafter.

To evaluate internal rotation contracture, maximum passive external shoulder rotation was measured bilaterally at 4 weeks and 8 weeks of age using a previously developed custom device (Crouch et al., 2015) with the animals under isoflurane anesthesia (0-5% inhalation). These timepoints correspond to approximately 3.1 and 5.8 years of postnatal human musculoskeletal development (Quinn, 2005), by which time the shoulder deformity is established clinically (Quinn, 2005). The forelimb was placed at 90° shoulder abduction and elbow flexion, with the limb oriented ventrally at a neutral shoulder rotation posture (Li et al., 2008). The forearm was fixed against the rotating lever on the device. A 10-g weight was hung on the wrist of the rat to apply a constant external rotational force at the shoulder. The external rotation was measured using a protractor built onto the custom device (Crouch et al., 2015).

At 8 weeks of age, animals were sacrificed via CO2 asphyxiation followed by bilateral thoracotomy. Upon sacrifice, the intact upper torso between the neck and the diaphragm was degloved and fixed in 10% neutral buffered formalin for 48 hours (Crouch et al., 2015) and then transferred to 70% ethanol, where it was positioned in a consistent limb posture (neutral adducted shoulder and elbow flexed to 90°) using a custom fixture (Crouch et al., 2015). Muscles crossing the glenohumeral joint (i.e., biceps short head, biceps long head, anterior deltoid, spinal deltoid, pectoralis major, supraspinatus, infraspinatus, teres major, and subscapularis) were dissected carefully (Crouch et al., 2015) and stored in 70% ethanol at 4°C.

For each muscle, muscle belly length was measured using a digital Vernier's caliper with resolution of 0.01 mm. Muscle mass was measured using a digital scale with 0.01-g resolution (Sartorious AG, Göttingen, Germany). Sarcomere length was measured using a standard laser diffraction technique (Lieber et al., 1984) with a 5.0-mW, 633-nm HeNe laser (Thorlabs, Newton, NJ). Optimal muscle length (Lo) was computed (Equation 1), where muscle length (Lm) is scaled by the ratio of optimal sarcomere length for rat skeletal muscle, 2.4 µm (Burkholder and Lieber, 2001), to measured sarcomere length (Ls).

$$L_o = L_m \left(\frac{2.4\ \mu m}{L_s}\right) \qquad (1)$$

Normality of the data was evaluated using the Shapiro-Wilk test. All data were normally distributed, with the exception of muscle mass for biceps long (preganglionic and postganglionic), biceps short (postganglionic), supraspinatus (preganglionic), infraspinatus (preganglionic), and teres major (preganglionic). For the normally distributed data, the following tests were used. To evaluate whether range of motion, muscle mass, optimal muscle length, and sarcomere length for each muscle differed between the affected and unaffected limb, paired t-tests were performed. All the data was normally distributed for the paired t-tests. To evaluate whether range of motion, muscle mass, and optimal muscle length for each muscle differed between the preganglionic and postganglionic injury groups, unpaired t-tests were performed, using the difference between affected and unaffected values expressed as a percentage of the unaffected value. For the non-normal data, a Mann-Whitney non-parametric test was performed to evaluate whether muscle mass differed between preganglionic and postganglionic injury groups. Significance was assessed at the $\alpha = 0.05$ level (GraphPad Software, La Jolla, CA).

We used a human musculoskeletal model of the shoulder (Crouch et al., 2014; Cheng et al., 2015) to examine the mechanical influence of the altered muscle structure on passive axial shoulder rotation ROM of the affected limb for both injury cases. To do so, we calculated the net passive shoulder joint moment generated by the muscles crossing the shoulder with the shoulder abducted and the elbow flexed to 90° similar to the rat posture on ROM apparatus. All the muscles were deactivated in the simulations to undertake only passive force contributions from the muscles. We assumed the ratio of muscle mass to optimal muscle length to be directly proportional to the changes in the muscle physiological cross-sectional area (PCSA) thus affecting the maximum isometric force development capacity of the muscle. Similarly, changes in the optimal muscle length in the musculoskeletal model were directly applied by reducing the optimal muscle length by the % restriction in the optimal muscle length of the affected muscles relative to the unimpaired side. The ROM was defined similar to a prior study from our group (Cheng et al., 2015) as the joint angle at which the muscles crossing the shoulder exerted sufficient passive force to restrict the shoulder rotation against a 3.5 kg load applied at the wrist which was not an equivalent weight for 10g used for rats.

Results

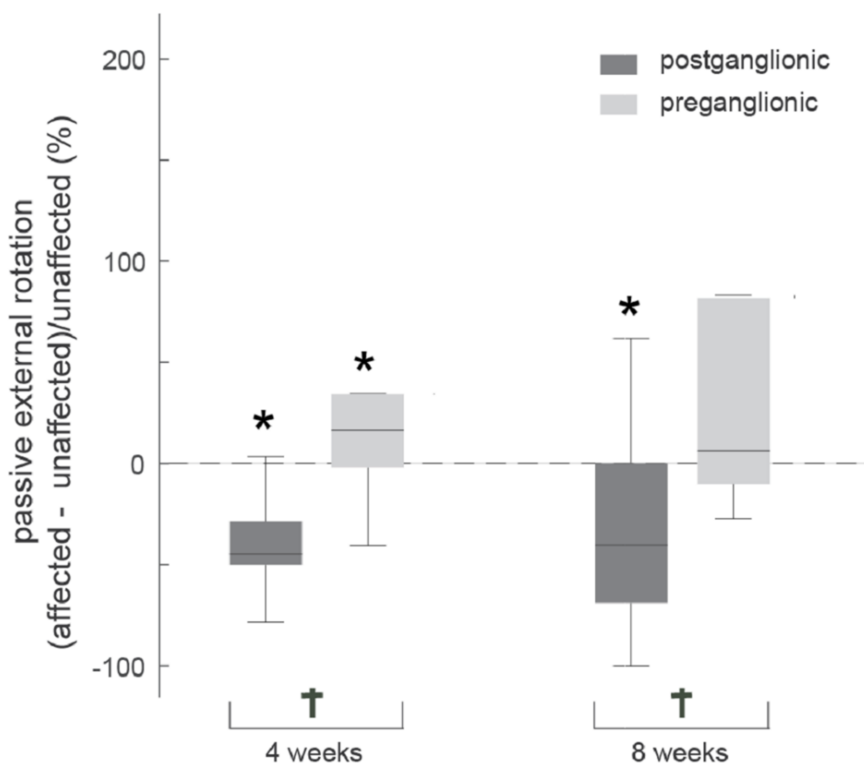

Figure 3.1. Passive external rotation for the affected limb relative to the unaffected limb at 4-week and 8-week timepoint. Postganglionic injury resulted in significant restriction of passive external rotation on the affected limb (relative to unaffected), and the restriction was significantly more pronounced (more negative) than after preganglionic. * p < 0.05 for affected vs. unaffected limb. † p < 0.05 compared to postganglionic group.

Passive external shoulder rotation ROM was significantly lower (Figure 3.1) in the affected limb than in the unaffected limb for the postganglionic group at both 4 weeks (-40.7 ± 21.8%, $p < 0.00010$) and 8 weeks (-29.6 ± 49.1%, $p = 0.0020$).

For the preganglionic group, the external rotation ROM in the affected limb was significantly greater than in the unaffected limb at 4 weeks (21.1 ± 41.0%, $p = 0.0049$), indicating less restricted movement in the affected forelimb. This comparison was not significant at 8 weeks (26.6 ± 45.8%, $p = 0.28$). Compared to the postganglionic group, the preganglionic had less restricted passive external rotation at 4 weeks ($p < 0.00010$) and 8 weeks ($p = 0.0069$).

Both postganglionic and preganglionic injury groups exhibited shorter optimal muscle lengths for several muscles in the affected compared to unaffected limb, suggesting restricted muscle growth in both cases. Compared to the unaffected limb, optimal muscle length was, on average, 5.1% shorter in the affected limb for the postganglionic group and 18.1% shorter for the preganglionic group (Figure 3.2). The optimal muscle lengths were significantly shorter in the affected limb than in the unaffected of the postganglionic group for pectoralis major (-11.1 ± 11.7%; $p = 0.018$), spinal deltoid (-14.6 ± 17.9%; $p=0.033$), and subscapularis (-9.0 ± 10.3%; $p=0.020$). The biceps long head was significantly longer in the affected limb in the postganglionic group (5.7 ± 6.2%; $p=0.017$). For the preganglionic group, optimal muscle lengths were significantly shorter in the affected limb compared to the unaffected limb for spinal deltoid (-22.4 ± 19.9%; $p=0.037$), biceps long head (-23.0 ± 13.9%; $p=0.0058$), biceps short head (-19.8 ± 16.2%; $p=0.025$), supraspinatus (-32.4 ± 9.8%; $p= 0.00050$), and teres major (-15.0 ± 18.6%; $p=0.012$). Compared to the postganglionic group, the preganglionic group had significantly shorter optimal muscle lengths for biceps long head (-28.7% vs. postganglionic; $p<0.00010$), biceps short head (-15.9% vs. postganglionic; $p=0.043$), and supraspinatus (-29.0% vs. postganglionic; $p<0.00010$).

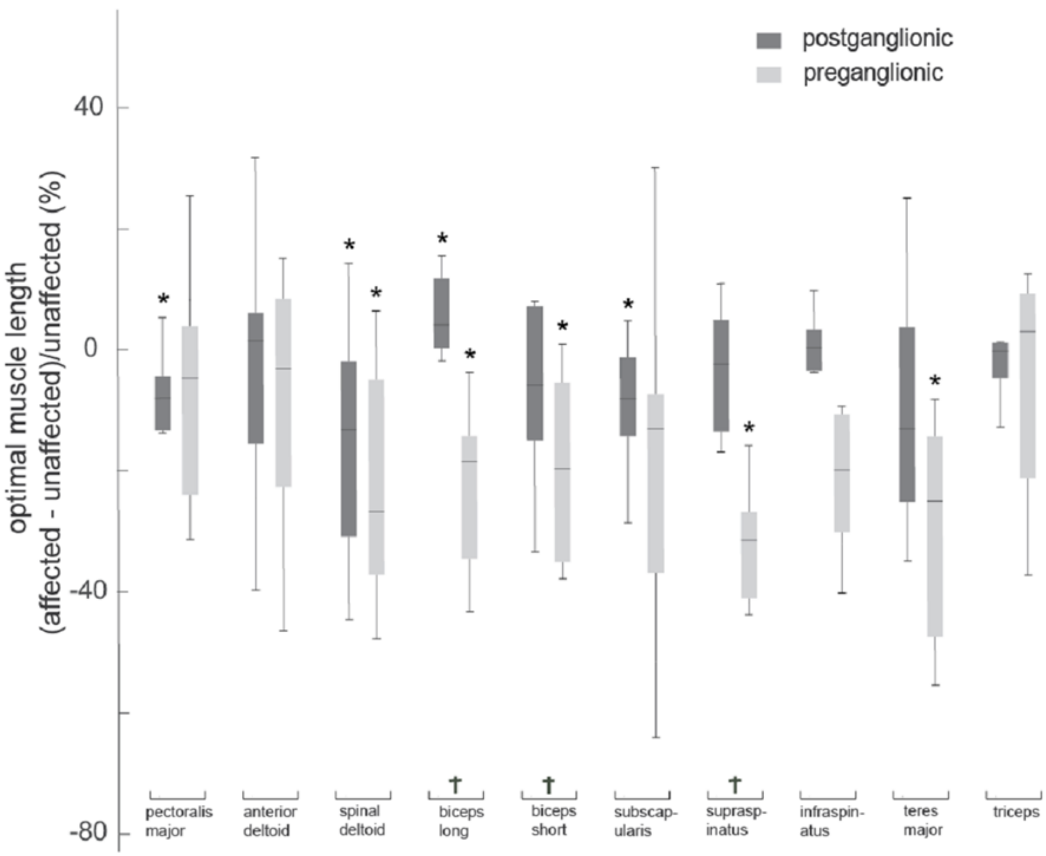

Figure 3.2. Optimal muscle length for muscles of the affected limb relative to the unaffected limb. Postganglionic injury resulted in significant restriction of muscle length in the affected limb (relative to unaffected), and the restriction was significantly more pronounced (more negative) in more muscles after preganglionic injury. * p < 0.05 for affected vs. unaffected limb. † p < 0.05 compared to postganglionic group.

All affected muscles with significantly shorter optimal muscle lengths on the affected side in both postganglionic and preganglionic injury groups exhibited longer or the same mean sarcomere lengths in the affected muscles than the same muscles on the unaffected limb (Table 3.1). Sarcomere lengths on the affected muscles were significantly longer in teres major (p=0.0046) in the postganglionic group; and pectoralis major (p=0.030), biceps long (p=0.0067), biceps short (p=0.0011), and teres major (p=0.00030) in the preganglionic group.

Table 3.1. Muscle sarcomere length, optimal muscle length, and muscle mass data for the affected and the unaffected limb in postganglionic, preganglionic, and preganglionic with phenotype injury groups. Mean ± standard deviation. # muscles with significantly longer sarcomeres in affected vs. unaffected limbs.

	postganglionic					
	affected			unaffected		
	sarcomere length (μm)	optimal length (mm)	mass (g)	sarcomere length (μm)	optimal length (mm)	mass (g)
pectoralis major	2.4 ± 0.2	18.0 ± 2.2	0.22 ± 0.09	2.3 ± 0.2	20.4 ± 2.0	0.30 ± 0.10
anterior deltoid	2.4 ± 0.2	10.7 ± 1.79	0.08 ± 0.05	2.5 ± 0.2	11.0 ± 1.4	0.13 ± 0.04
spinal deltoid	2.0 ± 0.1	16.5 ± 3.4	0.07 ± 0.05	2.0 ± 0.1	19.2 ± 1.3	0.12 ± 0.04
biceps long	2.2 ±0.2	16.6 ± 2.3	0.04 ± 0.04	2.3 ± 0.2	15.7 ± 1.6	0.09 ± 0.02
biceps short	2.2 ± 0.1	17.5 ± 2.8	0.02 ± 0.01	2.1 ± 0.2	18.1 ± 1.6	0.02 ± 0.01
subscapularis	2.2 ± 0.2	17.8 ± 1.9	0.19 ± 0.07	2.1 ± 0.1	19.3 ± 1.3	0.25 ± 0.05
supraspinatus	2.4 ± 0.2	20.6 ± 2.1	0.18 ± 0.06	2.4 ± 0.1	21.4 ± 2.2	0.2 ± 0.05
infraspinatus	2.3 ± 0.2	21.4 ± 1.65	0.19 ± 0.07	2.3 ± 0.1	21.7 ± 1.6	0.22 ± 0.05
teres major #	2.0 ± 0.1	21.0 ± 0.10	0.13 ± 0.09	1.8 ± 0.1	23.9 ± 2.5	0.21 ± 0.07
triceps long	2.0 ± 0.2	23.5 ± 0.7	0.40 ± 0.30	2.0 ± 0.2	23.9 ± 2.4	0.65 ± 0.11

	preganglionic					
	affected			unaffected		
	sarcomere length (μm)	optimal length (mm)	mass (g)	sarcomere length (μm)	optimal length (mm)	mass (g)
pectoralis major #	2.4 ± 0.4	20.6 ± 4.2	0.25 ± 0.11	2.1 ± 0.2	2.1 ± 0.17	0.3 ± 0.10
anterior deltoid	2.6 ± 0.2	10.8 ± 2.8	0.03 ± 0.01	2.4 ± 0.1	2.4 ± 0.1	0.12 ± 0.04
spinal deltoid	2.2 ± 0.2	16.1 ± 4.0	0.04 ± 0.05	2.0 ± 0.1	2.0 ± 0.1	0.13 ± 0.02
biceps long #	2.6 ± 0.2	12.5 ± 2.2	0.02 ± 0.01	2.3 ± 0.1	2.3 ± 0.1	0.1 ± 0.02
biceps short #	2.4 ± 0.2	15.7 ± 2.9	0.01 ± 0.01	2.0 ± 0.1	2.02 ± 0.1	0.03 ± 0.02
subscapularis	2.4 ± 0.6	15.5 ± 5.2	0.09 ± 0.07	2.3 ± 0.2	2.3 ± 0.2	0.29 ± 0.07
supraspinatus	2.6 ± 0.2	14.2 ± 2.8	0.04 ± 0.02	2.5 ± 0.1	2.5 ± 0.1	0.23 ± 0.08
infraspinatus	2.4 ± 0.2	17.0 ± 1.5	0.04 ± 0.02	2.4 ± 0.2	2.4 ± 0.2	0.19 ± 0.05
teres major #	2.1 ± 0.1	16.5 ± 3.9	0.09 ± 0.08	1.8 ± 0.1	1.8 ± 0.1	0.18 ± 0.08
triceps long	2.1 ± 0.3	23.6 ± 5.2	0.59 ± 0.20	2.1 ± 0.1	2.1 ± 0.1	0.76 ± 0.15

When compared to the unaffected limb, muscle mass was, on average, -28% lower in the affected limb for the postganglionic group and -57.2% lower for the preganglionic subgroup (Figure 3.3). In the postganglionic group, muscle mass was significantly lower in the affected than unaffected limb for pectoralis major (-25.4 ± 12.0%; p<0.00010), anterior deltoid (-38.0 ± 36.6%; p=0.0055), spinal deltoid (-41.5 ± 36.7%; p=0.0084), subscapularis (-24.0 ± 24.4%; p=0.013), teres major (-36.9 ± 39.9%; p = 0.023), and triceps (-42.1 ± 39.0%; p=0.0038).

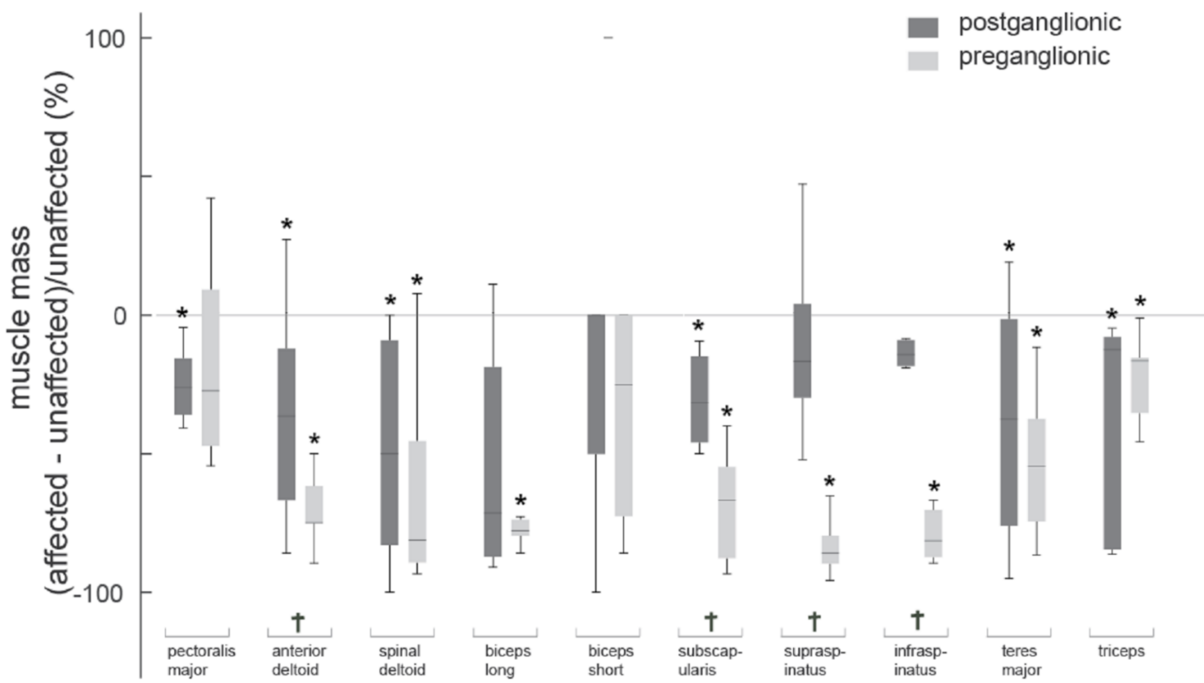

Figure 3.3. Muscle mass for muscles of the affected limb relative to the unaffected limb. Postganglionic injury resulted in significantly lower muscle mass in the affected limb (relative to unaffected), and lower mass was significantly more pronounced (more negative) in more muscles after preganglionic injury. * p < 0.05 for affected vs. unaffected limb. † p < 0.05 compared to postganglionic group.

In the preganglionic group, muscle mass was significantly lower in the affected limb compared to the unaffected limb for anterior deltoid (-70.6 ± 12.8%; p=0.0020), spinal deltoid (-64.7 ± 37.2%; p=0.0062), biceps long head (-75.6 ± 8.00%; p<0.00010), subscapularis (-69.9 ± 20.07%; p=0.00030), supraspinatus (-83.6 ± 9.8%, p=0.00050), infraspinatus (-78.8 ± 9.11%; p<0.00010), teres major (-54.5 ± 25.9%; p=0.0042), and triceps (-23.5 ± 15.5%; p=0.0046). Compared to the postganglionic group, the preganglionic exhibited significantly lower muscle mass for anterior deltoid (-32.6% vs. postganglionic; p=0.045), subscapularis (-45.9% vs. postganglionic; p=0.00090), supraspinatus (-72.2% vs. postganglionic; p<0.00010), and infraspinatus (-68.5% vs. postganglionic; p<0.00010).

In simulation, the changes in muscle structure in the preganglionic injury case were associated with an unrestricted axial rotation ROM, from -90° (full external rotation) to 120° (full internal rotation). In contrast, simulations of the postganglionic case predicted restricted axial rotation ROM in this case, limited from -90° to 114° amounting for about 3% restriction in ROM.

Discussion

This work demonstrates that both postganglionic and preganglionic injury lead to significant changes in both muscle mass and optimal muscle length in a rat model of BPBI. For both structural metrics, changes in the affected limb relative to the unaffected limb were more severe for the preganglionic injury both in terms of change magnitudes and the number of muscles affected. Despite the more pronounced changes to muscle structure in the preganglionic group, the range of motion for shoulder external rotation was significantly more restricted after the postganglionic neurectomy, consistent with clinical presentation (Al-Qattan, 2003). Alterations in muscle mass and length are consistent with prior work where data are available for comparison (Nikolaou et al., 2015; Crouch et al., 2015). For example, the changes in muscle mass and optimal muscle length for the postganglionic group are consistent with prior reports in the same rat model (Crouch et al., 2015), with pectoralis major, subscapularis, and teres major exhibiting shorter optimal muscle length; pectoralis major, spinal deltoid, and subscapularis exhibiting reduced mass; and biceps long head experiencing isolated increase in optimal muscle length, all relative to the unaffected side. Moreover, the current study provides new information about marked loss of muscle mass and reduced optimal muscle lengths after preganglionic injury in many of the muscles crossing the glenohumeral joint. Prior work in mice that report volumetric changes for only biceps and brachialis describe marked biceps atrophy in both preganglionic and postganglionic animals, but no significant difference between groups (Nikolaou et al., 2015). While many of the muscles crossing the shoulder in the current study were more severely affected for the preganglionic group, muscle atrophy was observed in the biceps long head for both groups, and for this muscle, the groups were not significantly different from one another, consistent with the prior report. It should be noted that there was marked variability across animals in biceps muscle mass in both the prior and current studies, in part because the mass of the biceps short head is extremely small on the affected side and in some cases at the resolution of the scale. Small changes in low muscle mass lead to a large percentage change relative to the unaffected limb.

Earlier studies comparing the effects of postganglionic injury in the biceps and subscapularis of mice following neurectomy reported longer sarcomeres in the impaired limb compared to control, suggesting that restricted longitudinal growth after postganglionic injury could be responsible for more severe elbow contractures (Nikolaou et al., 2011; Nikolaou et al., 2015). Passive force in a muscle can be expected to increase if a muscle with lower optimal muscle length is overstretched (Zajac, 1989), potentially leading to contracture. However, in the current study, we also found that the muscle longitudinal growth was severely restricted - as indicated by shorter optimal muscle length - in preganglionic injury, and to a greater extent overall than in postganglionic injury. Thus, shorter optimal muscle length cannot be solely responsible for governing the severity of contracture. A potential explanation for this observation is that muscle mass was also remarkably lower in the preganglionic group. The force developed by a muscle-tendon unit depends on both its optimal muscle length and cross-sectional area; cross-sectional area is directly proportional to peak force production while optimal muscle length is related to the excursion potential of muscle (Zajac, 1989). A muscle with a short optimal muscle length would produce active force over a limited range of lengths and could produce high passive forces after only limited stretch. The presence or absence of contractures that depend on the passive muscle force restricting the joint rotation would therefore be dependent on contributions from both shorter optimal muscle length (encouraging high passive forces) and loss of muscle mass (limiting force). Our simulations suggest that in the case of severe loss of muscle mass, the passive force-generating capacity of the muscles is substantially reduced, even in the presence of muscle fibers with shorter optimal muscle lengths and thus contracture is not predicted. Both muscle atrophy and shorter optimal muscle lengths are present, explaining the lack of contracture in this group. Therefore, the balance between these two factors is essential for defining the mechanical underpinnings of contracture development due to muscle architectural changes.

The study discusses factors contributing to contracture formation arising from architectural changes of the denervated muscles following BPBI which could lead to morphological changes at the glenohumeral joint. The observed changes to muscle mass and optimal muscle length inform our understanding of clinically-reported contracture and may extend to insight for the development of osseous deformities. Clinically, severity of contracture is correlated with glenoid deformity10, and evidence from computational modeling suggests that restricted muscle growth can lead to posteriorly directed glenoid forces consistent with development of retroversion, as well as restricted external rotation range of motion (Crouch et al., 2015; Cheng et al., 2015). However, in the computational study, the concurrent effects of atrophy were not considered. Other groups have linked mechanical loading to the morphogenesis of other joints such as the hip (Giorgi et al., 2015). This study provides critical new muscle architecture information that can inform future mechanical analyses of the role of each parameter in the loading of the shoulder, and the growth of the underlying bone which can be useful for improving computational models to predict morphological changes at the glenohumeral joint interface.

While the mechanisms driving the changes to muscle following BPBI remain unclear and are outside the scope of the current study, some considerations deserve future attention. The surgical approaches for the two neurectomy procedures differ. For example, the postganglionic neurectomy requires an infraclavicular approach and division of the pectoralis major to expose the brachial plexus, while the preganglionic procedure requires a supraclavicular approach. In prior work, we demonstrated that division of the pectoralis major disrupts its growth compared the unaffected limb (Crouch et al., 2015) and may explain differences in the growth of this muscle between the two groups in the current study. The preganglionic procedure also requires incision of the dura of the spinal cord, and visibility of the nerve root is more challenging. Therefore, it is possible that for some animals, a preganglionic neurectomy affects more of the surrounding nerve supply, which may explain the larger number of muscles exhibiting structural changes. Nevertheless, the presentation of reduced contracture in the preganglionic case is consistent with the clinical presentation (Al-Qattan, 2003). Although the neurectomy procedure is intended to create isolated C5-C6 disruption, the traumatic avulsion BPBI may also disrupt surrounding nerve supply and affect muscle growth. Prior work in a mouse model reported that postganglionic nerve injury was associated with degenerative changes to muscle spindles and absence of important cell signals (e.g., ErbB) necessary for myogenesis and muscle regeneration (Nikolaou et al., 2011; Nikolaou et al., 2015). In contrast, preganglionic injury retained ErbB signaling and spindle formation.

The results of the current study should be evaluated in light of the study limitations. Although nerve excision and phenotype were evaluated and confirmed immediately following the surgical procedure, spontaneous recovery of the nerve injury can occur and may contribute to the observed variability within groups (Pearl, 2009). Although the rat neuromuscular anatomy is similar to that of humans (Norlin et al., 1994), there are some differences in muscle anatomy, such as existence of the spinal deltoid in rats that is not present in humans, and glenohumeral loading may differ since rats are lifelong quadrupeds. Muscle architecture was investigated at only one timepoint in both injury groups, 8 weeks of age, at which time deformity is known to be well-established in this model (Crouch et al., 2015; Li et al., 2008; Li et al., 2010). Assessing the progression of the injury over time will be critical in future work to inform both timing of deformity initiation and optimal treatment timing.

This study provides new information regarding muscle architectural changes following preganglionic and postganglionic BPBI. While muscle mass loss and reduced optimal muscle length are present for both injury locations, the effects are significantly more severe in preganglionic BPBI. However, contractures at the shoulder are more severe in postganglionic BPBI compared to preganglionic, which is consistent with previous clinical studies in humans. The findings from this work suggest that contracture derives not only from restricted muscle growth, but also depends on the extent of muscle mass loss occurring simultaneously following the injury.

CHAPTER 4: Influence of brachial plexus birth injury location on morphology of the glenohumeral joint

Nikhil N. Dixit, M.Tech, Carolyn M. McCormick, MS, Jacqueline H. Cole, PhD, Katherine R. Saul, PhD

Abstract

Patient presentation following brachial plexus birth injury (BPBI) is influenced by nerve injury location, either ruptures distal to or avulses proximal to the dorsal root ganglion, with more significant contracture and bone deformity at the shoulder in postganglionic injuries. While the extent of bone deformity after postganglionic injury is well characterized, the nature of glenohumeral deformation following preganglionic BPBI are unclear. Twenty Sprague Dawley rat pups received preganglionic or postganglionic neurectomy on a single limb at postnatal days 3-4. Glenohumeral joints on the affected and unaffected side were analyzed through microCT scan following sacrifice at eight weeks postnatal. After postganglionic injury, the glenoid was significantly more declined on the affected limb compared to the unaffected limb ($-17.7 \pm 16.9°$, $p = 0.021$). Compared to the preganglionic group, the affected shoulder in the postganglionic group exhibited significantly higher declination ($p = 0.0010$) and higher glenoid radius of curvature ($p = 0.0070$). After preganglionic injury, humeral head radius of curvature, thickness, and width were significantly smaller on the affected side compared to the unaffected side (curvature: -0.5 ± 0.5 mm, $p = 0.0085$; thickness: -0.5 ± 0.5 mm, $p = 0.011$; width: -0.4 ± 0.4 mm, $p = 0.0076$) and tended to be linearly interlinked. The findings show severe glenoid deformities following a postganglionic BPBI unlike that in the preganglionic group. However, the preganglionic BPBI indicates a probable alteration in the biological growth of the bones following a nerve injury with this group showcasing smaller glenoid and humeral heads.

Introduction

Brachial plexus birth injury (BPBI) is the most common nerve injury among children (Foad et al., 2008), most frequently affecting the C5-C6 nerve roots of the brachial plexus (Lagerkvist et al., 2010). While many children spontaneously recover nerve and muscle function, 20% to 30% of the children do not have total neurological recovery (Pondaag et al., 2004). Up to 33% of children affected with BPBI sustain permanent postural and osseous deformities (Pearl, 2009), which can have lifelong consequences for upper limb function and quality of life (Bae et al., 2008). The osseous changes include abnormal glenoid development, such as higher retroversion and declination (Eismann et al., 2016; van Gelein Vitringa, Valerie M et al., 2013; Bhardwaj et al., 2013; Eismann et al., 2015), as well as increased humeral head flattening at the joint interface (Sibinski et al., 2010). The orientation and positioning of the glenohumeral joint is also affected, with posterior humeral head subluxation (Eismann et al., 2015; Eismann et al., 2016; Bhardwaj et al., 2013; van Gelein Vitringa, Valerie M et al., 2013). These severe osseous deformities due to BPBI affect function of the glenohumeral joint, making it unstable (Di Mascio et al., 2011; Kambhampati et al., 2006).

BPBI presentation in patients can vary significantly according to the location of the nerve injury relative to the nerve root ganglion. Nerve root ruptures which occur distal to the ganglion (postganglionic) frequently result in shoulder internal rotation and elbow flexion contractures (Waters, 2005). In contrast, avulsion injuries proximal to the ganglion (preganglionic) typically result in paralysis without contractures at the shoulder and elbow (Al-Qattan, 2003). The extent of bone deformity can be an indication used to inform surgical treatment planning (Eismann et al., 2016; Di Mascio et al., 2011), but the relationship between location of nerve injury and bone deformity is not well described. However, clinical reports suggest that postural contractures significantly correlate with osseous deformities at the glenohumeral joint (Al-Qattan, 2003), implying that the extent of deformity may be related to nerve injury location. Variability in patient condition - including the extent of nerve root involvement, completeness of nerve transection and recovery, and treatment history - can confound the ability to tease out direct relationships between nerve injury location and bone changes from the clinical record. (Pearl and Edgerton, 1998; Bhardwaj et al., 2013).

Rat models of BPBI offer an opportunity to examine the effect of nerve injury location on bone growth in a controlled manner. Prior work in postganglionic rat models of BPBI exhibit shoulder anatomy (Norlin et al., 1994) and bone deformities similar to that of human patients, including increased glenoid retroversion and declination (Li et al., 2008; Li et al., 2010; Crouch et al., 2015) along with smaller humeral head size. However, bone changes in preganglionic BPBI have not been previously reported, nor the relationship between humeral and glenoid changes. We conducted a study in a BPBI rat model with the goal to characterize the isolated effect of C5-C6 nerve injury location (preganglionic and postganglionic) on glenoid (version, inclination, radius of curvature) and humeral (head size, radius of curvature) morphology. Our hypothesis was that the preganglionic injures would have less severe deformities compared to the postganglionic injuries, and that glenoid and humeral head changes would be correlated with one another.

Methods

All procedures were approved by the Institutional Animal Care and Use Committee (IACUC). We examined 20 neonatal Sprague-Dawley rats in two injury groups: preganglionic and postganglionic. Neurectomies were performed on pups 3 days postnatal under isoflurane anesthesia. Preganglionic neurectomy was performed on the left limb of 12 rats, in which C5 and C6 nerve roots were excised proximal to the dorsal root ganglion via supraclavicular incision

(Nikolaou et al., 2015). Postganglionic neurectomy was performed on the right limb of 8 rats, in which C5 and C6 nerve roots underwent transverse infraclavicular incision distal to the dorsal root ganglion through pectoralis major muscle (Crouch et al., 2015). The rats were closely monitored and butorphanol tartrate was administered for pain relief if needed (Crouch et al., 2015). All pups were kept with their respective dams for three weeks, weaned at postnatal day 21, and subsequently housed in shared cages with no more than 3 pups per cage. The animals were maintained with food and water ad libitum. Body mass was measured every other day for 2 weeks following neurectomy, and biweekly thereafter. At 8 weeks postnatal, animals were sacrificed via CO2 asphyxiation followed by bilateral thoracotomy. Upon sacrifice, the upper limbs were fixed in 10% neutral buffered formalin (NBF).

Following sacrifice, micro-CT scans (36µm voxel size, SCANCO µCT 80) were used to assess glenohumeral morphology. The scapula and humerus were segmented in Mimics (Materialise, Leuven, Belgium) using a custom image threshold. Anatomical measurements were made from the reconstructed three-dimensional volumetric images, including glenoid version angle (GVA), glenoid inclination angle (GIA), glenoid radius of curvature (GRC), humeral radius of curvature (HRC), humeral head thickness, and humeral head width. GVA is defined in the transverse plane of the scapula and humeral head (Figure 4.1a), as the complementary angle between the centerline of the scapula in the transverse plane and the tangent to the glenoid rim. GIA is the complementary angle between the scapular spine and the tangent to the glenoid rim (Figure 4.1b). The glenoid radius of curvature (GRC) (Figure 4.1c) and the humeral head radius of curvature (HRC) (Figure 4.2b) were measured by fitting a circle to the respective surfaces. The humeral head width and thickness were measured in the same plane as GIA along a line parallel to the inferior region of the growth plate (Figure 4.2a).

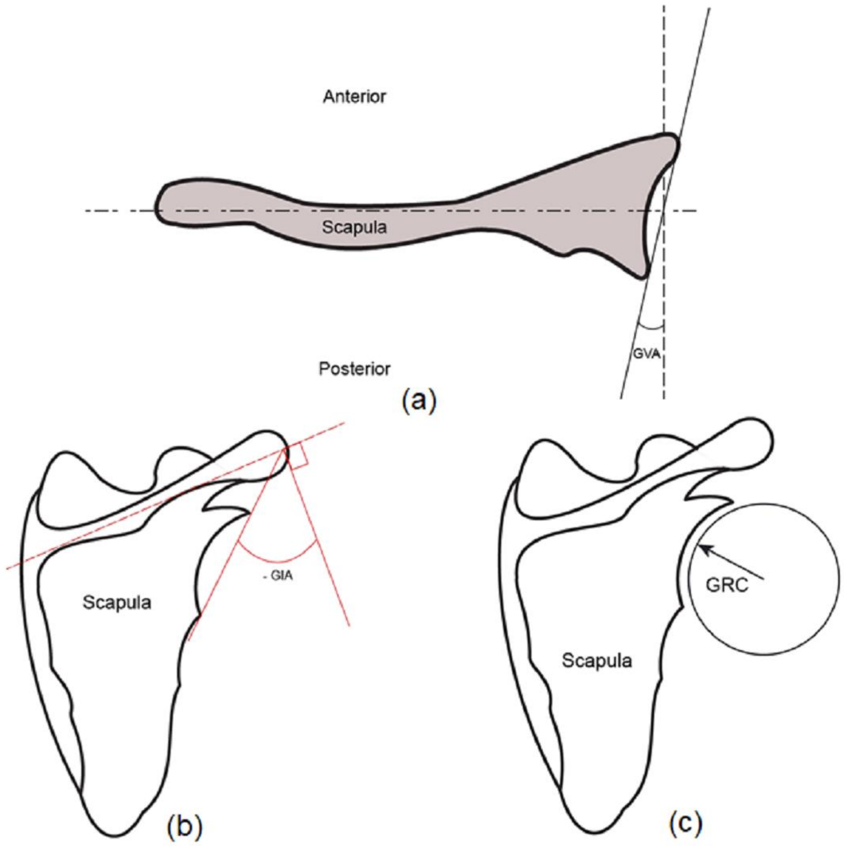

Figure 4.1: (a) GVA is defined in the transverse plane of the scapula and humeral head as shown. It is the complementary angle between the centerline of the scapula and the tangent to the rim of the glenoid. (b) GIA is the complementary angle between the scapular spine and the to the rim of the glenoid. (c) GRC is measured by fitting a circle at the glenoid surface.

The humeral head width and thickness was measured using an ellipse tool in Mimics such that the diameter D2 was drawn by connecting points 1 and 2 (Figure 4.2a) parallel to the inferior region of the growth plate and the center of this line lied on the growth plate as it curves. Point 3 was located at the articular surface of the humeral head to construct the ellipse with D1/2 representing the thickness of the humeral head (Figure 4.2a).

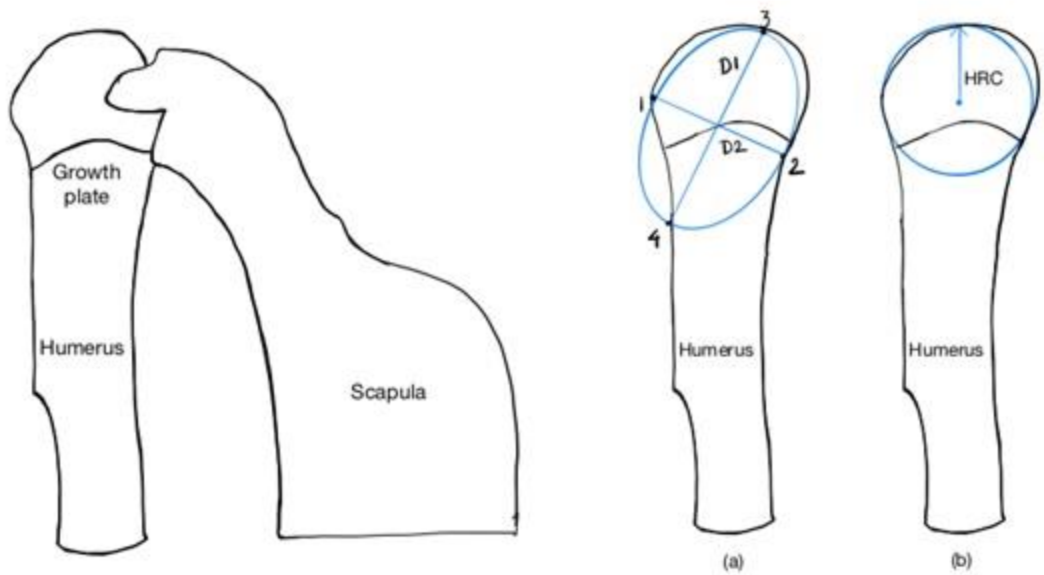

Figure 4.2: (a) The humeral head width and thickness was measured using an ellipse tool in Mimics such that the diameter D2 was drawn by connecting points 1 and 2 parallel to the inferior region of the growth plate and the center of this line lied on the growth plate as it curves. Point 3 was located at the articular surface of the humeral head to construct the ellipse with D1/2 representing the thickness of the humeral head (b) HRC is measured by fitting a circle at the humeral head surface

Normality of the data was evaluated using the Shapiro-Wilk test. To evaluate whether the anatomical measurements on the glenoid and humeral head differed in the affected and the unaffected limb, paired t tests were performed for normally distributed data and Wilcoxon signed rank test for non-normal data. To evaluate whether these measurements differed between the preganglionic and postganglionic injury groups, unpaired t tests were performed for normally distributed data and Mann-Whitney U test for non-normal data. Significance was assessed at the $\alpha < 0.05$ level. To investigate existence of any relationship between the GVA and GIA, GRC and HRC, and humeral head thickness and width in preganglionic and postganglionic injuries, multiple regression analysis was performed.

Results

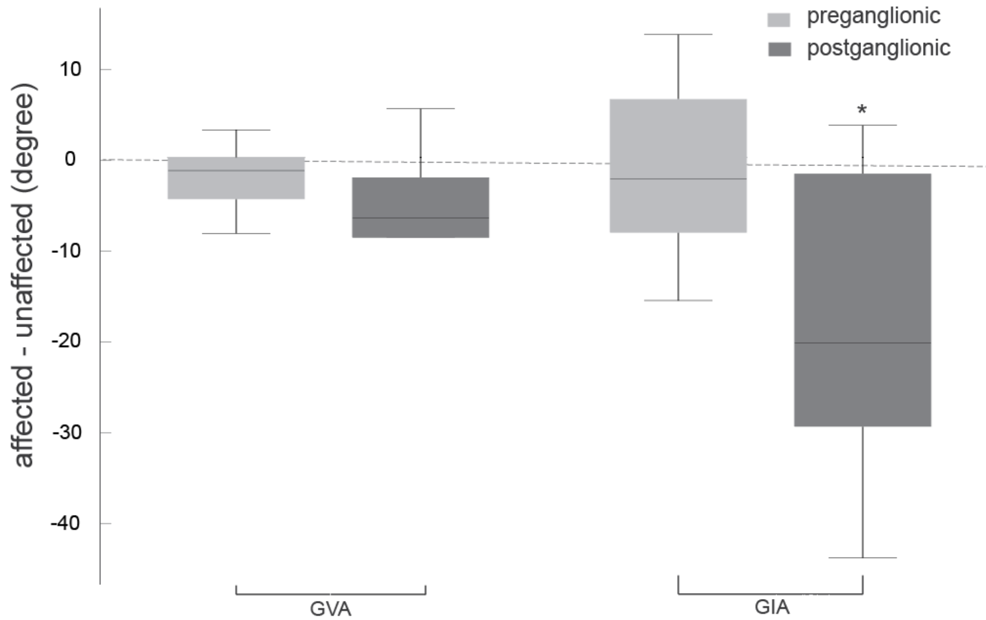

Figure 4.3: The glenoid tended to be retroverted and was significantly more declined on the affected limb compare to the unaffected limb in postganglionic injuries. The difference in GIA between the two injury groups was significant with higher declination in the postganglionic group compared to the preganglionic. * $p < 0.05$ for affected vs. unaffected limb. + $p < 0.05$ compared between the two injury groups.

All data were normally distributed except for humeral head thickness in the unaffected limb of the preganglionic group, and the difference in glenoid version between the affected and unaffected side in the postganglionic group. In the preganglionic group, the glenoid tended to be retroverted (-1.7 ± 3.4°, p = 0.11) relative to the unaffected limb, but there was no significant difference in glenoid declination (-0.9 ± 9.3°, p = 0.75) between affected and unaffected limbs (Figure 4.3). In contrast, the glenoid tended to be retroverted (-7.5 ± 11.1°, p = 0.10) and was significantly more declined (-17.7 ± 16.9°, p = 0.021) on the affected limb compared to the unaffected limb in postganglionic injuries (Figure 4.3).

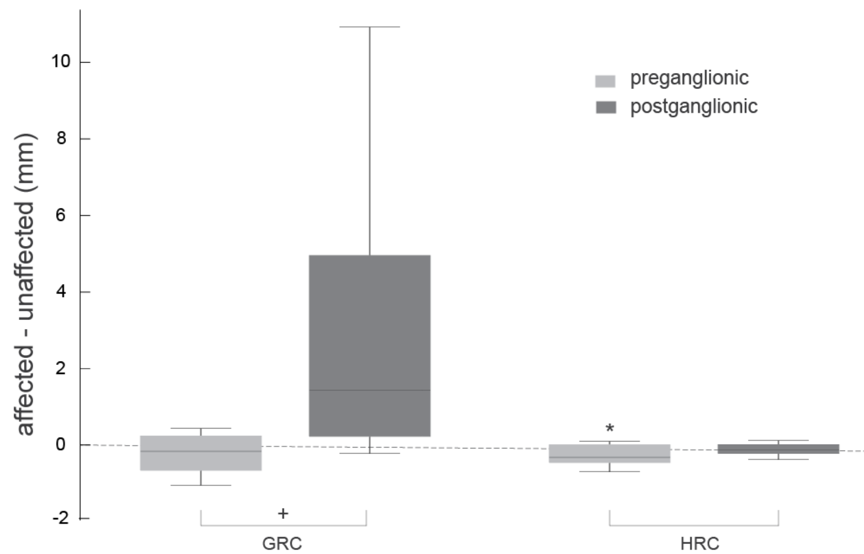

Figure 4.4: GRC tended to be smaller in both preganglionic and postganglionic injuries. The difference in GRC between the two injury groups was significant with the postganglionic injuries showing larger GRC than the preganglionic group. HRC was significantly smaller on the affected side compared to the unaffected side after preganglionic injury. * $p < 0.05$ for affected vs. unaffected limb. + $p < 0.05$ compared between the two injury groups.

GRC on the affected side of the preganglionic group tended to be smaller than the unaffected side (-0.5 ± 0.1 mm, p = 0.13) (Figure 4.4). In contrast, the GRC on the affected side of the postganglionic group tended to be substantially larger than the unaffected side (3.0 ± 3.8 mm, p = 0.060) (Figure 4.4). When comparing between injury groups, the glenoid exhibited significantly higher declination (p = 0.0010) and GRC (p = 0.0070) and a tendency to be more retroverted in the postganglionic group than preganglionic (p = 0.11).

In the preganglionic group, HRC was significantly smaller on the affected side compared to the unaffected side (-0.5 ± 0.5 mm, p = 0.0085), and the humeral head was significantly smaller in thickness (-0.5 ± 0.5 mm, p = 0.011) and width (-0.4 ± 0.4 mm, p = 0.0076) on the affected compared to unaffected limb (Figure 4.5). In contrast, there was no significant difference between the HRC in the two limbs following postganglionic injury (-0.1 ± 0.5 mm, p = 0.47) (Figure 4.4). The postganglionic group also did not exhibit significant changes in humeral head thickness (-0.3 ± 0.6 mm, p = 0.18) nor width (-0.2 ± 0.4 mm, p = 0.23) on the affected limb compared to the unaffected limb. There was no significant difference between the two injury groups in humeral head thickness (p = 0.6), width (p = 0.35) or HRC (p = 0.17) (Figure 4.5).

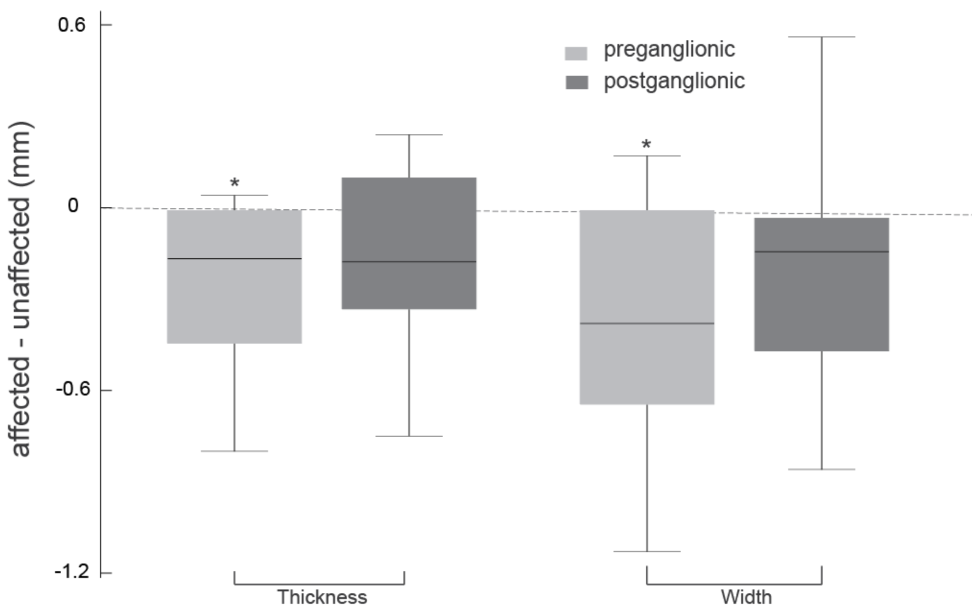

Figure 4.5: The humeral head was significantly smaller in thickness and width on the preganglionically affected limb compared to unaffected limb. The postganglionic group did not exhibit significant changes in humeral head thickness nor width on the affected limb. * p < 0.05 for affected vs. unaffected limb.

Changes in the glenoid version were excluded from the multiple regression analysis as changes were not found to be statistically significant through the t-tests. The preganglionic group through multiple regression indicated only changes in HRC and humeral head width (p = 0.019) to be correlated. All the other parameters were found to be statistically independent of others. A linear relationship between HRC and the humeral head thickness fitted the line with $R^2 = 0.84$ with equation as: humeral head width = (0.70 * HRC) – 0.039. Humeral head thickness and width in this group also tended to change together ($R^2 = 0.70$) with equation: humeral head width = (0.63 * humeral head thickness) – 0.079. In the postganglionic group, the multiple regression yielded a relationship where humeral head curvature tended to be dependent on humeral head thickness ($p = 0.13$) with a coefficient of 2.37 and GIA ($p = 0.098$) with a coefficient of 0.031. The changes in humeral head thickness tended to be correlated with humeral head width ($p = 0.15$) following an equation: humeral head width = (0.59 * humeral head thickness) - 0.004 with $R^2 = 0.77$.

Discussion

Glenohumeral morphology was markedly affected by neonatal injury to the brachial plexus, and effects were significantly influenced by the location of the injury relative to the nerve root ganglion. The morphology of the glenoid was most altered following postganglionic injury, with severe glenoid retroversion observed on the affected limb compared to the unaffected limb (difference between limbs = -7.4 ± 11.4°). Similar glenoid retroversion has been observed in humans (difference between limbs = -10.2 ± 11.57°) in prior work examining 102 children (Hogendoorn et al., 2010). We observed glenoid declination in the postganglionic group, consistent with reports of declination following BPBI in humans (Eismann et al., 2016) and rats (Li et al., 2008; Li et al., 2010; Crouch et al., 2015). We also observed higher glenoid radius of curvature on the affected side in the postganglionic injury group, indicating flatness at the glenoid surface. Since glenoid curvature is essential for maintaining shoulder stability (Flatow et al., 1991), flattening represents a meaningful alteration to the function of the joint. We found no evidence of altered glenoid version, declination, or radius of curvature relative to the unaffected limb in the preganglionic group. Taken together, these observations suggest that glenoid deformation is severe following postganglionic injury only, and prior clinical reports of glenoid deformities (Bhardwaj et al., 2013; Sibinski et al., 2010; Hogendoorn et al., 2010; Waters, Peter M. et al., 1998) are more likely to be associated with postganglionic BPBI.

The overall size of the humeral head was markedly smaller with a correspondingly smaller radius of curvature after preganglionic injury, while there were no observed significant changes to the humeral head after postganglionic injury. Prior work reports that humeral head width and thickness tended to be smaller after postganglionic injury in rats (Li et al., 2010), but no change to radius of curvature was reported. Clinical reports on injured children also describe smaller humeral head features on the affected limb which are not limited to axial plane but are a 3D phenomenon (Sheehan et al., 2014). Interestingly, the preganglionic group also exhibited smaller GRC on the affected limb, rather than the glenoid flattening and tilting common to the postganglionic group.

One possible explanation for the differences in bone development following preganglionic and postganglionic injury are differences in loading at the joint in the two presentations. Bone growth is driven by both mechanical and biological factors (Swan et al., 2014), and mechanical loading of bone tissue will be influenced by joint forces induced by muscle loading and limb use. Postganglionic injury is frequently associated with muscle contractures that cause restriction in external rotation (Bhardwaj et al., 2013; Eismann et al., 2015; Pearl and Edgerton, 1998). These observations in the clinical population are corroborated by work in rat and mouse models of BPBI, in which restricted muscle growth and increased contracture are reported (Nikolaou et al., 2015; Crouch et al., 2015). However, in both clinical and animal model reports, contractures are minimal following preganglionic injuries (Dixit, McCormick, Warren et al., 2019; Al-Qattan, 2003). In addition, computational models of BPBI suggest that reduced muscle growth would mechanically result in both restricted range of motion and altered glenoid loading consistent with the observed deformity following postganglionic neurectomy (Cheng et al., 2015). In contrast, in the absence of significant contracture following preganglionic injury, altered biological signals due to the nerve injury may dominate the changes in bone growth, leading to overall slower bone formation and smaller bones. Clinical (Sheehan et al., 2014) and animal (Li et al., 2010) studies has seen a smaller bone geometry as a result of probable altered biological growth following a nerve injury. This potential explanation is supported by the overall small humeral head and smaller GRC at the glenoid.

Surgical approaches are often used to correct shoulder dislocation and severe shoulder instability that are a consequence of glenohumeral deformity after BPBI (Kambhampati et al., 2006). Methods include anterior translation of the humeral head to prevent dislocation (Waters et al., 2009), external rotation tendon transfer (Eismann et al., 2016), or glenoid osteoplasty or osteotomy (Di Mascio et al., 2011). Surgical decisions are typically planned based on axial plane measures (Waters et al., 1998; Lippert et al., 2012), especially glenoid retroversion, which do not account for coronal or sagittal plane changes. However recent medical imaging reconstructions of glenoid geometry in BPBI patients suggest that declination of the glenoid may also be a significant alteration after nerve rupture (Crouch et al., 2015). The current study corroborates that glenoid declination is also a key marker in characterizing glenoid deformity in this animal model. However, the current work also suggests that glenohumeral deformation depends heavily on BPBI location, which may therefore provide additional foundation by which surgical decision-making can be informed.

The results of the current study should be evaluated in light of study limitations. The analysis could yield a clearer comparison between the two injuries especially with a larger sample size of the postganglionic group. Although nerve excision was evaluated and confirmed immediately following the surgical procedure, spontaneous recovery of the nerve injury can occur and may contribute to the observed variability within groups (Pearl, 2009). Even though the rat neuromuscular anatomy in the subacromial space is similar to humans (Norlin et al., 1994), the glenohumeral mechanical loading may differ between the species since rats are lifelong quadrupeds, while infants may for their first year (Crouch et al., 2015). Bone deformity was investigated at only one time point (8 weeks postnatal) for both injury groups; by this time point deformity is known to be well-established in this model (Li et al., 2008; Li et al., 2010; Crouch et al., 2015). However, progression of the injury over time will be critical to assess in future work to understand timing of deformity initiation and inform optimal treatment timing.

In summary, the study clearly identifies differences in glenohumeral osseous deformities following preganglionic and postganglionic BPBI. Glenoid deformities were more severe after postganglionic injury, with markedly altered shape profile, while preganglionic injury exhibited more pronounced humeral head alterations, with an overall smaller head.

CHAPTER 5: Integrated iterative musculoskeletal modeling analysis of influence of muscle structure on bone morphology following brachial plexus birth injury

Nikhil N. Dixit, M.Tech, Daniel C. McFarland, BS, Matthew B. Fisher, Jacqueline H. Cole, PhD, Katherine R. Saul, PhD

Abstract

Brachial plexus birth injury (BPBI) is the most common nerve injury among children. The glenohumeral joint of affected children can undergo severe osseous deformation and muscle contracture, depending on location of the injury relative to the dorsal root ganglion (preganglionic or postganglionic). We investigated whether changes to muscle mass and optimal muscle length over time following BPBI provide a mechanically-driven explanation for observed differences in postural and bone deformity between preganglionic and postganglionic BPBI. We developed a computational framework integrating musculoskeletal modeling to represent muscle changes over time and finite element modeling to simulate bone growth in response to mechanical and biological stimuli. The simulations predicted that the net glenohumeral joint loads in the postganglionic injury case were nearly 10.5% greater than the preganglionic. Predicted bone deformations were more severe in the postganglionic case, with the glenoid more declined (pre: -43.8°, post: -51.0°), flatter with higher radius of curvature (pre:3.0mm, post:3.7mm), and anteverted (pre: 2.53°, post: 4.93°) than the preganglionic case. These simulated glenoid deformations were consistent with previous experimental studies. Thus, we concluded that the differences in muscle mass and length between the preganglionic and postganglionic injuries are critical mechanical drivers of the altered glenohumeral joint shape.

Introduction

BPBI is the most common nerve injury among children (Foad et al., 2008), caused by neck hyper extension of the infant during childbirth. The C5-C6 nerve roots of the brachial plexus (Lagerkvist et al., 2010) are most commonly affected, leading to paralysis of the shoulder and elbow (Pearl et al. 2009). While many affected children spontaneously recover nerve and muscle function, 20% to 30% do not have total neurological recovery (Pondaag et al., 2004). As many as 33% of children affected with BPBI sustain permanent postural and bone deformities as a result (Pearl et al. 2009), greatly impairing quality of life (Bae et al., 2008). Postural deformities include shoulder internal rotation and elbow flexion contractures (Waters, 2005). Concomitant altered bone growth in the upper limb can include retroversion, declination, and flattening of the glenoid fossa (Sibinski et al., 2010; Eismann et al., 2016) which is associated with partial humeral dislocation and severe shoulder instability (Kambhampati et al., 2006; Di Mascio et al., 2011). Contractures are significantly correlated with bone deformities at the glenohumeral joint (Pearl and Edgerton, 1998; Kozin, 2004; Bhardwaj et al., 2013). Importantly, presentation in patients depends in part on the location of the nerve injury relative to the nerve root ganglion, with more severe contractures and bone deformity arising when nerve root ruptures occur distal to the ganglion (postganglionic) (Waters, 2005), in contrast to avulsions proximal (preganglionic) which typically result in paralysis without contractures at the shoulder and elbow or significant bone deformity (Al-Qattan 2003).

Understanding the timing and mechanism of deformity initiation is essential for developing treatments that prevent bone and postural deformities from developing, rather than correcting deformity after formation. Because almost nothing is known about how bone tissue deformity first initiates, there is currently no consensus on the optimal timing or treatment type (nerve repair or other therapy) post-injury (Belzberg et al., 2004; Hale et al., 2010), leading to inconsistent clinical outcomes and poor quality of life for children and their families who suffer from lack of information and guidance on the best treatment plan (DeMatteo et al., 2014). Animal and computational studies suggest that altered longitudinal growth of paralyzed muscles crossing the shoulder may explain both contracture and bone morphological changes in postganglionic BPBI (Crouch et al., 2014; Crouch et al., 2015). It is well known that bone shape and density are driven by mechanical environment (Duncan and Turner, 1995), but there are other possible drivers of growth and development which could be negatively affected by either disuse or by the direct nerve injury to bone that occurs in BPBI. Shorter optimal muscle lengths on the affected limb in rats and mice following a postganglionic neurectomy (Nikolaou et al., 2011; Crouch et al., 2015) may be associated with higher passive muscle forces. Multibody dynamic simulations (Crouch et al., 2014; Cheng et al., 2015) suggest that these reduced optimal muscle lengths are sufficient to lower the external rotation range of motion and redirect the glenohumeral joint reaction forces posteriorly. Our recent work using finite element analysis to predict morphology of the glenoid following BPBI suggests that declination and flattening of the glenoid fossa will be more severe under static joint loading (as with paralysis) and with increased restriction optimal muscle length growth compared to normal muscle lengths (Dixit, McFarland et al., 2019). Thus, increased static joint reaction loads due to the injury could cause the morphological changes at the glenohumeral joint.

Earlier work in mice suggested that longitudinal muscle growth is preserved after preganglionic injury, based on reduced joint contractures, and retained ErbB signaling and spindle formation in biceps (Nikolaou et al., 2015). However, more recent work by our group (Dixit et al., 2019) in a rat model of BPBI reveals that optimal muscle lengths in affected muscles were in fact shorter after preganglionic injury. Notably, however, the affected muscles in this injury case were markedly more atrophied than muscle in the contralateral unimpaired limb (Dixit et al., 2019) than muscles affected by a postganglionic injury (Crouch et al 2015). Muscle mass and length are known to be related to force production capacity of the muscle (Zajac, 1989), and our prior simulation work revealed that structural changes to these muscles was predictive of restricted shoulder range of motion in postganglionic but not preganglionic BPBI at 8 weeks of rat age (ref). Moreover, the skeletal muscle is the fastest growing protein mass in neonates (Davis and Fiorotto, 2009) and growing skeletal size at this neonatal stage requires increasing muscle length in healthy development. Therefore, this period of rapid musculoskeletal growth produces continual changes in the force production capacity of the muscles.

Therefore, the goal of this study is to evaluate whether changes to both muscle mass and optimal muscle length over time following BPBI provide a mechanically-driven explanation for the observed differences in postural and bone deformity between preganglionic and postganglionic BPBI. We answer this question using a computational framework integrating musculoskeletal modeling to represent muscle changes over time and finite element modeling to simulate bone growth in response to mechanical and biological stimuli.

Methods

To evaluate the effects of impaired longitudinal growth and muscle atrophy on the growth of bone over time, we implemented muscle changes measured in a rat model of BPBI in an integrated iterative musculoskeletal modeling (I2M2) framework (Figure 5.1). The approach incorporates muscle architecture properties such as muscle length and muscle mass as an input to a multibody dynamic simulation (OpenSim 3.3, Stanford University) to predict shoulder joint reaction forces and apply these loads in a finite element analysis (Abaqus 6.13.3, Dassault Systemes) to predict bone growth in response to biological and mechanical stimuli.

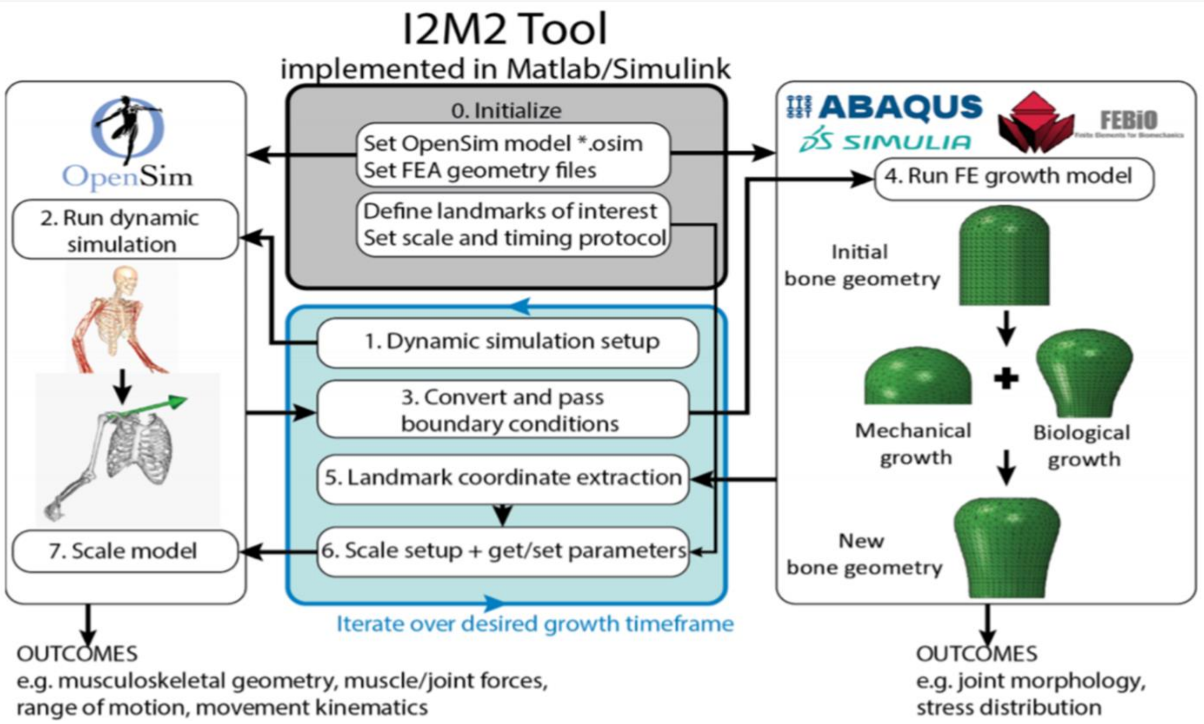

Figure 5.1: Integrated iterative musculoskeletal modeling tool performs cosimulation between OpenSim and Abaqus. Forward dynamic simulations in Abaqus predicts joint reaction forces which are used by Abaqus as boundary conditions for its bone growth computation. The changes to the bone shapes are communicated back to OpenSim so that it rescales the geometries for next cycle of simulations.

To capture the time dependent changes inherent in a growing musculoskeletal system, at each time point, new bone geometry is returned to the multibody simulation to scale the musculoskeletal model and inform the new joint reaction force. These closed-loop simulations (Matlab R2016b, Mathworks) predict the progression of joint deformation as a response to changing joint loads over time. Specifically, we simulated bone growth under a static postganglionic BPBI and preganglionic BPBI of a neonatal rat shoulder, for which quantitative descriptions of bone deformities are available (Li et al., 2010; Crouch et al., 2015); rat and human shoulder anatomy are known to be similar (Norlin et al., 1994). We simulated the growth of the humeral head and scapular glenoid over 8 weeks of rat growth (equivalent to 5.8 human years) (Crouch et al., 2015) from a baseline undeformed initial condition representing 0 days postnatal. The initial geometry representing a neonatal stage for the simulation was obtained by isometrically reducing geometry obtained via micro CT in a typically developed 8 week old rat to match bone lengths of a neonatal rat.

Multibody forward dynamic simulations of the glenohumeral joint were implemented in OpenSim 3.3 (Delp et al., 2007) using an existing musculoskeletal model as a foundation (Saul et al., 2015). Twenty actuators representing muscles crossing the glenohumeral joint were included in the model; we simulated denervation of muscles affected by C5-C6 level BPBI (deltoids, biceps long head, biceps short head, subscapularis, supraspinatus, infraspinatus, teres minor and teres major) (Crouch et al., 2014; Cheng et al., 2015; Hogendoorn et al., 2010; Waters et al., 2009) by restricting activation level to 0 (passive contributions only). Unaffected muscles were activated at 30% (Waters et al., 2009). The muscle mass and the optimal length (resting length) of the affected muscles showing significant changes relative to the unaffected limb in both injury setups were obtained from our previous studies performed in a rat model of BPBI (Dixit et al., 2019). Spinal deltoid was incorporated in our musculoskeletal model in place of the posterior deltoid muscle actuator due to its anatomical similarity. Since these muscle changes were at the end of 8 weeks of rat age (Table 5.1), their effect was implemented in the simulations by incrementing linearly from 0 to 8 weeks over 15 simulation cycles to represent 8 weeks of musculoskeletal growth. We assumed the ratio of muscle mass to optimal muscle length to be directly proportional to the changes in the muscle physiological cross-sectional area (PCSA) thus affecting the maximum isometric force development capacity of the muscle. Similarly, changes in the optimal muscle length in the musculoskeletal model were directly applied by reducing the optimal muscle length by the % restriction in the optimal muscle length of the affected muscles relative to the unimpaired side (Table 5.1). Joint loads were computed with the shoulder in 30° abduction and were scaled using a force reduction factor based on a ratio of average PCSA of subscapularis and infraspinatus in neonatal rats to humans (Swan et al., 2014; Crouch et al., 2014). Force direction and magnitude calculated in the simulations were applied as displacements in the finite element growth model. Our previous work describing simulations of glenohumeral joint growth using finite element analysis showed that the joint stiffness during 8 weeks of bone development remains within 10% of the joint stiffness the initial stage (Dixit et al., 2019). Therefore, the resultant joint reaction forces were converted into corresponding displacements using a constant joint stiffness of 0.7 N/mm (Giorgi et al., 2014).

Table 5.1: Changes in muscle mass and muscle length used as input in the computational simulations obtained from previous experimental studies (Dixit et al., 2019)

	Muscles -->	Deltoids	Spinal deltoid	Pectoralis	Bicep long	Bicep short	Subscapularis	Supraspinatus	Infraspinatus	Teres major	Tricep long
Preganglionic	% muscle mass affected growth	71	64.6	-	76.8	-	70	83.6	78.8	54.5	23.52
	% restriction in muscle length	-	22.3	-	22.3	20	-	32.4	-	29.3	-
Postganglionic	% muscle mass affected growth	38	41.5	25.3	54.3	-	24	-	-	36.9	42.1
	% restriction in muscle length	-	14.5	11.2	-5.7	-	9	-	-	-	-

Joint loading boundary conditions were applied in an iterative finite element modeling approach (Dixit et al., 2019) to predict the bone morphological growth in response to biological and mechanical stimuli (Abaqus 6.13.3, Dassault Systemes). Initial geometry for all simulations was derived from an existing micro CT scan of an uninjured neonatal rat (Crouch et al., 2015). The geometry was reduced to include only the glenoid and humeral head articulating surfaces to improve computational efficiency (Figure 5.2).

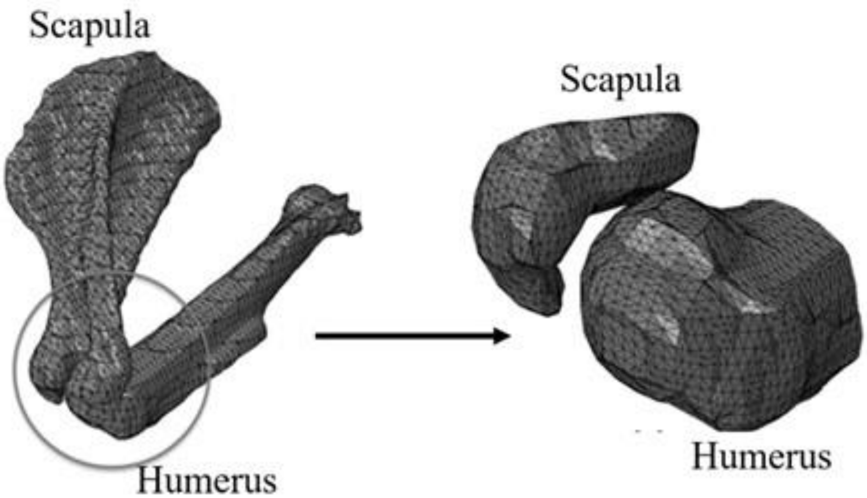

Figure 5.2: The simulations were performed on a reduced glenohumeral geometry to make the model computationally efficient.

Because these neonatal bones are mostly cartilaginous in the region of interest, a cartilage material model was applied, with a Young's modulus of 1.1 MPa and Poisson's ratio of 0.45 (Giorgi et al., 2014). The geometry was meshed using quadratic tetrahedral elements (type C3D10M) with smaller elements near the articulating surface. The final glenoid mesh contained 6447 elements with 9822 nodes and the final humeral head mesh contained 4730 elements with 7109 nodes based on the results of a convergence study from our previous work. A frictionless impenetrable surface-to-surface contact was defined between the two articulating surfaces.

Biological growth was applied to be isometric, with constants applied such that the contribution of biological growth component was 75-85% of the total growth (Germiller and Goldstein, 1997). Mechanical growth was proportional to the compressive hydrostatic stress at every node. Stress due to static loads restricted the growth (Giorgi et al., 2014) according to the following equations:

$$\dot{\varepsilon}_m = C_d \left(\frac{\sum_{i=1}^{N} \sigma_{hi}}{N} \right) \qquad (2)$$

where $\dot{\boldsymbol{\varepsilon}}_{\mathbf{m}}$ is the mechanical growth rate, $\boldsymbol{\sigma}_{\mathbf{hi}}$ is the compressive hydrostatic stress, N is the number of movements per step, and $\mathbf{C_d}$ is the chondrocyte density as determined from the biological calculation above (Eq. 1). Thermal expansion was used as an analog to incorporate deformations due to biological and mechanical growth. The growth rates determined from the biological and mechanical governing equations were used to define temperature gradients to identify nodal deformations, and the deformations were summed to create the resultant deformed geometry.

The glenohumeral geometry after each cycle of finite element simulation was captured the medial-lateral, superior-inferior, and anterior-posterior aspects of the glenoid cavity and humeral head by measuring specific nodal landmarks along the three cartesian axes of the geometry (Figure 5.3). The new positions of the nodal coordinates at the end of each finite element analysis cycle in Abaqus were stored and sent to a scaling algorithm in OpenSim which scaled the musculoskeletal model according to the change in the distance between landmarks before and after a simulation cycle. The muscle optimal length and mass of the affected muscles were then updated in the scaled musculoskeletal model as described earlier for either the preganglionic or postganglionic case (Table 5.1). A forward dynamic simulation with the revised model was then performed to obtain new glenohumeral joint loads, and the loads were used as a boundary condition for the next finite element iteration. In total, we performed 15 such iterative growth cycles to simulate 8 weeks of postnatal growth.

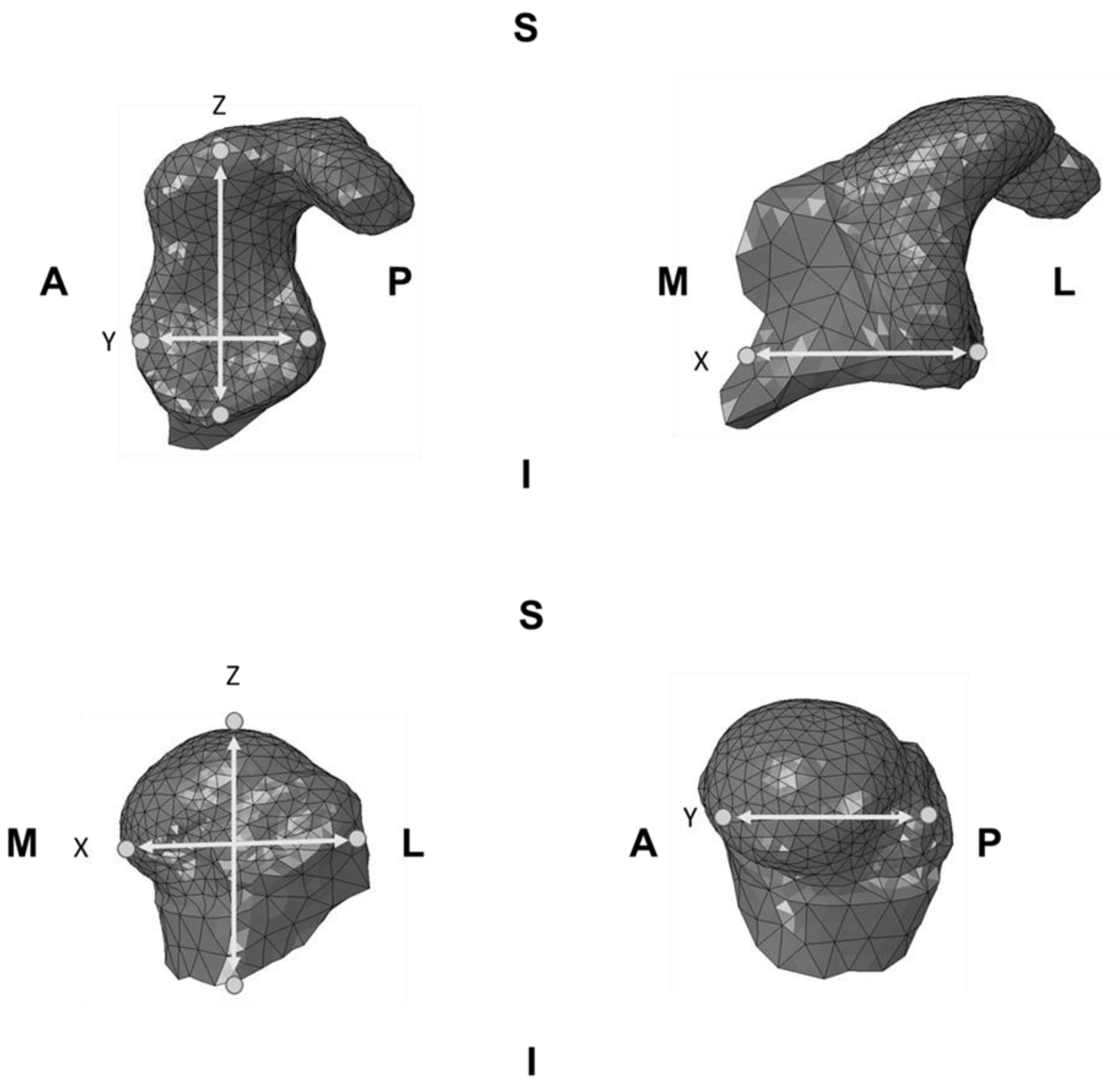

Figure 5.3: Landmarks for scaling. Bone geometry after the growth simulation in Abaqus were communicated to OpenSim for linear segment scaling by measuring the changes in critical nodal landmarks. S – superior, I – inferior, A – anterior, P – posterior, M – medial, L – lateral.

The final glenoid geometries for each simulation were assessed using glenoid inclination angle (GIA), glenoid radius of curvature (GRC), and glenoid version angle (GVA) after simulation cycles 0, 5, 10, and 15. GIA is the angle complementary to the angle between the centerline of the spine of the scapula and the tangent line to the rim of the glenoid cavity (Figure 5.4a). GRC is measured by fitting a circle to the glenoid curvature (Figure 5.4b). GVA is the angle complementary to the angle between the centerline of the scapula in the transverse plane and the tangent line along the cavity (Figure 5.4c). The GIA, GRC and the GVA predicted from the I2M2 simulations were compared with existing geometry derived from micro CT at the same 8-week timepoint in rats due to the preganglionic and postganglionic injuries (Dixit, McCormick, Cole et al., 2019). To evaluate the difference in the glenohumeral joint force production under the simulated preganglionic and postganglionic injury cases, we used the joint force analysis tool in OpenSim to predict the reaction loads at the glenohumeral interface.

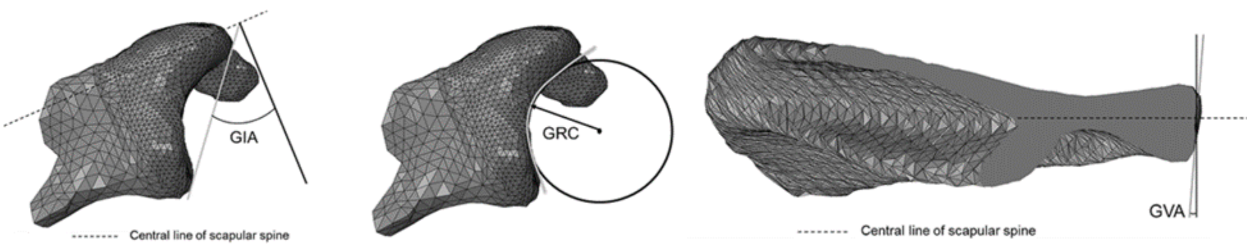

Figure 5.4: Bone deformations were quantified by measuring (a) GIA, (b) GRC, and (c) GVA

Results

GIA at the end of 8 weeks was more declined in the postganglionic case (-51.0°) compared to preganglionic case (-43.8°). GIA at every timepoint from 0 to 8 weeks (1-15 simulation cycles) was predicted to be more declined (Figure 5.5 a,d) in the postganglionic case by 4.2% after 5 cycles, 5.8% after 10 cycles, and 16.4% after 15 cycles compared to preganglionic case with same starting geometries at the 1st cycle.

Simulations predicted higher GRC, or flatter glenoids, at 8 weeks in postganglionic injury simulation case (3.7mm) than in the preganglionic case (3.0mm). Simulations predicted flatter glenoid in the postganglionic case by 20% after 5 cycles, 16.7% after 10 cycles, and 23.3% after 15 cycles compared to preganglionic case at every timepoint from 0 to 8 weeks (Figure 5.5 b, d).

At the end of 8 weeks simulations predicted more anteverted glenoid in the postganglionic case (4.93°) compared to preganglionic injuries (2.53°) (Figure 5.5 c, d).

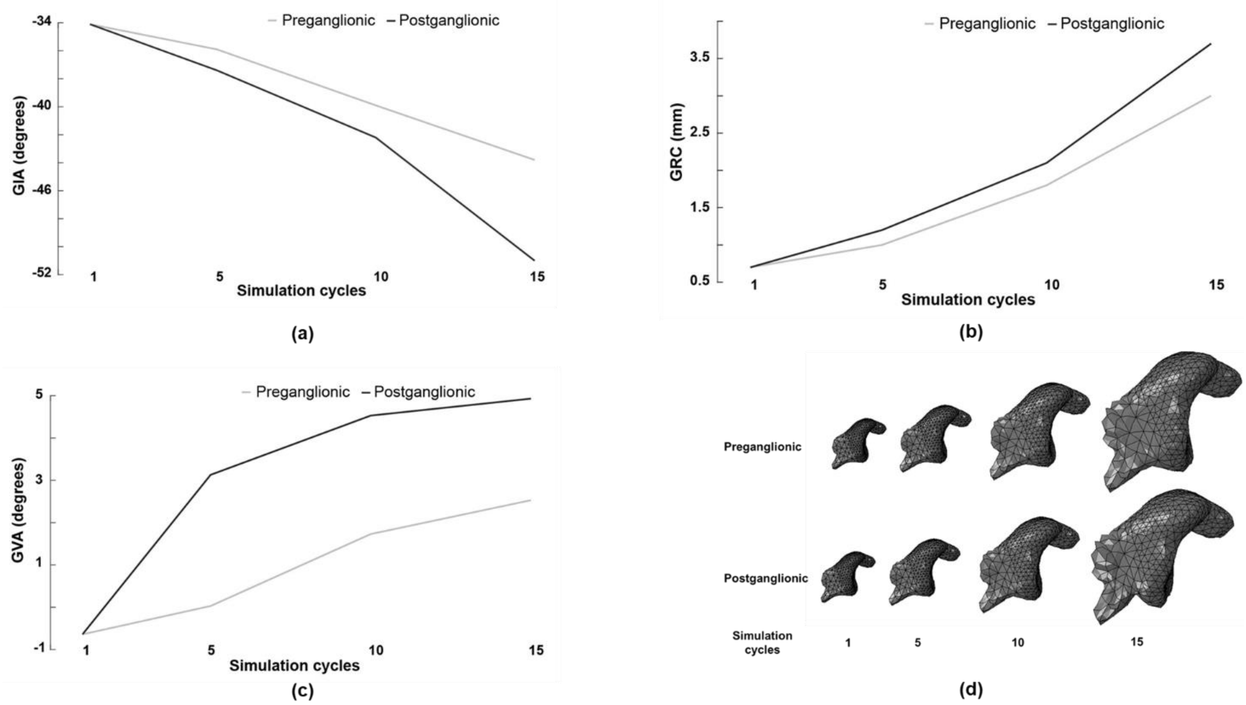

Figure 5.5: Glenoid measurements after preganglionic and postganglionic injures show changes in GIA (a), GRC (b), and GVA (d) from 0 to 8 weeks of growth. Timeline of the glenoid morphological changes over 8 weeks shows the deformation in glenoid due to preganglionic and postganglionic BPBI (d).

The net glenohumeral joint reaction forces in the preganglionic case were consistently lower at every timepoint than the postganglionic case. The net joint reaction forces in the postganglionic case were consistently higher by 4.4% after 5 cycles, 11.7% after 10 cycles, and 8.4% after 15 cycles than the preganglionic case at same timepoint (Figure 5.6).

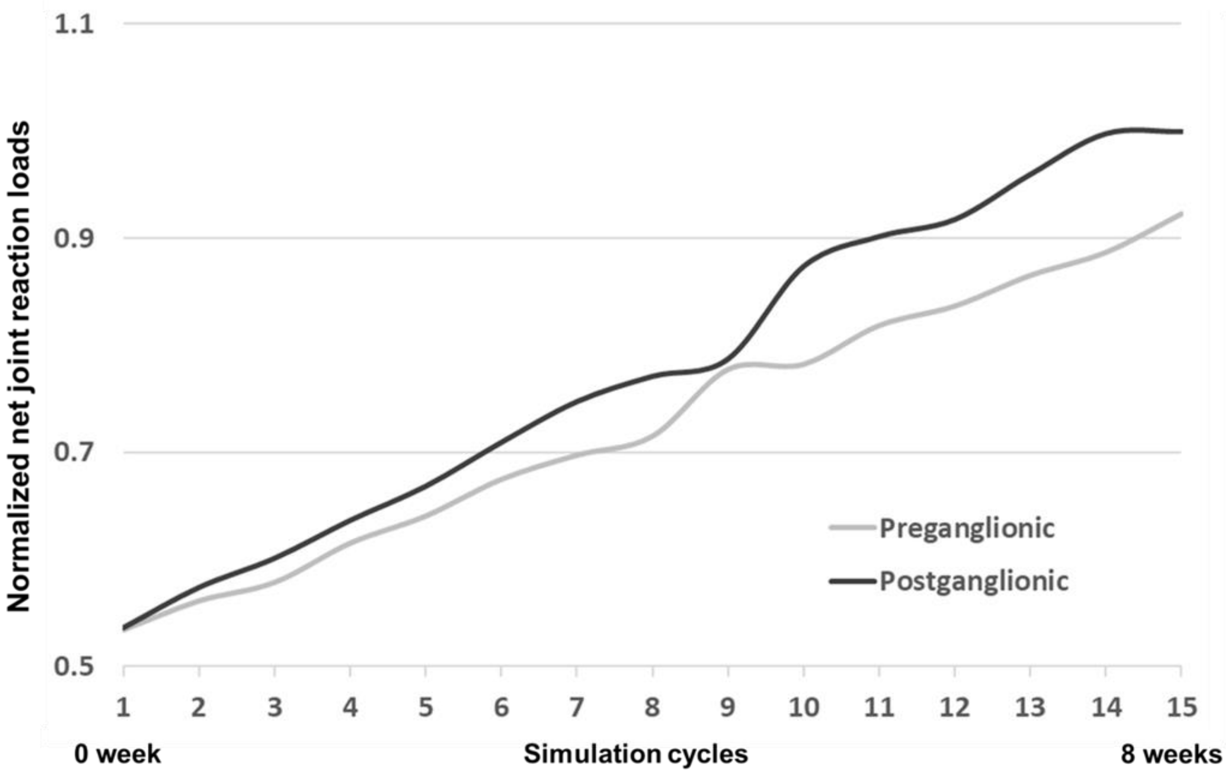

Figure 5.6: Resultant magnitudes of glenohumeral joint loads were lower in preganglionic case than the postganglionic cases over 15 simulation cycles representing 8 weeks of growth

Discussion

We examined whether and to what extent changes in muscle mass and optimal muscle length following preganglionic and postganglionic BPBI influences glenohumeral joint morphogenesis, elucidating the mechanical factors responsible for joint deformity. The computational analyses predicted that net glenohumeral joint loading in the simulations was higher in the postganglionic case than preganglionic and were consistent with more severe predicted bone deformations, suggesting that the altered mechanical loading could account for observed changes in glenoid morphology after BPBI. Our past simulations (Crouch et al., 2014) also revealed that these muscle changes were drivers of increased postural deformity in postganglionic but not preganglionic BPBI. Thus, taken together, these simulations reveal that the mechanical underpinnings of both postural and bone deformity can be linked to altered structure of muscle. This study also employed a new computational framework that integrated musculoskeletal dynamics with finite element simulations of growth to capture the interactions of tissue growth and mechanical loading over time.

The joint reaction loads in both injury cases increased from 0 to 8 weeks however, the increment in the loads was lower in the preganglionic condition than the postganglionic through the simulated 8 weeks. Even though the preganglionic condition had slower increase in the optimal muscle length, the muscle atrophy representing slower increase in muscle mass was also remarkably greater than the postganglionic condition (Dixit et al., 2019). The force developed by a muscle-tendon unit depends both on its length and on its cross-sectional area (Zajac, 1989). The shorter optimal muscle length causes the joint reaction forces to increase (Cheng et al., 2015), especially the passive force component due to increased stretch in the muscles. However, slower increase in muscle mass representing the muscle atrophy substantially reduced the passive force generation of the affected muscles; thus, this muscle atrophy which was higher in the preganglionic case contributed to an overall lower resultant joint load. Since the joint loads drop over time in both the injury cases, it could be because of the slower increase in muscle mass due to the nerve injury playing a dominant role than the slower increase in optimal muscle lengths thereby contributing towards the reduction of overall joint load (Dixit et al., 2019). Difference between the loads in the two injuries cases also increase over time. This further corroborates our observation on muscle atrophy playing a dominant role in governing joint loads as the preganglionic injuries shows significantly higher muscle atrophy and therefore higher lack of growth than postganglionic injuries (muscle paper).

The static nature of joint loading arising from muscle paralysis in BPBI is known to be an important driver of bone deformity (Dixit et al., 2019). Our simulations suggest that there are higher declination (negative GIA), higher flatness (higher GRC), and higher retroversion (negative GVA) in the postganglionic case, which also experiences higher joint loads at the glenohumeral interface. Clinical studies in humans (Al-Qattan, 2003) and rats (Dixit et al., 2019) have shown that postganglionic injury is associated with more severe glenoid deformation with higher declination, radius of curvature, and version than preganglionic injuries. Postganglionic injuries have been seen to develop severe shoulder contractures in previous animal (Nikolaou et al. 2011; Crouch et al. 2015) and computational (Cheng et al 2015) work. We observed greater ROM in preganglionic simulations compared to postganglionic condition indicating lower shoulder contracture in preganglionic case, consistent with previous clinical studies (Dixit et al., 2019; Al-Qattan, 2003).

The study provides a predictive framework for investigating the effects of injury and therapy on developing osseous deformation, which, going forward can assist clinicians in deciding the time line and targets for most effective intervention. For example, currently, even with surgical strategies such as external rotation tendon transfer (Eismann et al., 2016) or glenoid osteoplasty or osteotomy (Di Mascio et al., 2011), there is no consensus among surgeons as to whether these deformities are reversible after surgical intervention or not (Hogendoorn et al., 2010). This framework provides an opportunity to explore how bone deformation pattern would progress over time. Similarly, it may provide guidance for intervention timing. For example, GIA - one of the primary measures in quantifying glenoid deformities due to BPBI (Dixit et al., 2019) deforms at a different rate in the two injury scenarios (Figure 5.6), suggesting a varying rate of tissue change at odds with the wait-and-see approach in clinical practice (Patra et al., 2016; Hale et al., 2010). A surgical or physical therapy intervention within this time period that alters the direction and magnitude of the net joint reaction force may be a possible approach to limit progression of bone deformity.

Severe shoulder and elbow contractures have been reported in previous clinical findings (Al Qattan 2003) due to postganglionic injuries whereas the preganglionic injuries does not exhibit any contractures at the shoulder or elbow. Our previous study examining the passive external rotation in preganglionic and postganglionic BPBI affected rats showed similar observations in which postganglionic injuries in rats restricted passive external shoulder rotation (Crouch et al. 2015) We have been successful in predicting the same phenomenon through musculoskeletal simulations. While previous computational studies examined only the effect of muscle shortening on the shoulder ROM, our study found that both muscle mass and muscle shortening are critical parameters governing the shoulder contracture in the two injury scenarios.

The results of these computational simulations must be considered in light of the limitations. Muscle architecture due to the preganglionic and postganglionic BPBI were obtained at 8 weeks in rats (Dixit et al., 2019). These changes to the muscles were implemented in simulations from 0 to 8 weeks through linear incrementation. The current simulations have been validated for the 8-week time point. However, additional experimental data is needed to validate and refine at intermediate time points. Joint loading was obtained from a scaled musculoskeletal model using the human glenohumeral joint as a foundation; while muscle and joint geometry and osseous morphological response to BPBI is very similar in rat and human shoulders (Norlin et al., 1994), they are not identical. Development of a rat forelimb musculoskeletal model should be pursued to improve representation of muscle and joint forces in future work. Joint stiffness, which was used to convert joint forces to equivalent displacements for application in the finite element models, was kept constant in all simulations and over time. It is possible that the effective stiffness may vary with condition or time, however in our previous study (Dixit et al., 2019) the joint stiffness under morphogenesis was found to be within 10% variation with the stiffness of the initial geometry. Although cartilage is a biphasic material (Roddy et al., 2011), we modelled cartilage as a single phase nearly incompressible material (Poisson's ratio 0.49) (Giorgi et al., 2014). Previous studies (Carter and Wong, 2003; Shefelbine and Carter, 2004) have shown that the fluid pressure in biphasic models and hydrostatic stress in single phase models are similar when loaded at 1 Hz, which is approximately the frequency of muscle contraction (Vaal et al., 2000). In this study, static simulations did not have applied loading at 1 Hz. However, prior work in the simulation of joint morphogenesis using a single-phase model were successfully extended to predict bone development under hip dysplasia under both static and dynamic loading (Giorgi et al., 2015; Giorgi et al., 2014). Additional future work is needed to capture the effects of incorporating the single phase material in this framework when loading occurs under varied loading frequencies. The bones also ossify over time. We used a cartilaginous material model over the entire 8 weeks of growth simulation, however some ossification occurs during this period; future simulations could incorporate evolving material models over time to address this limitation. The effect of synovial fluid at the joint does not play a crucial role in defining joint morphology (Giorgi et al., 2014); hence we did not model the synovial fluid interface in our simulations. We modeled isotropic biological growth for the glenoid of the scapula, but there are multiple ossification centers at this stage of development (Nougarolis et al., 2017), and each may have different biological growth rates. Differing growth rates along the glenoid cavity may contribute to the differences observed between predicted and experimental GRC. We reported only changes in the glenoid. We do not report changes to the humeral head, however, flatness at the humeral head surface due to the injury has been reported in previous studies (Sibinski et al., 2010). The simulations captured the less severe glenoid deformations in the preganglionic compared to the postganglionic. However, preganglionic injury did not exhibit contractures at the shoulder in our experimental studies and some previous clinical work (Al-Qattan, 2003), so a fully static shoulder in the simulations is likely a conservative estimate of loading case in this injury case. To investigate the role of altered mechanical environment on the bone growth, the biological growth was held constant across all the simulations, but the nerve injury may affect the biological growth directly and its effect on glenoid as well as humeral head shape should be investigated. The simulations to predict the ROM can be improved further by involving all the affected muscles due to the injury from our previous study even though not significantly affected which could have been due to limited sample size (Dixit et al., 2019). The study involved 30% activation from all the other muscles on the arm which are not crossing the shoulder which can create a variation in the results due to the muscles active contributions.

In conclusion, we found that the changes to muscle mass and optimal muscle length over time due to BPBI are sufficient to explain the observed differences in bone and postural deformity between preganglionic and postganglionic BPBI. Using this computational framework, we predicted the time progression of these osseous deformities and found that these are driven by the altered joint reaction loads generated due to the changes in muscle mass and optimal muscle length. The implementation of this framework is not limited to glenohumeral joint growth simulations due to BPBI but it can be used to predict growth for other joints of the body following an injury or a disease.

CHAPTER 6: Conclusion

This dissertation comprehensively analyzed the mechanical loading of the shoulder that contribute to shoulder deformity after BPBI. In particular, this work characterized the effects of both postganglionic and preganglionic injuries on musculoskeletal development that were previously undescribed in either the clinical or basic science literature. The experimental data was incorporated in a novel computational framework to directly predict bone growth over time and the interactions of bone and muscle growth in BPBI. This work can have immediate clinical impact by identifying that passive loading and restricted longitudinal growth play a significant role in altering bone growth and may be important potential targets of intervention. This dissertation also has broad implications for future investigation, including predicting the effects of interventions (e.g. timing or type of nerve, muscle, or bony surgical intervention), or predicting joint growth for other joints of the body following injury or disease. Specific contributions are summarized below.

In Aim 1, this research developed a finite element based iterative model which predicted that static loading of the joint is primarily responsible for joint deformation due to BPBI, whereas dynamic loading will result in typical joint growth. The model captured the effects of altered muscle forces due to paralysis and restricted longitudinal growth on the glenoid and predicted that restricted longitudinal growth results in more severe changes to the glenoid. The computational analyses predicted morphological changes to glenoid that were consistent with experimental data describing glenoid morphology after BPBI in a rat model.

In Aim 2, experimental studies were conducted in a Sprague Dawley rat model of BPBI which provided crucial new information regarding muscle architectural changes following preganglionic and postganglionic BPBI. While muscle mass loss and reduced optimal muscle length are present for both injury locations, the effects are significantly more severe in preganglionic BPBI. However, contractures at the shoulder are more severe in postganglionic BPBI compared to preganglionic, which is consistent with previous clinical studies in humans. The findings from this work suggest that contracture derives not only from restricted muscle growth, but also depends on the extent of muscle mass loss occurring simultaneously following the injury.

In Aim 3, this dissertation clearly identifies differences in glenohumeral osseous deformities following preganglionic and postganglionic BPBI in the same rat model. Glenoid deformities were more severe after postganglionic injury, with markedly altered shape profile, while preganglionic injury exhibited more pronounced humeral head alterations, with an overall smaller head.

Finally, in Aim 4, a new computational framework was developed integrating musculoskeletal and finite element modeling in one comprehensive analysis to capture morphological changes to the glenohumeral joint in response to these altered muscle properties. This Aim found that the post-injury changes to muscle mass and optimal muscle length over time are sufficient to explain the observed differences in bone and postural deformity between preganglionic and postganglionic BPBI. The computational framework predicted the time progression of these osseous deformities and found that these are directly associated with the altered joint reaction loads generated due to the changes in muscle mass and optimal muscle length.

Primary conclusions

The main conclusions of this dissertation are:

- Demonstration through a finite element-based iterative growth model that static loading is primarily responsible for altered glenoid formation, and restricted longitudinal growth compounds these effects.

- Muscle mass loss and reduced optimal muscle length are present for both preganglionic and postganglionic injuries, but the effects are significantly more severe in preganglionic BPBI.
- Contracture derives not only from restricted muscle growth, but also depends on the extent of muscle mass loss occurring simultaneously following the injury.
- Glenoid deformities are more severe after postganglionic injury, while preganglionic injury exhibits more pronounced humeral head alterations, with an overall smaller head.
- A new computational framework that integrates musculoskeletal and finite element modeling was developed, and demonstrated that changes to muscle mass and optimal muscle length over time due to BPBI are sufficient to explain the observed differences in bone and postural deformity between preganglionic and postganglionic BPBI.

Applications

Specialized treatment strategies for preganglionic and postganglionic injuries: The effects of preganglionic and postganglionic BPBI on muscle architectural changes were compared in Chapter 3. It was found that the specific muscles that are most affected in the two injuries are different, and the severity of the overall muscle degeneration is more severe after preganglionic injury. This work provide guidance regarding which muscles should be potential targets for treatment intended to restore muscle function through surgical intervention or future pharmacological interventions.

Use of I2M2 simulation for treatment simulation: Chapter 5 performed I2M2 simulations to predict the effect of altered muscle architecture due to preganglionic and postganglionic BPBI on joint deformation. This framework presents a flexible platform on which to examine the effects of therapeutic interventions designed to restore shoulder function or alter joint loading. For example, common interventions to restore muscle balance (e.g. tendon transfers) or reduce the joint surfaces (e.g. bone osteotomies) could be represented and the effects on joint morphology explored. Because of the time-dependent iterative nature of the platform, the influence timing of the interventions could further be probed. This capability presents a new and transformative avenue to examine the expected outcomes of various surgical or other interventions to identify the most optimum treatment option.

I2M2 framework for other mechanical engineering applications: The I2M2 simulation framework is flexible, and thus is not necessarily limited to predicting joint deformations in a musculoskeletal system; this framework can also be extended to predict phenomenon such as material wear in a mechanical system. A pilot study was conducted through a collaboration with Schneider Electric (Cedar Rapids, Iowa), in which a new Python script was developed to integrate the Archard's wear law in the I2M2 framework.

$$Q = \frac{KWL}{H}$$

where;

Q is the total volume of wear debris produced

K is a dimensionless wear constant

W is the total normal load

L is the sliding distance

H is the hardness of the softest contacting surfaces

The simulations were performed on a trip lever and armature latch assembly of the Schneider's QOB3125VH miniature circuit breaker mechanism. For a constant unit load, two types of trip lever materials (thermoplastic and mild steel) were simulated against the harder cold rolled steel of the armature. As shown in the Figure 6.1, the wear of the thermoplastic trip lever was found to be greater than the steel trip lever. Such implementation of the I2M2 framework can be used as a critical tool to test different materials and optimize the product's performance for its design life.

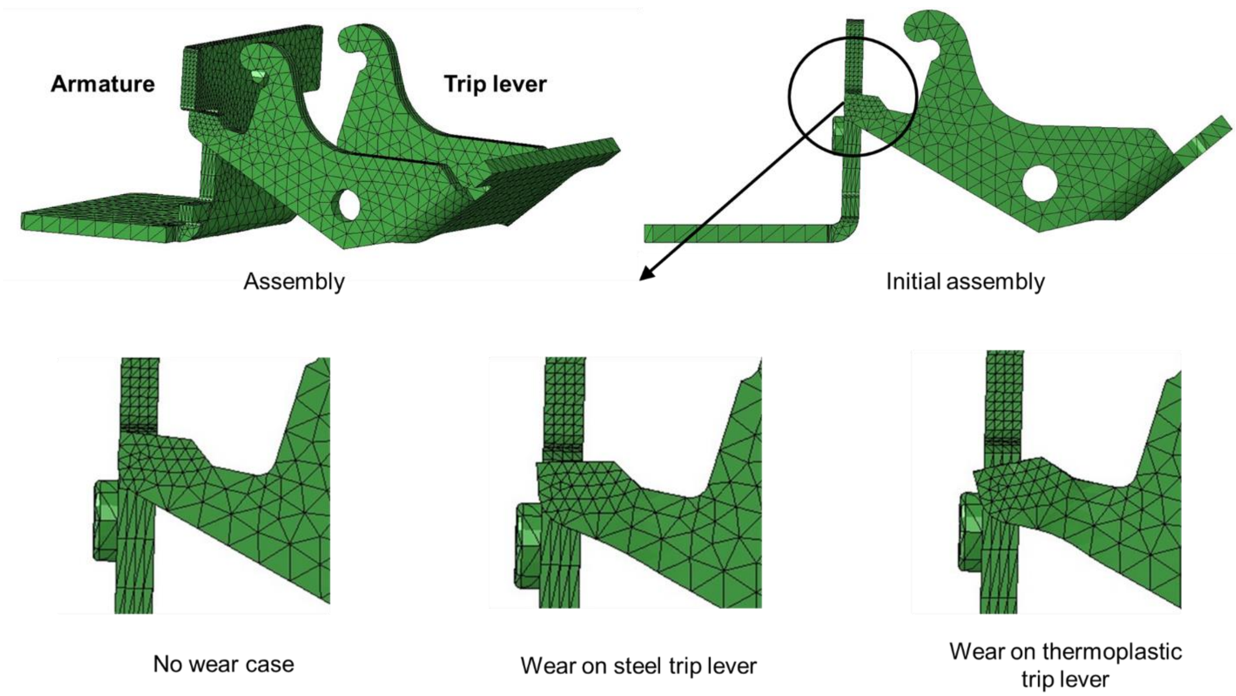

Figure 6.1: Wear of thermoplastic trip lever greater than that of the one made of steel

Future work

Determine timepoint and mechanisms for onset of shoulder deformity following BPBI. Understanding the timing and mechanism of deformity initiation is essential for developing treatments that prevent bone and postural deformities from developing, rather than correcting deformity after formation. Our group's preliminary work shows gross alterations in both muscle and bone by 4 weeks postnatal in rats, suggesting early tissue changes at odds with the wait-and-see approach in clinical practice. The experimental work described in this dissertation could be extend to investigate multiple earlier timepoints of the rat's muscle and bone development following preganglionic and postganglionic injuries. Such a study would identify the timepoint of the onset of shoulder deformity and its progression over time, as well as clarify in what order alterations to the macro and microstructural characteristics of muscle and bone occur.

Determine whether drivers of deformity in BPBI are present for postnatal pediatric traumatic brachial plexus injury (PTBPI). PTBPI occurring outside the neonatal period presents with limited contracture, compared to BPBI where shoulder contracture and its implications for bone deformity is a substantial concern. PTBPI provides a unique opportunity to explore why the same injury in a similar period of rapid musculoskeletal growth avoids marked deformity, supplying insight to processes governing muscle and bone growth through childhood and adolescence. This future study will illuminate differences in tissue response to injury that prevents deformity formation, providing potential targets for BPBI treatment.

Developing a rat musculoskeletal model to improvise the accuracy of joint load predictions in rats: The current implementation of the I2M2 framework uses a human musculoskeletal model to predict glenohumeral joint reaction loads due to altered muscle architecture following the BPBI. These loads are scaled to represent the rat musculoskeletal system using a conversion factor. However, a fully developed rat musculoskeletal model would improve the capability of this framework as it would more accurately represent the geometry and force-generating capacity of the rat muscles, including those not present in humans such as spinal deltoid. It would also offer opportunities to directly include measured musculoskeletal parameter from experiment, and directly simulate functional outcomes assessed in rats, such as range of motion or gait alterations.

Summary

Understanding the difference in effects of preganglionic and postganglionic BPBI on the shoulder musculoskeletal system is essential to designing and implementing treatment strategies towards restoring or preserving upper extremity function following the injury. The research in this dissertation provides new knowledge about the changes in muscle characteristics and osseous deformations in the two BPBI variants. Through animal studies, it was found that the negative effect on muscles was more severe on animals affected with preganglionic injuries compared the postganglionic BPBI. However, the contractures and the deformation of the glenohumeral joint was more severe in the postganglionic case. This research developed a computational scheme through which it was found that these changes to muscles due to the injury altered the joint reaction loads, which ultimately mechanically drive deformations at the joint. This research has greatly enhanced our understanding of glenohumeral deformity formation after BPBI, which can have immediate clinical impact in identifying the most significant factors altering the bone growth as potential targets for intervention and potentially predicting the outcomes of surgeries for patients.

REFERENCES

Al-Qattan, M.M, 2003. Obstetric brachial plexus palsy associated with breech delivery. Annals of plastic surgery. 51, 257-264. http://japanknowledge.com/lib/display/?lid=40300SPED510400

Al-Qattan, M.M., Clarke, C., 2000. The prognostic value of concurrent Hoerner's syndrome in total obstetric brachial plexus injury. Journal of Hand Surgery. 25, 166-167. https://www.sciencedirect.com/science/article/pii/S0266768199903510

Bae, D., Waters, P., Zurakowski, D., 2008. Correlation of Pediatric Outcomes Data Collection Instrument With Measures of Active Movement in Children With Brachial Plexus Birth Palsy. Journal of Pediatric Orthopaedics. 28, 584-592. http://ovidsp.ovid.com/ovidweb.cgi?T=JS&NEWS=n&CSC=Y&PAGE=fulltext&D=ovft&AN=01241398-200807000-00018

Belzberg, A.J., Dorsi, M.J., Storm, P.B., Moriarity, J.L., 2004. Surgical repair of brachial plexus injury: a multinational survey of experienced peripheral nerve surgeons. Journal of neurosurgery. 101, 365-376. https://www.ncbi.nlm.nih.gov/pubmed/15352592

Bhardwaj, P., Burgess, T., Sabapathy, S.R., Venkataramani, H., Ilayaraja, V., 2013. Correlation between clinical findings and CT scan parameters for shoulder deformities in birth brachial plexus palsy. The Journal of hand surgery. 38, 1557-1566. http://www.ncbi.nlm.nih.gov/pubmed/23816519

Brochard, S., Mozingo, J.D., Alter, K.E., Sheehan, F.T., 2016. Three dimensionality of gleno-humeral deformities in obstetrical brachial plexus palsy. Journal of Orthopaedic Research. 34, 675-682. http://onlinelibrary.wiley.com/doi/10.1002/jor.23049/abstract

Burkholder, T.J., Lieber, R.L., 2001. Sarcomere length operating range of vertebrate muscles during movement. Journal of Experimental Biology. 204, 1529. http://jeb.biologists.org/cgi/content/abstract/204/9/1529

Carriero, A., Jonkers, I., Shefelbine, S.J., 2011. Mechanobiological prediction of proximal femoral deformities in children with cerebral palsy. Computer Methods in Biomechanics and Biomedical Engineering. 14, 253-262. http://www.tandfonline.com/doi/abs/10.1080/10255841003682505

Carter, D.R., Orr, T.E., Fyhrie, D.P., Schurman, D.J., 1987. Influences of Mechanical Stress on Prenatal and Postnatal Skeletal Development. Clinical orthopaedics and related research. 237-250. https://www.ncbi.nlm.nih.gov/pubmed/3581576

Carter, D., Wong, M., 2003. Modelling cartilage mechanobiology. Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences. 358, 1461-1471. http://rstb.royalsocietypublishing.org/content/358/1437/1461.abstract

Cheng, W., Cornwall, R., Crouch, D.L., Li, Z., Saul, K.R., 2015. Contributions of muscle imbalance and impaired growth to postural and osseous shoulder deformity following brachial plexus birth palsy: a computational simulation analysis. The Journal of hand surgery. 40, 1170-1176. http://www.ncbi.nlm.nih.gov/pubmed/25847723

Clarke, H.M., Curtis, C.G., 1995. An approach to obstetrical brachial plexus injuries. Hand clinics. 11, 563. https://www.ncbi.nlm.nih.gov/pubmed/8567739

Crouch, D.L., Hutchinson, I.D., Plate, J.F., Antoniono, J., Gong, H., Cao, G., Li, Z., Saul, K.R., 2015. Biomechanical Basis of Shoulder Osseous Deformity and Contracture in a Rat Model of

Brachial Plexus Birth Palsy. The Journal of bone and joint surgery. American volume. 97, 1264-1271. http://www.ncbi.nlm.nih.gov/pubmed/26246261

Crouch, D.L., Plate, J.F., Li, Z., Saul, K.R., 2014. Computational sensitivity analysis to identify muscles that can mechanically contribute to shoulder deformity following brachial plexus birth palsy. The Journal of hand surgery. 39, 303-311. http://www.ncbi.nlm.nih.gov/pubmed/24342260

Davis, T., Fiorotto, M., 2009. Regulation of muscle growth in neonates. Current Opinion in Clinical Nutrition and Metabolic Care. 12, 78-85. http://ovidsp.ovid.com/ovidweb.cgi?T=JS&NEWS=n&CSC=Y&PAGE=fulltext&D=ovft&AN=00075197-200901000-00014

Delp, S.L., Anderson, F.C., Arnold, A.S., Loan, P., Habib, A., John, C.T., Guendelman, E., Thelen, D.G., 2007. OpenSim: Open-Source Software to Create and Analyze Dynamic Simulations of Movement. IEEE Transactions on Biomedical Engineering. 54, 1940-1950. http://ieeexplore.ieee.org/document/4352056

DeMatteo, C., Bain, J.R., Gjertsen, D., Harper, J.A., 2014. 'Wondering and waiting' after obstetrical brachial plexus injury: Are we underestimating the effects of the traumatic experience on the families? Plastic surgery (Oakville, Ont.). 22, 183-187. https://www.ncbi.nlm.nih.gov/pubmed/25332647

Di Mascio, L., Chin, K., Fox, M., Sinisi, M., 2011. Glenoplasty for complex shoulder subluxation and dislocation in children with obstetric brachial plexus palsy. The Journal of bone and joint surgery. British volume. 93, 102-107. http://www.ncbi.nlm.nih.gov/pubmed/21196552

Dixit, N.N., McCormick, C.M., Cole, J.H., Saul, K.R., 2019. Influence of brachial plexus birth injury location on morphology of the glenohumeral joint (In review).

Dixit, N.N., McCormick, C.M., Warren, E., Cole, J.H., Saul, K.R., 2019. Preganglionic and postganglionic brachial plexus birth injury effects on shoulder muscle growth (In review).

Dixit, N.N., McFarland, D.C., Saul, K.R., 2019. Computational analysis of glenohumeral joint growth and morphology following a brachial plexus birth injury. Journal of Biomechanics. 86, 48-54. https://www.sciencedirect.com/science/article/pii/S0021929019300776

Duncan, R.L., Turner, C.H., 1995. Mechanotransduction and the functional response of bone to mechanical strain. Calcified tissue international. 57, 344-358. https://www.ncbi.nlm.nih.gov/pubmed/8564797

Eismann, E.A., Laor, T., Cornwall, R., 2016. Three-Dimensional Magnetic Resonance Imaging of Glenohumeral Dysplasia in Neonatal Brachial Plexus Palsy. The Journal of bone and joint surgery. American volume. 98, 142-151. http://www.ncbi.nlm.nih.gov/pubmed/26791035

Eismann, E.A., Little, K.J., Laor, T., Cornwall, R., 2015. Glenohumeral abduction contracture in children with unresolved neonatal brachial plexus palsy. The Journal of bone and joint surgery. American volume. 97, 112-118. http://www.ncbi.nlm.nih.gov/pubmed/25609437

Flatow, E.L., Cuomo, F., Maday, M.G., Miller, S.R., McIlveen, S.J., Bigliani, L.U., 1991. Open reduction and internal fixation of two-part displaced fractures of the greater tuberosity of the proximal part of the humerus. The Journal of bone and joint surgery. American volume. 73, 1213-1218. https://www.ncbi.nlm.nih.gov/pubmed/1890123

Foad, S.L., Mehlman, C.T., Ying, J., 2008. The Epidemiology of Neonatal Brachial Plexus Palsy in the United States. The Journal of Bone & Joint Surgery. 90, 1258-1264. http://jbjs.org/article.aspx?articleid=28711

Germiller, J.A., Goldstein, S.A., 1997. Structure and function of embryonic growth plate in the absence of functioning skeletal muscle. Journal of orthopaedic research: official publication of the Orthopaedic Research Society. 15, 362-370. http://www.ncbi.nlm.nih.gov/pubmed/9246082

Giorgi, M., Carriero, A., Shefelbine, S.J., Nowlan, N.C., 2015. Effects of normal and abnormal loading conditions on morphogenesis of the prenatal hip joint: application to hip dysplasia. Journal of Biomechanics. 48, 3390-3397. http://www.sciencedirect.com/science/article/pii/S0021929015003395

Giorgi, M., Carriero, A., Shefelbine, S.J., Nowlan, N.C., 2014. Mechanobiological simulations of prenatal joint morphogenesis. Journal of biomechanics. 47, 989-995. http://www.ncbi.nlm.nih.gov/pubmed/24529755

Haerle, M., Gilbert, A., 2004. Management of Complete Obstetric Brachial Plexus Lesions. Journal of pediatric orthopedics. 24, 194-200. http://www.sciencedirect.com/science/book/9781555582890

Hale, H.B., Bae, D.S., Waters, P.M., 2010. Current Concepts in the Management of Brachial Plexus Birth Palsy. Journal of Hand Surgery. 35, 322-331. http://www.sciencedirect.com/science/article/pii/S0363502309010612

Hawkins, R.J., Janda, D.H., 1996. Posterior Instability of the Glenohumeral Joint. The American Journal of Sports Medicine. 24, 275-278. https://journals.sagepub.com/doi/full/10.1177/036354659602400305

Heegaard, J.H., Beaupré, G.S., Carter, D.R., 1999. Mechanically modulated cartilage growth may regulate joint surface morphogenesis. Journal of Orthopaedic Research. 17, 509-517. https://onlinelibrary.wiley.com/doi/abs/10.1002/jor.1100170408

Hogendoorn, S., Van Overvest, Karlijn L J, Watt, I., Duijsens, A.H.B., Nelissen, Rob G H H, 2010. Structural Changes in Muscle and Glenohumeral Joint Deformity in Neonatal Brachial Plexus Palsy. The Journal of Bone & Joint Surgery. 92, 935-942. http://jbjs.org/article.aspx?articleid=5973

Jellicose, P., Parsons, S., 2008. Brachial plexus birth palsy. Journal of Hand Surgery. 289-294. https://www.clinicalkey.es/playcontent/1-s2.0-S0363502309010612

Kambhampati, S.B.S., Birch, R., Cobiella, C., Chen, L., 2006. Posterior subluxation and dislocation of the shoulder in obstetric brachial plexus palsy. The Journal of bone and joint surgery. British volume. 88, 213-219. http://www.ncbi.nlm.nih.gov/pubmed/16434527

Kim, H.M., Galatz, L.M., Das, R., Patel, N., Thomopoulos, S., 2010. Musculoskeletal deformities secondary to neurotomy of the superior trunk of the brachial plexus in neonatal mice. Journal of orthopaedic research: official publication of the Orthopaedic Research Society. 28, 1391-1398. http://www.ncbi.nlm.nih.gov/pubmed/20225283

Kirjavainen, M., Remes, V., Peltonen, J., Kinnunen, P., Pöyhiä, T., Telaranta, T., Alanen, M., Helenius, I., Nietosvaara, Y., 2007. Long-Term Results of Surgery for Brachial Plexus Birth Palsy. The Journal of bone and joint surgery. American volume. 89, 18-26. https://www.ncbi.nlm.nih.gov/pubmed/17200305

Kozin, S.H., 2004. Correlation between external rotation of the glenohumeral joint and deformity after brachial plexus birth palsy. Journal of pediatric orthopedics. 24, 189-193. http://japanknowledge.com/lib/display/?lid=40300SPED510400

Lagerkvist, A., Johansson, U., Johansson, A., Bager, B., Uvebrant, P., 2010. Obstetric brachial plexus palsy: a prospective, population-based study of incidence, recovery, and residual impairment at 18 months of age. Developmental medicine and child neurology. 52, 529. http://www.ncbi.nlm.nih.gov/pubmed/20041937

Li, Z., Barnwell, J., Tan, J., Koman, A.L., Smith, B.P., 2010. Microcomputed Tomography Characterization of Shoulder Osseous Deformity After Brachial Plexus Birth Palsy: A Rat Model Study. The Journal of Bone & Joint Surgery. 92, 2583-2588. http://jbjs.org/article.aspx?articleid=5797

Li, Z., Ma, J., Apel, P., Carlson, C.S., Smith, T.L., Koman, L.A., 2008. Brachial Plexus Birth Palsy–Associated Shoulder Deformity: A Rat Model Study. Journal of Hand Surgery. 33, 308-312. http://www.sciencedirect.com/science/article/pii/S0363502307010398

Lieber, R.L., Yeh, Y., Baskin, R.J., 1984. Sarcomere length determination using laser diffraction. Effect of beam and fiber diameter. Biophysical Journal. 45, 1007-1016. https://www.sciencedirect.com/science/article/pii/S0006349584842460

Lippert, W., Mehlman, C., Cornwall, R., Foad, M., Laor, T., Anton, C., Welge, J., 2012. The Intrarater and Interrater Reliability of Glenoid Version and Glenohumeral Subluxation Measurements in Neonatal Brachial Plexus Palsy. Journal of Pediatric Orthopaedics. 32, 378-384. http://ovidsp.ovid.com/ovidweb.cgi?T=JS&NEWS=n&CSC=Y&PAGE=fulltext&D=ovft&AN=01241398-201206000-00011

Little, K.J., Zlotolow, D.A., Soldado, F., Cornwall, R., Kozin, S.H., 2014. Median and/or Ulnar Nerve Fascicle Transfer for the Restoration of Elbow Flexion in Upper Neonatal Brachial Plexus Palsy. JBJS essential surgical techniques. 4, e8. https://www.ncbi.nlm.nih.gov/pubmed/30775115

Maganaris, C.N., Paul, J.P., 2002. Tensile properties of the in vivo human gastrocnemius tendon. Journal of Biomechanics. 35, 1639-1646. https://www.sciencedirect.com/science/article/pii/S0021929002002403

Nath, R., Paizi, M., 2007. Scapular deformity in obstetric brachial plexus palsy: a new finding. Surgical and Radiologic Anatomy. 29, 133-140. https://www.ncbi.nlm.nih.gov/pubmed/17262175

Nikolaou, S., Hu, L., Cornwall, R., 2015. Afferent Innervation, Muscle Spindles, and Contractures Following Neonatal Brachial Plexus Injury in a Mouse Model. The Journal of hand surgery. 40, 2007. http://www.ncbi.nlm.nih.gov/pubmed/26319770

Nikolaou, S., Liangjun, H., Tuttle, L.J., Weekley, H., Christopher, W., Lieber, R.L., Cornwall, R., 2014. Contribution of denervated muscle to contractures after neonatal brachial plexus injury: Not just muscle fibrosis. Muscle & Nerve. 49, 398-404. http://onlinelibrary.wiley.com/doi/10.1002/mus.23927/abstract

Nikolaou, S., Peterson, E., Kim, A., Wylie, C., Cornwall, R., 2011. Impaired Growth of Denervated Muscle Contributes to Contracture Formation Following Neonatal Brachial Plexus Injury. The Journal of Bone & Joint Surgery. 93, 461-470. http://jbjs.org/article.aspx?articleid=6228

Norlin, R., Hoe-Hansen, C., OQuist, G., Hilderbrand, C., 1994. Shoulder Region of the Rat: Anatomy and Fiber Composition of Some Suprascapular Nerve Branches. The Anatomical Record. 239, 332-342.

Nougarolis, F., Mokrane, F., Sans, N., Rousseau, H., Dedouit, F., Telmon, N., 2017. Bone age estimation based on multislice computed tomography study of the scapula. International Journal of Legal Medicine. 131, 547-558. https://search.proquest.com/docview/1867560323

Patra, S., Narayana Kurup, J.K., Acharya, A.M., Bhat, A.K., 2016. Birth brachial plexus palsy: a race against time. BMJ case reports. 2016, bcr2016215996. https://www.ncbi.nlm.nih.gov/pubmed/27402656

Pearl, M., Edgerton, B., Kon, D., Darakjian, A., 2003. Comparison of arthroscopic findings with magnetic resonance imaging and arthrography in children with glenohumeral deformities secondary to brachial plexus birth palsy. Journal of Bone and Joint Surgery. 85, 890-898. https://www.ncbi.nlm.nih.gov/pubmed/12728041

Pearl, M.L., 2009. Shoulder Problems in Children With Brachial Plexus Birth Palsy: Evaluation and Management. Journal of the American Academy of Orthopaedic Surgeons. 17, 242. http://www.jaaos.org/content/17/4/242.abstract

Pearl, M.L., Edgerton, B.W., 1998. Glenoid Deformity Secondary to Brachial Plexus Birth Palsy. The Journal of Bone & Joint Surgery. 80, 659-67. http://jbjs.org/article.aspx?articleid=24019

Pichon, F., Carlioz, H., 1979. Disinsertion of the subscapularis muscle in the treatment of obstetric paralysis of the upper limb (author's transl). Chirurgie pediatrique. 20, 135. https://www.ncbi.nlm.nih.gov/pubmed/487496

Pondaag, W., Malessy, M.J., Gert van Dijk, J., Thomeer, R.T., 2004. Natural history of obstetric brachial plexus palsy: a systematic review. Developmental Medicine and Child Neurology. 46, 138-144. http://journals.cambridge.org/abstract_S0012162204000258

Quinn, R., 2005. Comparing rat’s to human’s age: How old is my rat in people years? Nutrition. 21, 775-777. https://www.sciencedirect.com/science/article/pii/S0899900705001711

Roddy, K.A., Kelly, G.M., van Es, M.H., Murphy, P., Prendergast, P.J., 2011. Dynamic patterns of mechanical stimulation co-localise with growth and cell proliferation during morphogenesis in the avian embryonic knee joint. Journal of Biomechanics. 44, 143-149. http://www.sciencedirect.com/science/article/pii/S0021929010004860

Saul, K.R., Hu, X., Goehler, C.M., Vidt, M.E., Daly, M., Velisar, A., Murray, W.M., 2015. Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model. Computer Methods in Biomechanics and Biomedical Engineering. 18, 1445-1458. http://www.tandfonline.com/doi/abs/10.1080/10255842.2014.916698

Sever, J.W., 1920. Obstetric Paralysis-Its Cause and Treatment. Canadian Medical Association journal. 10, 141-161. https://www.ncbi.nlm.nih.gov/pubmed/20312211

Sheehan, F.T., Brochard, S., Behnam, A.J., Alter, K.E., 2014. Three-dimensional humeral morphologic alterations and atrophy associated with obstetrical brachial plexus palsy. Journal of shoulder and elbow surgery / American Shoulder and Elbow Surgeons ... [et al.]. 23, 708-719. http://www.ncbi.nlm.nih.gov/pubmed/24291045

Shefelbine, S.J., Tardieu, C., Carter, D.R., 2002. Development of the femoral bicondylar angle in hominid bipedalism. Bone. 30, 765-770. https://www.sciencedirect.com/science/article/pii/S8756328202007007

Shefelbine, S., Carter, D., 2004. Mechanobiological Predictions of Femoral Anteversion in Cerebral Palsy. Annals of Biomedical Engineering. 32, 297-305. http://www.ncbi.nlm.nih.gov/pubmed/15008378

Sibinski, M., Woźniakowski, B., Drobniewski, M., Synder, M., 2010. Secondary gleno-humeral joint dysplasia in children with persistent obstetric brachial plexus palsy. International Orthopaedics. 34, 863-867. http://www.ncbi.nlm.nih.gov/pubmed/20174796

Soldado, F., Bertelli, J.A., Ghizoni, M.F., 2016. High Median Nerve Injury: Motor and Sensory Nerve Transfers to Restore Function. Hand clinics. 32, 209. https://www.ncbi.nlm.nih.gov/pubmed/27094892

Stein, J., Laor, T., Carr, P., Zbojniewicz, A., Cornwall, R., 2017. The Effect of Scapular Position on Magnetic Resonance Imaging Measurements of Glenohumeral Dysplasia Caused by Neonatal Brachial Plexus Palsy. Journal of Hand Surgery. 42, 1030.e11. https://www.sciencedirect.com/science/article/pii/S0363502317311632

Swan, M.A., Sato, E., Galatz, L.M., Thomopoulos, S., Ward, S.R., 2014. The effect of age on rat rotator cuff muscle architecture. Journal of shoulder and elbow surgery / American Shoulder and Elbow Surgeons ... [et al.]. 23, 1786-1791. http://www.ncbi.nlm.nih.gov/pubmed/24878035

Thelen, D.G., Anderson, F.C., 2006. Using computed muscle control to generate forward dynamic simulations of human walking from experimental data. Journal of Biomechanics. 39, 1107-1115. http://www.sciencedirect.com/science/article/pii/S0021929005000990

Vaal, J., Van Soest, A. J., Hopkins, B., 2000. Spontaneous kicking behavior in infants: Age-related effects of unilateral weighting. Developmental Psychobiology. 36, 111-122. http://onlinelibrary.wiley.com/doi/10.1002/(SICI)1098-2302(200003)36:23.0.CO;2-H/abstract

Vaca-González, J., Moncayo-Donoso, M., Guevara, J., Hata, Y., Shefelbine, S., Garzón-Alvarado, D., 2018. Mechanobiological modeling of endochondral ossification: an experimental and computational analysis. Biomechanics and Modeling in Mechanobiology. 17, 853-875. https://www.ncbi.nlm.nih.gov/pubmed/29322335

Van Der Meulen, M C H, Beaupré, G.S., Carter, D.R., 1993. Mechanobiologic influences in long bone cross-sectional growth. Bone. 14, 635-642. http://www.sciencedirect.com/science/article/pii/875632829390085O

van Gelein Vitringa, Valerie M, van Royen, B.J., van der Sluijs, Johannes A, 2013. Scapular deformity in obstetric brachial plexus palsy and the Hueter-Volkmann law; a retrospective study. BMC musculoskeletal disorders. 14, 107. http://www.ncbi.nlm.nih.gov/pubmed/23522350

Waters, 1999. Comparison of Natural History, the Outcome of Microsurgical Repair, and the Outcome of Operative Reconstruction in Brachial Plexus Birth Palsy. 81, 649-659.

Waters, P., 2005. Update on management of pediatric brachial plexus palsy. Journal of Pediatric Orthopaedics B. 14, 233-244. http://ovidsp.ovid.com/ovidweb.cgi?T=JS&NEWS=n&CSC=Y&PAGE=fulltext&D=ovft&AN=01202412-200507000-00001

Waters, P.M., Monica, J.T., Earp, B.E., Zurakowski, D., Bae, D.S., 2009. Correlation of Radiographic Muscle Cross-Sectional Area with Glenohumeral Deformity in Children with Brachial Plexus Birth Palsy. The Journal of Bone & Joint Surgery. 91, 2367-2375. http://jbjs.org/article.aspx?articleid=28958

Waters, P.M., Smith, G.R., Jaramillo, D., 1998. Glenohumeral Deformity Secondary to Brachial Plexus Birth Palsy. The Journal of Bone & Joint Surgery. 80, 668-77. http://jbjs.org/article.aspx?articleid=24020

Zafeiriou, D., Psychogiou, K., 2008. Obstetrical Brachial Plexus Palsy. Pediatric Neurology. 38, 235-242. https://www.clinicalkey.es/playcontent/1-s2.0-S0887899407004894

Zajac, F.E., 1989. Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. Critical reviews in biomedical engineering. 17, 359. https://www.ncbi.nlm.nih.gov/pubmed/2676342

APPENDICES

Appendix A: Muscle mass (g)

This appendix contains data from Chapter 3 about the muscle mass measurements on both affected and unaffected limbs of the rats due to preganglionic and postganglionic BPBI

Preganglionic

Right limb unaffected		Muscles										
Pheno type	Rat name	Pectoralis major	Anterior Deltoid	Spinal deltoid	Bicep long	Bicep short	Subcscap	Supraspin	Infraspin	Teres minor	Teres major	Ticeps long
Yes	23D	0.44	0.16	0.13	0.10	0.04	0.35	0.23	0.27	0.01	0.26	0.89
Yes	24E	0.26	0.12	0.15	0.09	0.05	0.36	0.23	0.18	0.01	0.09	0.87
Yes	25C	0.33	0.12	0.16	0.13	0.02	0.35	0.35	0.23	0.04	0.30	0.92
Yes	24A	0.26	0.10	0.11	0.10	0.01	0.27	0.17	0.17	0.03	0.21	0.62
Yes	23A	0.46	0.19	0.15	0.11	0.07	0.30	0.31	0.19	0.12	0.11	0.85
Yes	24B	0.18	0.07	0.11	0.09	0.01	0.23	0.16	0.15	0.03	0.13	0.62
Yes	25A	0.22	0.08	0.10	0.07	0.01	0.17	0.14	0.13	0.02	0.15	0.59
No	24C	0.32	0.16	0.17	0.10	0.02	0.34	0.25	0.19	0.04	0.15	0.78
No	25D	0.19	0.11	0.09	0.10	0.02	0.28	0.18	0.19	0.18	0.08	0.70
No	26B	0.40	0.16	0.21	0.14	0.02	0.46	0.26	0.22	0.11	0.38	0.97
No	25B	0.46	0.13	0.19	0.14	0.02	0.50	0.33	0.16	0.06	0.15	1.03
No	26D	0.25	0.11	0.08	0.09	0.02	0.25	0.26	0.18	0.02	0.23	0.68
No	24D	0.36	0.10	0.14	0.11	0.03	0.33	0.25	0.22	0.02	0.32	0.75
	SD	**0.10**	**0.04**	**0.04**	**0.02**	**0.02**	**0.09**	**0.06**	**0.04**	**0.05**	**0.09**	**0.14**
	Mean	**0.32**	**0.12**	**0.14**	**0.11**	**0.03**	**0.32**	**0.24**	**0.19**	**0.06**	**0.20**	**0.79**

Left limb affected		Muscles										
Pheno type	Rat name	Pectoralis major	Anterior Deltoid	Spinal deltoid	Bicep long	Bicep short	Subscap	Supraspin	Infraspin	Teres minor	Teres major	Ticeps long
Yes	23D	0.32	0.04	0.14	0.04	0.03	0.21	0.08	0.05		0.23	0.88
Yes	24E	0.16	0.05	0.02	0.02	0.01	0.05	0.01	0.02	0.01	0.06	0.57
Yes	25C	0.37	0.03	0.03	0.03	0.01	0.13	0.05	0.06	0.01	0.15	0.78
Yes	24A	0.37	0.05	0.07	0.02	0.01	0.09	0.03	0.03	0.03	0.05	0.40
Yes	23A	0.21	0.02	0.01	0.03	0.01	0.02	0.03	0.02	0.04	0.05	0.71
Yes	24B	0.18	0.02	0.03	0.02	0.01	0.11	0.02	0.05	0.06	0.04	0.52
Yes	25A	0.11	0.02	0.01	0.01	0.01	0.02	0.03	0.04	0.01	0.02	0.32
No	24C	0.44	0.09	0.13	0.07	0.01	0.48	0.26	0.12	0.06	0.15	0.94
No	25D	0.24	0.10	0.10	0.07	0.04	0.25	0.18	0.18	0.14	0.14	0.64
No	26B	0.38	0.21	0.18	0.16	0.03	0.56	0.30	0.28	0.07	0.37	1.07
No	25B	0.40	0.13	0.18	0.12	0.04	0.37	0.34	0.23	0.02	0.36	0.98
No	26D	0.18	0.11	0.06	0.09	0.03	0.24	0.22	0.18	0.03	0.22	0.66
No	24D	0.52	0.02	0.02	0.01	0.01	0.06	0.02	0.02	0.01	0.05	0.43
	SD	**0.13**	**0.06**	**0.06**	**0.05**	**0.01**	**0.18**	**0.12**	**0.09**	**0.04**	**0.12**	**0.24**
	Mean	**0.30**	**0.07**	**0.08**	**0.05**	**0.02**	**0.20**	**0.12**	**0.10**	**0.04**	**0.15**	**0.68**

Postganglionic

Right limb affected	Muscles										
Rat name	Pectoralis major	Anterior Deltoid	Spinal deltoid	Bicep long	Bicep short	Subscap	Supraspin	Infraspin	Teres minor	Teres major	Ticeps long
98A	0.28	0.02		0.01	0.01	0.14	0.11	0.24	0.03	0.05	0.64
97B	0.17	0.10	0.09	0.02	0.01	0.14	0.21	0.16	0.01	0.18	0.49
00D	0.28	0.17	0.13	0.10	0.05	0.32	0.28	0.30	0.02	0.25	0.74
00E	0.12	0.05	0.05	0.01	0.01	NA	0.15	0.13	NA	0.11	0.14
00C	0.31	0.13	0.15	0.04	NA	0.29	0.22	0.24	0.01	0.32	0.71
97A	0.20	0.14	0.11	0.10	0.01	0.25	0.20	0.18	0.02	0.16	0.57
98B	0.17	0.07	0.03	0.06	0.04	0.22	0.11	0.17	0.01	0.10	0.09
98C	0.21	0.04	0.01	0.01	0.01	0.14	0.15	0.17	0.01	0.03	0.08
99A	0.14	0.08	0.05	0.01	0.01	0.13	0.11	0.10	0.02	0.08	0.09
99C	0.16	0.02	0.01	0.01	0.01	0.10	0.15	0.17	0.01	0.01	0.09
99F	0.42	0.06	0.07	0.10	0.02	0.21	0.29	0.32	0.01	0.14	0.81
SD	**0.09**	**0.05**	**0.05**	**0.04**	**0.01**	**0.07**	**0.06**	**0.07**	**0.01**	**0.09**	**0.30**
Mean	**0.22**	**0.08**	**0.07**	**0.04**	**0.02**	**0.19**	**0.18**	**0.20**	**0.02**	**0.13**	**0.40**

Left limb unaffected	Muscles										
Rat name	Pectoralis major	Anterior Deltoid	Spinal deltoid	Bicep long	Bicep short	Subscap	Supraspin	Infraspin	Teres minor	Teres major	Ticeps long
98A	0.36	0.14	0.13	0.11	0.02	0.28	0.23	0.17	0.02	0.25	0.73
97B	0.23	0.10	0.09	0.07	0.01	0.19	0.19	0.19	0.01	0.18	0.55
00D	0.34	0.20	0.14	0.10	0.02	0.24	0.19	0.25	0.01	0.21	0.79
00E	0.19	0.12	0.10	0.06	0.01	0.22	0.15	0.21	0.01	0.15	0.54
00C	0.45	0.17	0.16	0.10	0.02	0.32	0.27	0.28	0.01	0.29	0.77
97A	0.30	0.11	0.13	0.09	0.03	0.29	0.19	0.21	0.01	0.17	0.62
98B	0.20	0.11	0.08	0.08	0.02	0.27	0.18	0.21	0.02	0.16	0.59
98C	0.22	0.12	0.09	0.08	0.02	0.24	0.18	0.19	0.01	0.18	0.58
99A	0.23	0.09	0.07	0.07	0.02	0.19	0.16	0.18	0.01	0.14	0.54
99C	0.27	0.11	0.12	0.09	0.01	0.19	0.20	0.19	0.01	0.20	0.59
99F	0.48	0.18	0.20	0.12	0.02	0.32	0.31	0.35	0.01	0.39	0.85
SD	**0.10**	**0.04**	**0.04**	**0.02**	**0.01**	**0.05**	**0.05**	**0.05**	**0.00**	**0.07**	**0.11**
Mean	**0.30**	**0.13**	**0.12**	**0.09**	**0.02**	**0.25**	**0.20**	**0.22**	**0.01**	**0.21**	**0.65**

Appendix B: Sarcomere length (um)

This appendix contains the data from chapter 3 about the muscle sarcomere length measurements on both the limbs of the rats affected by preganglionic and postganglionic BPBI

Preganglionic

Right limb unaffected		Muscles										
Phenotype	Rat name	Pectoralis major	Anterior Deltoid	Spinal deltoid	Bicep long	Bicep short	Subscap	Supraspin	Infraspin	Teres minor	Teres major	Ticeps long
Yes	23D	1.91	2.34	1.98	2.37	2.08	2.40	2.56	2.10	2.60	1.86	2.10
Yes	24E	2.15	2.42	2.05	2.13	1.92	2.31	2.44	2.34	2.32	1.73	2.06
Yes	25C	1.92	2.24	2.19	2.29	1.98	2.05	2.31	2.33	2.40	1.82	2.12
Yes	24A	1.97	2.47	2.05	2.14	2.13	2.17	2.55	2.57	2.59	1.73	2.14
Yes	23A	2.35	2.39	1.95	2.26	2.10	2.49	2.52	2.35	1.73	1.68	1.98
Yes	24B	2.00	2.46	1.88	2.41	1.94	2.53	2.50	2.18	2.36	1.81	1.84
Yes	25A	2.19	2.45	2.13	2.25	1.98	1.95	2.80	2.71	2.86	1.77	2.19
No	24C	2.11	2.42	2.08	2.49	2.07	2.30	2.69	2.43	2.49	1.77	1.90
No	25D	2.34	2.46	1.92	2.28	1.99	2.11	2.39	1.81	2.39	1.73	1.86
No	26B	1.97	2.24	2.04	1.97	1.81	2.08	2.47	2.44	2.35	1.76	2.04
No	25B	1.89	2.24	1.82	2.35	2.07	2.42	2.35	2.23	2.23	1.74	1.88
No	26D	2.73	2.54	2.02	2.11	1.90	2.37	2.55	2.36	2.20	1.71	2.00
No	24D	2.27	2.39	2.07	2.28	2.06	2.21	2.71	2.67	2.52	1.94	2.05
	SD	**0.24**	**0.10**	**0.10**	**0.14**	**0.09**	**0.18**	**0.14**	**0.24**	**0.25**	**0.07**	**0.11**
	Mean	**2.13**	**2.39**	**2.01**	**2.25**	**2.00**	**2.26**	**2.53**	**2.34**	**2.38**	**1.77**	**2.01**

Left limb affected		Muscles										
Phenotype	Rat name	Pectoralis major	Anterior Deltoid	Spinal deltoid	Bicep long	Bicep short	Subscap	Supraspin	Infraspin	Teres minor	Teres major	Ticeps long
Yes	23D	2.24	2.55	2.24	2.49	2.15	1.89	2.74	2.56	No data	2.12	1.87
Yes	24E	2.25	2.23	2.27	2.64	2.40	2.14	2.41	2.30	2.40	2.08	1.87
Yes	25C	2.02	2.43	2.09	2.69	2.37	2.28	2.31	2.29	2.36	1.97	2.13
Yes	24A	2.12	2.58	1.86	3.03	2.72	2.57	2.69	2.53	2.40	2.18	1.85
Yes	23A	3.23	2.94	2.32	2.64	2.36	2.78	2.98	2.52	2.27	2.17	2.20
Yes	24B	2.09	2.52	2.40	2.60	2.32	1.78	2.67	2.11	1.94	2.11	2.05
Yes	25A	2.70	2.67	2.59	2.51	2.47	3.54	2.65	2.34	2.46	2.12	2.68
No	24C	1.94	2.48	2.07	2.42	2.06	2.29	2.64	2.52	2.43	1.61	2.04
No	25D	2.21	2.47	1.92	2.45	2.29	2.25	2.57	2.46	1.89	1.81	1.86
No	26B	2.01	2.37	1.99	2.26	2.11	2.24	2.17	2.16	2.32	2.05	1.96
No	25B	2.00	2.40	2.03	2.33	1.92	2.23	2.46	2.37	2.33	1.76	1.85
No	26D	2.14	2.44	1.93	2.37	1.97	2.29	2.52	2.38	2.28	1.77	1.91
No	24D	2.11	2.45	2.15	2.47	2.55	2.05	2.23	2.36	2.33	2.12	2.25
	SD	**0.35**	**0.17**	**0.21**	**0.20**	**0.23**	**0.44**	**0.22**	**0.14**	**0.17**	**0.19**	**0.23**
	Mean	**2.23**	**2.50**	**2.14**	**2.53**	**2.28**	**2.33**	**2.54**	**2.37**	**2.28**	**1.99**	**2.04**

Postganglionic

Right limb affected	Muscles									
Rat name	Pectoralis major	Anterior Deltoid	Spinal deltoid	Bicep long	Bicep short	Subscap	Supraspin	Infraspin	Teres minor	Teres major	Ticeps long
98A	2.38	2.30	#DIV/0!	2.48	2.24	2.43	2.53	2.44	2.38	2.02	1.84
97B	2.29	2.41	2.01	2.42	2.19	2.13	2.52	2.41	2.14	1.99	2.10
00D	2.13	2.14	1.82	2.19	2.11	2.51	2.41	2.23	2.14	2.01	2.16
00E	2.45	2.42	1.99	2.17	2.27	2.34	2.60	2.34	2.20	1.93	2.28
00C	2.14	2.51	1.95	NA	2.23	2.27	2.38	2.30	2.26	1.89	2.02
97A	2.33	2.18	1.87	2.03	1.78	2.21	2.47	2.30	2.30	1.67	2.11
98B	2.35	2.39	2.03	2.01	2.04	2.06	2.17	1.94	2.20	1.99	2.24
98C	2.30	2.35	1.84	2.35	2.21	2.44	2.53	2.20	2.44	2.06	1.85
99A	2.25	2.39	1.99	2.44	2.35	2.20	2.13	2.21	2.16	2.06	2.04
99C	2.87	2.23	2.04	2.10	2.20	2.02	2.09	2.37	2.61	2.23	2.26
99F	2.52	2.79	2.22	2.37	2.19	2.19	2.56	2.54	2.38	1.86	1.83
SD	**0.20**	**0.18**	**0.11**	**0.18**	**0.15**	**0.16**	**0.18**	**0.16**	**0.14**	**0.14**	**0.17**
Mean	**2.36**	**2.37**	**1.97**	**2.25**	**2.16**	**2.25**	**2.40**	**2.30**	**2.90**	**1.97**	**2.01**

Left limb unaffected	Muscles									
Rat name	Pectoralis major	Anterior Deltoid	Spinal deltoid	Bicep long	Bicep short	Subscap	Supraspin	Infraspin	Teres minor	Teres major	Ticeps long
98A	2.35	2.50	2.02	2.14	1.82	2.07	2.56	2.58	2.46	1.79	2.17
97B	2.52	2.50	1.96	2.27	1.97	2.44	2.28	2.27	2.08	1.94	2.11
00D	2.01	2.44	2.10	2.13	2.05	2.06	2.32	2.23	2.37	1.85	2.21
00E	2.09	2.38	1.92	2.38	2.19	2.25	2.50	2.38	2.21	1.88	2.07
00C	2.22	2.38	1.90	2.20	1.99	2.25	2.34	2.23	2.08	1.86	1.92
97A	2.68	2.73	1.78	2.50	2.32	2.06	2.37	2.28	2.16	1.75	1.86
98B	2.29	2.39	1.85	2.16	1.82	2.18	2.47	2.26	2.28	1.69	2.42
98C	2.28	2.53	2.02	2.58	2.30	2.09	2.51	2.38	2.24	1.63	1.67
99A	1.94	2.40	1.97	2.17	1.95	2.07	2.47	2.37	2.48	1.87	2.03
99C	2.66	3.01	2.06	2.19	2.08	2.13	2.51	2.48	2.64	1.94	2.21
99F	2.38	2.49	2.13	2.47	2.17	2.04	2.66	2.41	2.48	1.73	1.83
SD	**0.24**	**0.19**	**0.11**	**0.17**	**0.17**	**0.12**	**0.11**	**0.11**	**0.17**	**0.10**	**0.21**
Mean	**2.31**	**2.52**	**1.97**	**2.30**	**2.06**	**2.15**	**2.45**	**2.35**	**2.31**	**1.81**	**2.05**

Appendix C: Optimal length (mm)

This appendix contains the data from chapter 3 about optimal muscle length measurements on both the limbs of the rats affected by preganglionic and postganglionic BPBI. NA refers to missing data.

Preganglionic

Right limb unaffected		Muscles										
Phenotype	Rat name	Pectoralis major	Anterior Deltoid	Spinal deltoid	Bicep long	Bicep short	Subscap	Supraspin	Infraspin	Teres minor	Teres major	Ticeps long
Yes	23D	28.82	12.67	21.58	14.90	20.99	18.87	19.70	25.62	9.48	21.46	26.25
Yes	24E	17.91	14.76	21.67	20.03	27.55	21.62	25.09	22.56	9.10	19.67	25.02
Yes	25C	22.91	13.16	21.14	16.10	18.37	21.11	25.88	23.20	22.51	26.76	25.13
Yes	24A	20.69	9.17	16.07	15.75	17.20	17.67	20.19	19.08	14.84	23.43	22.64
Yes	23A	24.47	14.96	21.41	18.04	21.15	18.98	22.18	21.78	23.07	29.77	29.63
Yes	24B	26.21	10.88	33.55	15.87	17.58	17.53	21.42	20.36	17.16	24.76	26.63
Yes	25A	15.62	9.40	14.46	13.63	16.50	19.33	13.89	13.75	10.69	20.46	20.42
No	24C	22.80	14.45	20.74	18.58	21.57	19.73	22.66	23.39	16.20	23.63	30.61
No	25D	18.27	12.01	18.14	15.21	18.30	20.25	20.30	24.91	15.16	20.25	25.53
No	26B	21.40	14.34	22.36	18.81	22.03	21.03	23.27	21.23	21.09	27.94	25.70
No	25B	22.12	12.27	22.61	15.74	18.24	17.20	24.29	24.41	20.90	21.76	30.33
No	26D	18.93	9.93	16.56	15.18	18.12	15.67	21.15	19.47	8.79	20.78	24.22
No	24D	24.02	13.86	22.01	18.07	21.82	20.95	22.34	22.06	10.92	27.34	27.72
	SD	**3.63**	**2.05**	**4.64**	**1.89**	**2.99**	**1.80**	**3.02**	**3.11**	**5.09**	**3.35**	**2.94**
	Mean	**21.85**	**12.45**	**20.94**	**16.60**	**19.95**	**19.22**	**21.72**	**21.67**	**15.87**	**23.69**	**26.14**

Left limb affected		Muscles										
Phenotype	Rat name	Pectoralis major	Anterior Deltoid	Spinal deltoid	Bicep long	Bicep short	Subscap	Supraspin	Infraspin	Teres minor	Teres major	Ticeps long
Yes	23D	20.42	10.74	21.74	12.54	20.00	16.77	16.57	15.31	NA	17.11	29.48
Yes	24E	22.46	16.99	15.89	16.33	17.89	15.34	15.52	15.07	8.56	18.03	25.93
Yes	25C	21.85	9.86	15.48	13.88	14.76	19.78	17.74	18.32	13.90	23.40	25.85
Yes	24A	20.47	10.11	17.09	11.03	11.20	15.35	14.97	16.22	15.14	17.55	25.09
Yes	23A	16.84	8.04	11.21	10.24	13.16	11.50	12.85	17.43	16.60	13.99	22.22
Yes	24B	27.69	10.55	19.97	10.16	16.24	22.77	12.04	18.43	16.28	11.03	23.85
Yes	25A	14.41	9.69	11.25	13.11	16.64	6.96	9.73	18.45	7.29	14.21	12.81
No	24C	22.46	14.31	22.21	18.06	20.18	18.79	23.33	21.76	20.52	26.03	27.15
No	25D	18.51	11.38	20.75	12.48	17.45	18.80	19.50	18.85	17.72	21.97	27.04
No	26B	20.63	12.45	25.28	18.88	20.21	20.49	24.46	23.65	19.62	24.88	27.57
No	25B	19.64	12.04	21.35	16.23	20.20	21.60	21.23	23.42	13.72	26.58	28.46
No	26D	18.10	10.02	14.47	14.38	16.34	16.61	19.00	20.88	15.53	23.96	25.86
No	24D	35.67	11.38	15.33	19.21	19.31	26.86	18.81	19.17	14.83	24.53	19.76
	SD	**5.32**	**2.28**	**4.37**	**3.15**	**2.88**	**5.05**	**4.35**	**2.78**	**3.73**	**5.19**	**4.41**
	Mean	**21.47**	**11.34**	**17.84**	**14.34**	**17.19**	**17.81**	**17.36**	**18.99**	**14.97**	**20.25**	**24.69**

Postganglionic

Right limb affected	Muscles										
Rat name	Pectoralis major	Anterior Deltoid	Spinal deltoid	Bicep long	Bicep short	Subscap	Supraspin	Infraspin	Teres minor	Teres major	Ticeps long
98A	18.93	10.62	NA	19.78	18.61	18.39	18.05	22.84	17.36	19.72	23.54
97B	17.72	12.06	17.27	16.68	16.58	18.30	21.73	20.63	10.11	22.54	22.75
00D	19.20	12.28	21.90	14.94	19.12	16.17	22.10	22.31	9.66	21.60	23.83
00E	16.10	10.70	14.75	13.94	12.49	NA	17.88	18.17	NA	19.86	18.32
00C	21.10	12.36	20.46	NA	NA	20.18	22.39	23.90	10.39	27.62	25.24
97A	18.17	13.65	19.62	20.44	21.12	18.34	20.97	22.19	8.01	25.90	24.42
98B	16.60	9.15	11.89	15.26	16.50	17.47	19.42	22.65	5.22	16.69	21.19
98C	18.75	9.53	13.02	14.91	17.89	16.32	18.13	21.78	3.33	17.35	25.23
99A	15.37	9.14	13.37	15.46	13.05	14.00	19.13	20.30	15.07	15.94	19.97
99C	14.60	10.72	15.74	15.54	19.21	17.93	22.66	19.52	5.63	17.92	27.12
99F	21.41	7.52	16.63	19.19	20.05	20.56	23.74	20.97	9.07	25.42	27.54
SD	**2.20**	**1.79**	**3.36**	**2.32**	**2.85**	**1.93**	**2.11**	**1.65**	**4.00**	**4.00**	**2.85**
Mean	**18.00**	**10.70**	**16.46**	**16.61**	**17.46**	**17.76**	**20.56**	**21.38**	**9.38**	**20.96**	**23.56**

Left limb unaffected	Muscles										
Rat name	Pectoralis major	Anterior Deltoid	Spinal deltoid	Bicep long	Bicep short	Subscap	Supraspin	Infraspin	Teres minor	Teres major	Ticeps long
98A	17.96	12.44	18.23	17.91	20.01	21.33	21.17	20.83	12.87	24.90	23.42
97B	20.17	11.88	18.47	15.85	17.62	17.53	22.97	21.17	9.16	20.23	22.64
00D	20.85	11.52	19.16	14.51	17.72	18.88	21.32	21.23	7.60	24.40	23.93
00E	18.63	11.32	17.79	14.00	15.14	17.28	21.01	21.22	8.63	21.85	24.62
00C	23.60	11.83	20.85	15.46	18.21	21.57	26.98	23.84	7.67	22.12	26.67
97A	19.41	10.68	19.40	18.30	19.82	19.90	21.00	21.75	8.81	24.01	24.59
98B	17.31	11.14	21.43	15.21	17.83	17.70	18.45	21.98	10.64	22.74	18.55
98C	19.38	11.32	18.80	15.18	16.57	19.95	20.13	22.66	4.85	26.57	26.05
99A	22.45	8.78	20.23	15.31	19.57	19.60	19.61	20.78	9.49	24.50	22.94
99C	21.87	8.14	17.48	13.45	17.90	19.56	20.44	18.91	8.67	21.96	22.41
99F	22.56	12.45	19.90	17.12	19.64	19.64	22.24	24.65	11.79	29.29	27.22
SD	**2.05**	**1.39**	**1.27**	**1.62**	**1.58**	**1.31**	**2.21**	**1.56**	**2.06**	**2.52**	**2.41**
Mean	**20.38**	**11.04**	**19.25**	**15.66**	**18.18**	**19.35**	**21.39**	**21.72**	**9.10**	**23.87**	**23.91**

Appendix D: Range of motion (degrees)

This appendix contains the data from chapter 3 about passive external rotation range of motion measurements made on both the limbs of the rats affected by preganglionic and postganglionic BPBI at 4 weeks and 8 weeks postnatal timepoints.

Preganglionic								
4 weeks					8 weeks			
		Unafected	Affected				Unafected	Affected
Rat id	Phenotype	Right limb	Left limb		Rat id	Phenotype	Right limb	Left limb
23A	Yes	89.67	50.67		23A	Yes	106.67	62.33
23E	Yes	100.67	115.00		23D	Yes	57.67	115.33
23D	Yes	113.33	95.00		24A	Yes	81.67	128.33
23F	Yes	109.00	96.67		24B	Yes	116.67	90.00
24I	Yes	119.67	78.67		24D	Yes	67.33	20.00
24H	Yes	133.33	124.67		24E	Yes	76.67	56.67
24K	Yes	89.67	92.00		25A	Yes	81.67	55.00
24D	Yes	177.00	46.33		25C	Yes	96.67	90.00
24E	Yes	143.00	114.67		26C	Yes	88.33	118.33
24F	Yes	209.67	65.67		23E	Yes	50.00	95.00
24A	Yes	166.00	177.33		23F	Yes	50.00	153.33
24B	Yes	216.00	100.00		24F	Yes	66.67	10.00
25G	Yes	125.33	88.33		24H	Yes	41.67	50.00
25C	Yes	144.33	120.33		24I	Yes	43.33	76.67
25A	Yes	179.67	71.00		24K	Yes	71.67	170.00
26F	Yes	68.00	62.67		25G	Yes	55.00	78.33
26E	No	154.67	207.33		26F	Yes	46.67	50.00
26A	No	147.00	164.33		24C	No	88.33	70.00
26D	No	149.33	174.00		25B	No	40.00	73.33
24C	No	166.00	218.33		25D	No	78.33	143.33
24J	No	169.00	156.67		26A	No	103.33	75.00
25D	No	121.67	121.33		26D	No	106.67	113.33
25E	No	133.67	179.67		24J	No	70.00	71.67
25F	No	149.00	88.67		25E	No	100.00	93.33
25B	No	72.00	151.00		25F	No	53.33	96.67
	SD	**37.50**	**47.43**		26E	No	70.00	96.67
	Mean	**137.86**	**118.41**			**SD**	**22.17**	**37.05**
						Mean	**73.39**	**86.64**

	Postganglionic			
	4 weeks		8 weeks	
	Unaffected	Affected	Unaffected	Affected
Rat id	Left limb	Right limb	Left limb	Right limb
00C	163.33	160.00	36.67	26.67
00D	136.67	105.00	33.33	26.67
00E	173.33	110.00	121.67	20.00
00H	166.67	110.00	96.67	30.00
00I	165.00	103.33	85.00	50.00
97A	185.00	96.67	46.67	20.00
97B	193.33	83.33	76.67	40.00
97C	173.33	90.00	66.67	70.00
97F	128.33	120.00	35.00	56.67
97G	158.33	90.00	51.67	70.00
97H	160.00	81.67	60.00	18.33
98A	116.67	43.33	86.67	21.67
98B	136.67	91.67	76.67	missing data
98C	101.67	105.00	60.00	31.67
98D	128.33	91.67	66.67	20.00
98E	175.00	91.67	70.00	70.00
98F	166.67	90.00	40.00	43.33
99A	180.00	76.67	63.33	23.33
99C	103.33	75.00	106.67	66.67
99D	120.00	20.00	53.33	110.00
99E	140.00	30.00	80.00	48.33
99H	146.67	73.33	66.67	53.33
SD	**25.90**	**29.13**	**22.63**	**23.37**
Mean	**150.83**	**88.10**	**67.27**	**43.65**

Appendix E: Bone measurements

This appendix contains the data from chapter 4 about micro CT scan measurements of humeral head and glenoid features on both the limbs of the rats affected by preganglionic and postganglionic BPBI.

Preganlionic	Right limb unaffected						Left limb affected					
Rat name	HRC (mm)	HH thcikness (mm)	HH width (mm)	GRC (mm)	GIA (deg)	GVA (deg)	HRC (mm)	HH thcikness (mm)	HH width (mm)	GRC (mm)	GIA (deg)	GVA (deg)
24I	2.75	2.19	5.00	2.93	-23.03	3.98	2.41	2.00	4.53	2.05	-38.44	-4.01
25F	2.50	2.11	4.65	2.59	-29.24	1.44	2.49	2.06	4.82	2.83	-18.02	1.21
26F	2.82	2.49	5.46	2.78	-41.63	7.33	2.32	2.04	4.57	2.25	-36.64	5.91
23E	2.94	2.11	5.26	3.02	-31.04	3.92	2.64	2.10	4.82	2.91	-17.16	-0.99
26E	2.91	2.59	5.39	2.83	-32.05	0.72	2.90	2.32	5.34	2.63	-29.61	3.80
23F	2.52	2.12	4.71	2.48	-19.50	2.86	2.54	2.16	4.47	2.75	-24.82	3.70
24K	2.72	2.44	4.75	2.91	-28.06	3.08	2.27	2.30	4.43	3.34	-38.50	-1.09
24H	2.72	2.76	4.97	3.01	-33.13	0.08	2.39	2.32	4.31	2.91	-37.37	-0.68
24J	2.67	2.69	4.94	2.96	-25.90	2.97	2.76	2.69	4.97	2.65	-27.94	2.41
24F	2.62	2.64	4.61	2.47	-29.90	1.36	1.92	1.84	3.48	1.69	-31.85	-2.12
23A	2.88	2.75	5.56	2.82	-17.73	2.23	2.43	2.20	4.93	1.77	-29.85	5.63
25B	2.62	2.68	5.12	2.67	-24.88	3.53	2.69	2.70	5.20	2.89	-16.43	-0.69
SD	**0.13**	**0.26**	**0.31**	**0.18**	**6.18**	**1.82**	**0.24**	**0.26**	**0.46**	**0.48**	**7.92**	**3.04**
Mean	**2.72**	**2.46**	**5.03**	**2.78**	**-28.00**	**2.79**	**2.47**	**2.22**	**4.65**	**2.55**	**-28.88**	**1.09**

Postganglionic	Right limb unaffected						Left limb affected					
Rat name	HRC (mm)	HH thcikness (mm)	HH width (mm)	GRC (mm)	GIA (mm)	GVA (mm)	HRC (mm)	HH thcikness (mm)	HH width (mm)	GRC (mm)	GIA (mm)	GVA (mm)
L3_C1_A1	2.47	2.10	4.73	1.42	-38.06	11.87	2.34	2.13	4.75	1.64	-69.40	-20.45
L3_C1_A2	2.06	2.05	4.18	1.56	-42.39	6.67	2.54	2.29	4.74	2.03	-69.55	-1.76
L3_C2_A1	2.57	2.18	4.97	1.50	-44.53	9.23	2.45	1.87	4.35	2.45	-40.66	9.09
L3_C5_A3	2.59	2.64	4.93	1.29	-35.14	8.00	2.70	2.41	4.61	3.76	-39.61	4.33
L3_C4_A3	2.39	2.26	4.66	1.27	-39.29	12.25	2.15	1.51	3.80	6.73	-63.02	3.82
L1_C1_A1	2.44	2.17	4.66	1.30	-38.00	4.90	2.34	2.33	4.56	1.19	-36.52	10.60
L1_C2_A1	2.65	2.38	4.80	1.18	-33.31	-0.28	2.46	2.26	4.71	1.19	-49.67	-6.97
L1_C2_A2	2.49	2.47	4.92	1.31	-36.65	9.18	2.11	2.12	4.73	3.79	-80.41	3.21
SD	**0.16**	**0.20**	**0.24**	**0.12**	**3.43**	**3.79**	**0.18**	**0.29**	**0.30**	**1.75**	**15.39**	**9.40**
Mean	**2.45**	**2.28**	**4.73**	**1.35**	**-38.42**	**7.72**	**2.38**	**2.11**	**4.53**	**2.84**	**-56.10**	**0.23**

Appendix F: Python script for FEA modeling

This appendix contains a Python script used to perform finite element simulations to predict bone morphological changes due to muscle length scaling and shortening scenarios investigated in Chapter 2.

FEA modeling script:

For muscle scaling simualtions, following script was developed in Python and implemented in Abaqus.

```python
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from optimization import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *

count = 1
mech_hum = 35
mech_scap = 35
bio_hum = 0.05
```

```
bio_scap = 750

for j in range(1,15):
    #if (j-1)%3 == 0:
    #   zmove = 0.1
    #else:
    #   zmove = 0.0
    mdb.openAuxMdb(pathName=
        'C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_64/M1.cae')
    mdb.copyAuxMdbModel(fromName='Model-%d' %count, toName='Model-%d' %(count+1)) # names of the new model can be changed
    mdb.copyAuxMdbModel(fromName='Model-%d' %count, toName='Model-%d' %(count+1))
    mdb.closeAuxMdb()
    mdb.models['Model-%d' %count].Material(name='Material-1')
    mdb.models['Model-%d' %count].materials['Material-1'].Density(table=((0.007, ), ))
    mdb.models['Model-%d' %count].materials['Material-1'].Elastic(table=((1.1, 0.45), ))
    mdb.models['Model-%d' %count].materials['Material-1'].Expansion(table=((8e-06, ), ))
    mdb.models['Model-%d' %count].HomogeneousSolidSection(material='Material-1', name=
        'Section-1', thickness=None)
    mdb.models['Model-%d' %count].parts['Humerus_%d' %(j-1)].SectionAssignment(offset=0.0,
        offsetField='', offsetType=MIDDLE_SURFACE, region=
        mdb.models['Model-%d' %count].parts['Humerus_%d' %(j-1)].sets['TOTAL'], sectionName=
        'Section-1', thicknessAssignment=FROM_SECTION)
    mdb.models['Model-%d' %count].parts['Scapula_%d' %(j-1)].SectionAssignment(offset=0.0,
        offsetField='', offsetType=MIDDLE_SURFACE, region=
        mdb.models['Model-%d' %count].parts['Scapula_%d' %(j-1)].sets['TOTAL'], sectionName=
        'Section-1', thicknessAssignment=FROM_SECTION)
```

```
    mdb.models['Model-%d' %count].rootAssembly.translate(instanceList=('Humerus_%d-1' %(j-1), ), vector=(-0.0, (-0.0),(7.0)))
    #if j == 2:
    #   mdb.models['Model-%d' %count].rootAssembly.translate(instanceList=('Humerus_%d-1' %(j-1), ), vector=(-0.0, (0.0),7.0))
    mdb.models['Model-%d' %count].ExplicitDynamicsStep(name='Step-1', previous='Initial')
    mdb.models['Model-%d' %count].OperatorFilter(halt=ON, limit=0.01, name='Filter-1',
        operation=MAX)
    mdb.models['Model-%d' %count].fieldOutputRequests['F-Output-1'].setValues(filter=
        'Filter-1', variables=('S', ))
    mdb.models['Model-%d' %count].ContactProperty('IntProp-1')
    mdb.models['Model-%d' %count].interactionProperties['IntProp-1'].TangentialBehavior(
        formulation=FRICTIONLESS)
    mdb.models['Model-%d' %count].interactionProperties['IntProp-1'].NormalBehavior(
        allowSeparation=ON, constraintEnforcementMethod=DEFAULT,
        pressureOverclosure=HARD)
    mdb.models['Model-%d' %count].SurfaceToSurfaceContactExp(clearanceRegion=None,
        createStepName='Step-1', datumAxis=None, initialClearance=OMIT,
        interactionProperty='IntProp-1', master=
        mdb.models['Model-%d' %count].rootAssembly.instances['Scapula_%d-1' %(j-1)].surfaces['CONT_SCAP']
        , mechanicalConstraint=KINEMATIC, name='Int-1', slave=
        mdb.models['Model-%d' %count].rootAssembly.instances['Humerus_%d-1' %(j-1)].surfaces['CONT_HUM']
        , sliding=FINITE)
    mdb.models['Model-%d' %count].BodyForce(comp3=-0.1, createStepName='Step-1', name=
        'Load-1', region=
        mdb.models['Model-%d' %count].rootAssembly.instances['Humerus_%d-1' %(j-1)].sets['TOTAL'])
```

```
mdb.models['Model-%d' %count].EncastreBC(createStepName='Initial', localCsys=None,
    name='BC-1', region=
    mdb.models['Model-%d' %count].rootAssembly.instances['Scapula_%d-1' %(j-1)].sets['FIX'])
mdb.models['Model-%d' %count].DisplacementBC(amplitude=UNSET, createStepName='Step-1',
    distributionType=UNIFORM, fieldName='', fixed=OFF, localCsys=None, name=
    'BC-2', region=
    mdb.models['Model-%d' %count].rootAssembly.instances['Humerus_%d-1' %(j-1)].sets['DISP'],
    u1=0.0, u2=0.0, u3=UNSET, ur1=0.0, ur2=0.0, ur3=0.0)
pos = 0
pos_m = [0] * 1
n = 0
import string
mdb.models['Model-%d' %count].keywordBlock.synchVersions()
for block in mdb.models['Model-%d' %count].keywordBlock.sieBlocks:
    if string.lower(block[0:len('*Output, field')])==string.lower('*Output, field'):
        pos_m[n] = pos
        n = n+1
    pos = pos +1
mdb.models['Model-%d' %count].keywordBlock.replace(pos_m[0] + 1, """
*Element Output,POSITION= NODES, directions=YES
S,
""")
mdb.Job(activateLoadBalancing=False, atTime=None, contactPrint=OFF,
    description='', echoPrint=OFF, explicitPrecision=SINGLE, historyPrint=OFF,
    memory=90, memoryUnits=PERCENTAGE, model='Model-%d' %count, modelPrint=OFF,
    multiprocessingMode=DEFAULT, name='dyn_%d' %j, nodalOutputPrecision=SINGLE,
    numCpus=1, numDomains=1, parallelizationMethodExplicit=DOMAIN, queue=None,
    scratch='', type=ANALYSIS, userSubroutine='', waitHours=0, waitMinutes=0)
```

```
mdb.jobs['dyn_%d' %j].submit(consistencyChecking=OFF)
mdb.jobs['dyn_%d' %j].waitForCompletion()
odbWritea=openOdb(path="C:\Nikhil_D\Research\Abaqus\Rat_bone_geometry\Asmbly_64\dyn_%d.odb" %j)
Phum = odbWritea.rootAssembly.instances['HUMERUS_%d-1' %(j-1)].nodeSets['TOTAL']
fieldValuesa_Phum = odbWritea.steps['Step-1'].frames[-1].fieldOutputs['S_FILTER-1'].getSubset(region=Phum).values
noda = [0] * (len(fieldValuesa_Phum))
for k in range(len(fieldValuesa_Phum)):
    noda[k] = fieldValuesa_Phum[k].nodeLabel
hum_data = [0] * (len(fieldValuesa_Phum))
for n in range(len(fieldValuesa_Phum)):
    hum_data[n] = fieldValuesa_Phum[n].data[1]
min_hd = min(hum_data)
min_index_hd = hum_data.index(min_hd)
min_nod = noda[min_index_hd]
Pscap = odbWritea.rootAssembly.instances['SCAPULA_%d-1' %(j-1)].nodeSets['TOTAL']
fieldValuesa_scap = odbWritea.steps['Step-1'].frames[-1].fieldOutputs['S_FILTER-1'].getSubset(region=Pscap).values
noda = [0] * (len(fieldValuesa_scap))
for k in range(len(fieldValuesa_scap)):
    noda[k] = fieldValuesa_scap[k].nodeLabel
scap_data = [0] * (len(fieldValuesa_scap))
for n in range(len(fieldValuesa_scap)):
    scap_data[n] = fieldValuesa_scap[n].data[1]
min_sc = min(scap_data)
min_index_sc = scap_data.index(min_sc)
min_nod_sc = noda[min_index_sc]
```

```
odbWritea.save()
odbWritea.close()
mdb.models['Model-%d' %(count+1)].Material(name='Material-1')
mdb.models['Model-%d' %(count+1)].materials['Material-1'].Density(table=((0.007, ), ))
mdb.models['Model-%d' %(count+1)].materials['Material-1'].Elastic(table=((1.1, 0.45), ))
mdb.models['Model-%d' %(count+1)].materials['Material-1'].Expansion(table=((8e-06, ), ))
mdb.models['Model-%d' %(count+1)].HomogeneousSolidSection(material='Material-1', name=
    'Section-1', thickness=None)
mdb.models['Model-%d' %(count+1)].parts['Humerus_%d' %(j-1)].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=
    mdb.models['Model-%d' %(count+1)].parts['Humerus_%d' %(j-1)].sets['TOTAL'], sectionName=
    'Section-1', thicknessAssignment=FROM_SECTION)
mdb.models['Model-%d' %(count+1)].parts['Scapula_%d' %(j-1)].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=
    mdb.models['Model-%d' %(count+1)].parts['Scapula_%d' %(j-1)].sets['TOTAL'], sectionName=
    'Section-1', thicknessAssignment=FROM_SECTION)
mdb.models['Model-%d' %(count+1)].rootAssembly.DatumCsysByDefault(CARTESIAN)
mdb.models['Model-%d' %(count+1)].rootAssembly.translate(instanceList=('Humerus_%d-1' %(j-1), ), vector=(-(0.0), (-(0.0)), (7.0)))
#mdb.models['Model-%d' %(count+1)].rootAssembly.translate(instanceList=('Humerus_%d-1' %(j-1), ), vector=(-(0.5), (-(0.5)), 0.0))
#if j == 2:
#   mdb.models['Model-%d' %(count+1)].rootAssembly.translate(instanceList=('Humerus_%d-1' %(j-1), ), vector=(-0.0, (0.0),7.0))
dist_x = ((mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[min_nod_sc-1].coordinates[0]) - (mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[min_nod-1].coordinates[0]))
```

```
dist_y = ((mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[min_nod_sc-1].coordinates[1]) - (mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[min_nod-1].coordinates[1]))
dist_z = ((mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[min_nod_sc-1].coordinates[2]) - (mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[min_nod-1].coordinates[2]))
#mdb.models['Model-%d' %(count+1)].rootAssembly.translate(instanceList=('Humerus_%d-1' %(j-1), ), vector=(-(dist_x*2), -(dist_y*2), 0))
mdb.models['Model-%d' %(count+1)].StaticStep(initialInc=1.0, maxNumInc=1000,
    name='Step-1', nlgeom=ON, previous='Initial')
mdb.models['Model-%d' %(count+1)].fieldOutputRequests['F-Output-1'].setValues(variables=(
    'PRESSONLY', 'U'))
mdb.models['Model-%d' %(count+1)].ContactProperty('IntProp-1')
mdb.models['Model-%d' %(count+1)].interactionProperties['IntProp-1'].TangentialBehavior(
    formulation=FRICTIONLESS)
mdb.models['Model-%d' %(count+1)].interactionProperties['IntProp-1'].NormalBehavior(
    allowSeparation=ON, constraintEnforcementMethod=DEFAULT,
    pressureOverclosure=HARD)
mdb.models['Model-%d' %(count+1)].SurfaceToSurfaceContactStd(adjustMethod=NONE,
    clearanceRegion=None, createStepName='Step-1', datumAxis=None,
    initialClearance=OMIT, interactionProperty='IntProp-1', master=
    mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Scapula_%d-1' %(j-1)].surfaces['CONT_SCAP']
    , name='Int-1', slave=
    mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Humerus_%d-1' %(j-1)].surfaces['CONT_HUM']
    , sliding=FINITE, thickness=ON)
mdb.models['Model-%d' %(count+1)].EncastreBC(createStepName='Initial', localCsys=None,
```

```
    name='BC-1', region=
    mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Scapula_%d-1' %(j-1)].sets['FIX'])
mdb.models['Model-%d' %(count+1)].DisplacementBC(amplitude=UNSET, createStepName='Step-1',
    distributionType=UNIFORM, fieldName='', fixed=OFF, localCsys=None, name=
    'BC-2', region=
    mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Humerus_%d-1' %(j-1)].sets['DISP'],
    u1=(0.06), u2=(0.05), u3=-((-1*dist_z) + 0.08), ur1=0.0, ur2=0.0, ur3=0.0) #from muscle scaling opensim simulation a compressive force of 235 N was found. Joint stiffness of 0.0005N/mm was assumed for the simulation.
pos = 0
pos_m = [0] * 1
n = 0
import string
mdb.models['Model-%d' %(count+1)].keywordBlock.synchVersions()
for block in mdb.models['Model-%d' %(count+1)].keywordBlock.sieBlocks:
    if string.lower(block[0:len('*Output, field')])==string.lower('*Output, field'):
        pos_m[n] = pos
        n = n+1
    pos = pos +1
mdb.models['Model-%d' %(count+1)].keywordBlock.replace(pos_m[0] + 2, """
*Element Output,POSITION= NODES, directions=YES
PRESSONLY
""")
mdb.save()
mdb.Job(atTime=None, contactPrint=OFF, description='', echoPrint=OFF,
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF,
    memory=90, memoryUnits=PERCENTAGE, model='Model-%d' %(count+1), modelPrint=OFF,
```

```
    multiprocessingMode=DEFAULT, name='static_%d' %j, nodalOutputPrecision=SINGLE,
    numCpus=8, numDomains=8, numGPUs=0, queue=None, scratch='', type=ANALYSIS,
    userSubroutine='', waitHours=0, waitMinutes=0)
mdb.jobs['static_%d' %j].submit(consistencyChecking=OFF)
mdb.jobs['static_%d' %j].waitForCompletion()
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from optimization import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
from odbAccess import *
from abaqusConstants import *
# Opening the odb file
odbWritea=openOdb(path="C:\Nikhil_D\Research\Abaqus\Rat_bone_geometry\Asmbly_64\static_%d.odb" %j , readOnly = False)
#checking the steps in the file
#for stepName in odbWritea.steps.keys():
 #   print ' ', stepName
Phum = odbWritea.rootAssembly.instances['HUMERUS_%d-1' %(j-1)].nodeSets['TOTAL']
```

```
fieldValuesa_Phum = odbWritea.steps['Step-1'].frames[-1].fieldOutputs['PRESSONLY'].getSubset(region=Phum).values
# Creating a new step to write the PRESSONLY output as NODAL TEMPERATURE
newResultsStepa=odbWritea.Step(name='Test-a',description='User Defined Results',domain=TIME, timePeriod=1)
newResultsFramea=newResultsStepa.Frame(incrementNumber=1,frameValue=0.1)
newResultsFielda=newResultsFramea.FieldOutput(name='NT11',description='User Defined Results', type=SCALAR)
# Following is a code take the average value of all the assigned values to one particular node
import numpy
import math
noda = [0] * (len(fieldValuesa_Phum))
for k in range(len(fieldValuesa_Phum)):
    noda[k] = fieldValuesa_Phum[k].nodeLabel
hum_data = [0] * (len(fieldValuesa_Phum))
for n in range(len(fieldValuesa_Phum)):
    hum_data[k] = fieldValuesa_Phum[n].data
#max_hd = max(hum_data)
#expo_hd = (log10(max_hd) +2)
nda = [0] * (max(noda))
Data = [0] * (max(noda))
Dispdata = [0] * (max(noda))
k = 11000
for L in range(max(noda)):
    D = []
    Y_Cd = 1 #((mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[L].coordinates[1])-
```

```
21.968)/5 # 5 is derived from uppermost z coordinate of vertex of humerus  minus lowermost z coordinate of vertex and 21.968 is the z coordinate of the lower point
    Cd = k *((0.14) -(0.87* Y_Cd) +(4.4* (Y_Cd ** 2)) - (2.66 * (Y_Cd ** 3)))
    for m in range(len(fieldValuesa_Phum)):
        if noda[m] == L+1 :
            if math.isnan(fieldValuesa_Phum[m].data) == False:
                D.append(fieldValuesa_Phum[m].data)
    Data[L] = -1*(numpy.mean(D))*Cd*mech_hum
    nda[L] = L+1
    Dispdata[L] = [Data[L]]
max_stress_hum = max(Data)
max_node_hum = nda[Data.index(max_stress_hum)]
## mdb.models['Model-%d' %count].rootAssembly.instances['Part-1-1'].nodes[0].coordinates[1]
# Compiling the nodal data and assigning the nodal temperature values in the new step
newResultsFielda.addData(position= NODAL , instance = openOdb(r'C:\Nikhil_D\Research\Abaqus\Rat_bone_geometry\Asmbly_64\static_%d.odb' %j).rootAssembly.instances
    ['HUMERUS_%d-1' %(j-1)] , labels = nda , data = Dispdata)
Pscap = odbWritea.rootAssembly.instances['SCAPULA_%d-1' %(j-1)].nodeSets['TOTAL']
fieldValuesa_Pscap = odbWritea.steps['Step-1'].frames[-1].fieldOutputs['PRESSONLY'].getSubset(region=Pscap).values
import numpy
import math
noda_S = [0] * (len(fieldValuesa_Pscap))
for k in range(len(fieldValuesa_Pscap)):
    noda_S[k] = fieldValuesa_Pscap[k].nodeLabel
scap_data = [0] * (len(fieldValuesa_Pscap))
for n in range(len(fieldValuesa_Pscap)):
    scap_data[k] = fieldValuesa_Pscap[n].data
```

```
#max_sd = max(scap_data)
#expo_sd = (log10(max_sd) +2)
nda_S = [0] * (max(noda_S))
Data_S = [0] * (max(noda_S))
Dispdata_S = [0] * (max(noda_S))
k = 11000
for L in range(max(noda_S)):
    D_S = []
    #Y_Cd_S = 1
    #Y_Cd = ((mdb.models['Model-%d' %count].rootAssembly.instances['Humerus_8_4_cut_nosurfnish-1'].nodes[L].coordinates[1])-21.968)/5 # 5 is derived from uppermost z coordinate of vertex of humerus  minus lowermost z coordinate of vertex and 21.968 is the z coordinate of the lower point
    Cd_S = k *((0.14) -(0.87* 1) +(4.4* (1 ** 2)) - (2.66 * (1 ** 3))) #Y_Cd is assumed 1 for time being
    for m in range(len(fieldValuesa_Pscap)):
        if noda_S[m] == L+1 :
            if math.isnan(fieldValuesa_Pscap[m].data) == False:
                D_S.append(fieldValuesa_Pscap[m].data)
    Data_S[L] = -1*(numpy.mean(D_S))*Cd_S*mech_scap
    nda_S[L] = L+1
    Dispdata_S[L] = [Data_S[L]]
max_stress_scap = max(Data_S)
#max_node_scap = nda[Data_S.index(max_stress_scap)]
newResultsFielda.addData(position= NODAL , instance = openOdb(r'C:\Nikhil_D\Research\Abaqus\Rat_bone_geometry\Asmbly_64\static_%d.odb' %j).rootAssembly.instances
    ['SCAPULA_%d-1' %(j-1)] , labels = nda_S , data = Dispdata_S)
odbWritea.save()
odbWritea.close()
```

```
# Importing the first stress analysis cae model
mdb.openAuxMdb(pathName=
    'C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_64/M1.cae')
mdb.copyAuxMdbModel(fromName='Model-%d' %(count+1), toName='Model-%d' %(count+2)) # names of the new model can be changed
mdb.copyAuxMdbModel(fromName='Model-%d' %(count+1), toName='Model-%d' %(count+2))
mdb.closeAuxMdb()
mdb.models['Model-%d' %(count+2)].boundaryConditions['BC-2'].setValues(u1=0.0)
mdb.models['Model-%d' %(count+2)].boundaryConditions['BC-2'].setValues(u2=0.0)
mdb.models['Model-%d' %(count+2)].interactions['Int-1'].suppress()
mdb.models['Model-%d' %(count+2)].Temperature(absoluteExteriorTolerance=0.0,
    beginIncrement=None, beginStep=2, createStepName='Step-1',
    distributionType=FROM_FILE, endIncrement=None, endStep=2,
    exteriorTolerance=0.05, fileName=
    'C:\Nikhil_D\Research\Abaqus\Rat_bone_geometry\Asmbly_64\static_%d.odb' %j,
    interpolate=OFF, name='Predefined Field-1')
# Creating and submitting the job
mdb.Job(atTime=None, contactPrint=OFF, description='', echoPrint=OFF,
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF,
    memory=90, memoryUnits=PERCENTAGE, model='Model-%d' %(count+2), modelPrint=OFF,
    multiprocessingMode=DEFAULT, name='Mechgrowth_%d' %j, nodalOutputPrecision=SINGLE,
    numCpus=8, numDomains=8, numGPUs=0, queue=None, scratch='', type=ANALYSIS,
    userSubroutine='', waitHours=0, waitMinutes=0)
mdb.jobs['Mechgrowth_%d' %j].submit(consistencyChecking=OFF)
mdb.jobs['Mechgrowth_%d' %j].waitForCompletion()
# Importing the first stress analysis cae model
mdb.openAuxMdb(pathName=
    'C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_64/M1.cae')
```

```
mdb.copyAuxMdbModel(fromName='Model-%d' %(count+1), toName='Model-%d' %(count+3)) # names of the new model can be changed
mdb.copyAuxMdbModel(fromName='Model-%d' %(count+1), toName='Model-%d' %(count+3))
mdb.closeAuxMdb()
mdb.models['Model-%d' %(count+3)].boundaryConditions['BC-2'].setValues(u1=0.0)
mdb.models['Model-%d' %(count+3)].boundaryConditions['BC-2'].setValues(u2=0.0)
mdb.models['Model-%d' %(count+3)].interactions['Int-1'].suppress()
mdb.models['Model-%d' %(count+3)].rootAssembly.DatumCsysByThreePoints(coordSysType=
    CARTESIAN, name='scap_cs', origin=(9.156945,27.0467,22.118969), point1=(9.067071,25.469379,21.794357)
    , point2=(9.309071,25.873672,22.668428))
mdb.models['Model-%d' %(count+3)].rootAssembly.DatumCsysByThreePoints(coordSysType=
    CARTESIAN, name='HUm_cs', origin= mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[513].coordinates, point1= mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[520].coordinates,
    point2= mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[240].coordinates)
y_hum = sqrt((((mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[513].coordinates[0]) - (mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[240].coordinates[0])) **2) +
    (((mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[513].coordinates[1]) - (mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[240].coordinates[1]))**2) +
    (((mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[513].coordinates[2]) - (mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[240].coordinates[2]))**2))
mdb.models['Model-%d' %(count+3)].ExpressionField(description='', expression=
```

```
    '(11000*(0.14 -(0.87*((-Y/y_hum)**1))+(4.4*((-Y/y_hum)**2))-(2.66*((-Y/y_hum)**3))))'
    , localCsys=mdb.models['Model-%d' %(count+3)].rootAssembly.datums[mdb.models['Model-%d' %(count+3)].rootAssembly.features['HUm_cs'].id], name=
    'AnalyticalField-1')
mdb.models['Model-%d' %(count+3)].Temperature(createStepName='Step-1',
    crossSectionDistribution=CONSTANT_THROUGH_THICKNESS, distributionType=FIELD
    , field='AnalyticalField-1', magnitudes=(bio_hum, ), name='Hum_temp', region=
    mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].sets['TOTAL'])
mdb.models['Model-%d' %(count+3)].TabularAmplitude(data=((0.0, 0.0), (1.0, 1.0)), name=
'Amp-1', smooth=SOLVER_DEFAULT, timeSpan=STEP)
mdb.models['Model-%d' %(count+3)].predefinedFields['Hum_temp'].setValues(amplitude='Amp-1')
#mdb.models['Model-%d' %(count+3)].ExpressionField(description='', expression=
#    '(11000*(0.14 -(0.87*((-Z/3.0)**1))+(4.4*((-Z/3.0)**2))-(2.66*((-Z/3.0)**3))))'
#    , localCsys=mdb.models['Model-%d' %(count+3)].rootAssembly.datums[mdb.models['Model-%d' %(count+3)].rootAssembly.features['scap_cs'].id], name=
#    'AnalyticalField-2')
#mdb.models['Model-%d' %(count+3)].Temperature(createStepName='Step-1',
#    crossSectionDistribution=CONSTANT_THROUGH_THICKNESS, distributionType=FIELD
#    , field='AnalyticalField-2', magnitudes=(0.02, ), name='Scap_temp', region=
#    mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].sets['TOTAL'])
mdb.models['Model-%d' %(count+3)].Temperature(amplitude='Amp-1', createStepName='Step-1',
    crossSectionDistribution=CONSTANT_THROUGH_THICKNESS, distributionType=
    UNIFORM, magnitudes=(bio_scap, ), name='Scap_temp', region=
    mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].sets['TOTAL'])
mdb.Job(atTime=None, contactPrint=OFF, description='', echoPrint=OFF,
```

```
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF,
    memory=90, memoryUnits=PERCENTAGE, model='Model-%d' %(count+3), modelPrint=OFF,
    multiprocessingMode=DEFAULT, name='Biogrowth_%d' %j, nodalOutputPrecision=SINGLE,
    numCpus=8, numDomains=8, numGPUs=0, queue=None, scratch='', type=ANALYSIS,
    userSubroutine='', waitHours=0, waitMinutes=0)
mdb.jobs['Biogrowth_%d' %j].submit(consistencyChecking=OFF)
mdb.jobs['Biogrowth_%d' %j].waitForCompletion()
mdb.Model(modelType=STANDARD_EXPLICIT, name='Model-%d' %(count+4))
mdb.models['Model-%d' %(count+4)].PartFromOdb(frame=-1, instance='HUMERUS_%d-1' %(j-1), name=
    'Humerus_Mech_%d' %j, odb=session.openOdb(
    r'C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_64/Mechgrowth_%d.odb' %j), shape=
    DEFORMED, step=0)
mdb.models['Model-%d' %(count+4)].PartFromOdb(frame=-1, instance='SCAPULA_%d-1' %(j-1), name=
    'Scapula_Mech_%d' %j, odb=session.openOdb(
    r'C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_64/Mechgrowth_%d.odb' %j), shape=
    DEFORMED, step=0)
mdb.models['Model-%d' %(count+4)].PartFromOdb(instance='HUMERUS_%d-1' %(j-1), name='Humerus_%d' %(j-1),
    odb=session.openOdb(
    r'C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_64/Biogrowth_%d.odb' %j))
mdb.models['Model-%d' %(count+4)].PartFromOdb(instance='SCAPULA_%d-1' %(j-1), name='Scapula_%d' %(j-1),
    odb=session.openOdb(
    r'C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_64/Biogrowth_%d.odb' %j))
mdb.models['Model-%d' %(count+4)].PartFromOdb(frame=-1, instance='HUMERUS_%d-1' %(j-1), name='Humerus_Bio_%d' %j,
```

```
    odb=session.openOdb(
    r'C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_64/Biogrowth_%d.odb' %j),
    shape=DEFORMED, step=0)
mdb.models['Model-%d' %(count+4)].PartFromOdb(frame=-1, instance='SCAPULA_%d-1' %(j-1), name='Scapula_Bio_%d' %j,
    odb=session.openOdb(
    r'C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_64/Biogrowth_%d.odb' %j),
    shape=DEFORMED, step=0)
mdb.models['Model-%d' %(count+4)].Part(name='Humerus_%d' %j, objectToCopy=
    mdb.models['Model-%d' %(count+4)].parts['Humerus_%d' %(j-1)])
mdb.models['Model-%d' %(count+4)].Part(name='Scapula_%d' %j, objectToCopy=
    mdb.models['Model-%d' %(count+4)].parts['Scapula_%d' %(j-1)])
# trial loop
for i in range(len(mdb.models['Model-%d' %(count+4)].parts['Humerus_Mech_%d' %j].nodes)): ### for some weird reason +1 needs to be added only in this case
     mdb.models['Model-%d' %(count+4)].parts['Humerus_%d' %j].editNode(coordinate1=(mdb.models['Model-%d' %(count+4)].parts['Humerus_Mech_%d' %j].nodes[i].coordinates[0]+
        ((mdb.models['Model-%d' %(count+4)].parts['Humerus_Bio_%d' %j].nodes[i].coordinates[0])-
        (mdb.models['Model-%d' %(count+4)].parts['Humerus_%d' %(j-1)].nodes[i].coordinates[0]))),
        coordinate2 = (mdb.models['Model-%d' %(count+4)].parts['Humerus_Mech_%d' %j].nodes[i].coordinates[1]+
        ((mdb.models['Model-%d' %(count+4)].parts['Humerus_Bio_%d' %j].nodes[i].coordinates[1])-
        (mdb.models['Model-%d' %(count+4)].parts['Humerus_%d' %(j-1)].nodes[i].coordinates[1]))),
        coordinate3 = (mdb.models['Model-%d' %(count+4)].parts['Humerus_Mech_%d' %j].nodes[i].coordinates[2]+
        ((mdb.models['Model-%d' %(count+4)].parts['Humerus_Bio_%d' %j].nodes[i].coordinates[2])-
```

```
        (mdb.models['Model-%d' %(count+4)].parts['Humerus_%d' %(j-1)].nodes[i].coordinates[2]))),
        nodes=mdb.models['Model-%d' %(count+4)].parts['Humerus_%d' %j].nodes[i:(i+1)])
for i in range(len(mdb.models['Model-%d' %(count+4)].parts['Scapula_Mech_%d' %j].nodes)): ### check for the instances in the odb file and add the proper instance coordinate
    mdb.models['Model-%d' %(count+4)].parts['Scapula_%d' %j].editNode(coordinate1=(mdb.models['Model-%d' %(count+4)].parts['Scapula_Mech_%d' %j].nodes[i].coordinates[0]+
        ((mdb.models['Model-%d' %(count+4)].parts['Scapula_Bio_%d' %j].nodes[i].coordinates[0])-
        (mdb.models['Model-%d' %(count+4)].parts['Scapula_%d' %(j-1)].nodes[i].coordinates[0]))),
        coordinate2 = (mdb.models['Model-%d' %(count+4)].parts['Scapula_Mech_%d' %j].nodes[i].coordinates[1]+
        ((mdb.models['Model-%d' %(count+4)].parts['Scapula_Bio_%d' %j].nodes[i].coordinates[1])-
        (mdb.models['Model-%d' %(count+4)].parts['Scapula_%d' %(j-1)].nodes[i].coordinates[1]))),
        coordinate3 = (mdb.models['Model-%d' %(count+4)].parts['Scapula_Mech_%d' %j].nodes[i].coordinates[2]+
        ((mdb.models['Model-%d' %(count+4)].parts['Scapula_Bio_%d' %j].nodes[i].coordinates[2])-
        (mdb.models['Model-%d' %(count+4)].parts['Scapula_%d' %(j-1)].nodes[i].coordinates[2]))),
        nodes=mdb.models['Model-%d' %(count+4)].parts['Scapula_%d' %j].nodes[i:(i+1)])
count = count+4
mdb.Model(name='Model-%d' %(count+1), objectToCopy=mdb.models['Model-%d' %count])
#mdb.Model(name='Model-%d' %(count+2), objectToCopy=mdb.models['Model-%d' %count])
mdb.models['Model-%d' %count].Material(name='Material-1')
mdb.models['Model-%d' %count].materials['Material-1'].Density(table=((0.007, ), ))
mdb.models['Model-%d' %count].materials['Material-1'].Elastic(table=((1.1, 0.45), ))
mdb.models['Model-%d' %count].materials['Material-1'].Expansion(table=((8e-06, ), ))
mdb.models['Model-%d' %count].HomogeneousSolidSection(material='Material-1', name=
    'Section-1', thickness=None)
```

```
mdb.models['Model-%d' %count].parts['Humerus_%d' %j].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=
    mdb.models['Model-%d' %count].parts['Humerus_%d' %j].sets['TOTAL'], sectionName=
    'Section-1', thicknessAssignment=FROM_SECTION)
mdb.models['Model-%d' %count].parts['Scapula_%d' %j].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=
    mdb.models['Model-%d' %count].parts['Scapula_%d' %j].sets['TOTAL'], sectionName=
    'Section-1', thicknessAssignment=FROM_SECTION)
mdb.models['Model-%d' %count].rootAssembly.Instance(dependent=ON, name='Humerus_%d-1' %j,
    part=mdb.models['Model-%d' %count].parts['Humerus_%d' %j])
mdb.models['Model-%d' %count].rootAssembly.Instance(dependent=ON, name='Scapula_%d-1' %j,
    part=mdb.models['Model-%d' %count].parts['Scapula_%d' %j])
#mdb.models['Model-%d' %count].rootAssembly.translate(instanceList=('Humerus_%d-1' %j, ), vector=((0.0), (0.0),(j+0.5)))
#mdb.models['Model-%d' %count].rootAssembly.translate(instanceList=('Humerus_%d-1' %j, ), vector=((0.5), (0.5),0.0))
mdb.save()
mdb.saveAs(pathName='C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_64/M1_%d.cae' %j)
mdb.saveAs(pathName='C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_64/M1.cae')
```

FEA modeling script:

For muscle shortening simulations, following script was developed in Python and implemented in Abaqus.

```
from part import *
```

```
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from optimization import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *

count = 1
mech_hum = 35
mech_scap = 35
bio_hum = 0.05
bio_scap = 750

for j in range(1,15):
    #if (j-1)%3 == 0:
    #   zmove = 0.1
    #else:
    #   zmove = 0.0
    mdb.openAuxMdb(pathName=
        'C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_65/M1.cae')
    mdb.copyAuxMdbModel(fromName='Model-%d' %count, toName='Model-%d' %(count+1)) # names of the new model can be changed
```

```
mdb.copyAuxMdbModel(fromName='Model-%d' %count, toName='Model-%d' %(count+1))
mdb.closeAuxMdb()
mdb.models['Model-%d' %count].Material(name='Material-1')
mdb.models['Model-%d' %count].materials['Material-1'].Density(table=((0.007, ), ))
mdb.models['Model-%d' %count].materials['Material-1'].Elastic(table=((1.1, 0.45), ))
mdb.models['Model-%d' %count].materials['Material-1'].Expansion(table=((8e-06, ), ))
mdb.models['Model-%d' %count].HomogeneousSolidSection(material='Material-1', name=
    'Section-1', thickness=None)
mdb.models['Model-%d' %count].parts['Humerus_%d' %(j-1)].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=
    mdb.models['Model-%d' %count].parts['Humerus_%d' %(j-1)].sets['TOTAL'], sectionName=
    'Section-1', thicknessAssignment=FROM_SECTION)
mdb.models['Model-%d' %count].parts['Scapula_%d' %(j-1)].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=
    mdb.models['Model-%d' %count].parts['Scapula_%d' %(j-1)].sets['TOTAL'], sectionName=
    'Section-1', thicknessAssignment=FROM_SECTION)
mdb.models['Model-%d' %count].rootAssembly.translate(instanceList=('Humerus_%d-1' %(j-1), ), vector=(-0.0, (-0.0),(4.0)))
if j == 2:
    mdb.models['Model-%d' %count].rootAssembly.translate(instanceList=('Humerus_%d-1' %(j-1), ), vector=(-0.0, (0.0),4.0))
if j%2 == 0:
    mdb.models['Model-%d' %count].rootAssembly.translate(instanceList=('Humerus_%d-1' %(j-1), ), vector=(-0.0, -(0.2),0.0))
mdb.models['Model-%d' %count].ExplicitDynamicsStep(name='Step-1', previous='Initial')
mdb.models['Model-%d' %count].OperatorFilter(halt=ON, limit=0.01, name='Filter-1',
    operation=MAX)
mdb.models['Model-%d' %count].fieldOutputRequests['F-Output-1'].setValues(filter=
```

```
    'Filter-1', variables=('S', ))
mdb.models['Model-%d' %count].ContactProperty('IntProp-1')
mdb.models['Model-%d' %count].interactionProperties['IntProp-1'].TangentialBehavior(
    formulation=FRICTIONLESS)
mdb.models['Model-%d' %count].interactionProperties['IntProp-1'].NormalBehavior(
    allowSeparation=ON, constraintEnforcementMethod=DEFAULT,
    pressureOverclosure=HARD)
mdb.models['Model-%d' %count].SurfaceToSurfaceContactExp(clearanceRegion=None,
    createStepName='Step-1', datumAxis=None, initialClearance=OMIT,
    interactionProperty='IntProp-1', master=
    mdb.models['Model-%d' %count].rootAssembly.instances['Scapula_%d-1' %(j-
1)].surfaces['CONT_SCAP']
    , mechanicalConstraint=KINEMATIC, name='Int-1', slave=
    mdb.models['Model-%d' %count].rootAssembly.instances['Humerus_%d-1' %(j-
1)].surfaces['CONT_HUM']
    , sliding=FINITE)
mdb.models['Model-%d' %count].BodyForce(comp3=-0.1, createStepName='Step-1', name=
    'Load-1', region=
    mdb.models['Model-%d' %count].rootAssembly.instances['Humerus_%d-1' %(j-1)].sets['TOTAL'])
mdb.models['Model-%d' %count].EncastreBC(createStepName='Initial', localCsys=None,
    name='BC-1', region=
    mdb.models['Model-%d' %count].rootAssembly.instances['Scapula_%d-1' %(j-1)].sets['FIX'])
mdb.models['Model-%d' %count].DisplacementBC(amplitude=UNSET, createStepName='Step-1',
    distributionType=UNIFORM, fieldName='', fixed=OFF, localCsys=None, name=
    'BC-2', region=
    mdb.models['Model-%d' %count].rootAssembly.instances['Humerus_%d-1' %(j-1)].sets['DISP'],
    u1=0.0, u2=0.0, u3=UNSET, ur1=0.0, ur2=0.0, ur3=0.0)
pos = 0
```

```
pos_m = [0] * 1
n = 0
import string
mdb.models['Model-%d' %count].keywordBlock.synchVersions()
for block in mdb.models['Model-%d' %count].keywordBlock.sieBlocks:
    if string.lower(block[0:len('*Output, field')])==string.lower('*Output, field'):
        pos_m[n] = pos
        n = n+1
    pos = pos +1
mdb.models['Model-%d' %count].keywordBlock.replace(pos_m[0] + 1, """
*Element Output,POSITION= NODES, directions=YES
S,
""")
mdb.Job(activateLoadBalancing=False, atTime=None, contactPrint=OFF,
    description='', echoPrint=OFF, explicitPrecision=SINGLE, historyPrint=OFF,
    memory=90, memoryUnits=PERCENTAGE, model='Model-%d' %count, modelPrint=OFF,
    multiprocessingMode=DEFAULT, name='dyn_%d' %j, nodalOutputPrecision=SINGLE,
    numCpus=1, numDomains=1, parallelizationMethodExplicit=DOMAIN, queue=None,
    scratch='', type=ANALYSIS, userSubroutine='', waitHours=0, waitMinutes=0)
mdb.jobs['dyn_%d' %j].submit(consistencyChecking=OFF)
mdb.jobs['dyn_%d' %j].waitForCompletion()
odbWritea=openOdb(path="C:\Nikhil_D\Research\Abaqus\Rat_bone_geometry\Asmbly_65\dyn_%d.odb" %j)
Phum = odbWritea.rootAssembly.instances['HUMERUS_%d-1' %(j-1)].nodeSets['TOTAL']
fieldValuesa_Phum = odbWritea.steps['Step-1'].frames[-1].fieldOutputs['S_FILTER-1'].getSubset(region=Phum).values
noda = [0] * (len(fieldValuesa_Phum))
for k in range(len(fieldValuesa_Phum)):
```

```
    noda[k] = fieldValuesa_Phum[k].nodeLabel
hum_data = [0] * (len(fieldValuesa_Phum))
for n in range(len(fieldValuesa_Phum)):
    hum_data[n] = fieldValuesa_Phum[n].data[1]
min_hd = min(hum_data)
min_index_hd = hum_data.index(min_hd)
min_nod = noda[min_index_hd]
Pscap = odbWritea.rootAssembly.instances['SCAPULA_%d-1' %(j-1)].nodeSets['TOTAL']
fieldValuesa_scap = odbWritea.steps['Step-1'].frames[-1].fieldOutputs['S_FILTER-1'].getSubset(region=Pscap).values
noda = [0] * (len(fieldValuesa_scap))
for k in range(len(fieldValuesa_scap)):
    noda[k] = fieldValuesa_scap[k].nodeLabel
scap_data = [0] * (len(fieldValuesa_scap))
for n in range(len(fieldValuesa_scap)):
    scap_data[n] = fieldValuesa_scap[n].data[1]
min_sc = min(scap_data)
min_index_sc = scap_data.index(min_sc)
min_nod_sc = noda[min_index_sc]
odbWritea.save()
odbWritea.close()
mdb.models['Model-%d' %(count+1)].Material(name='Material-1')
mdb.models['Model-%d' %(count+1)].materials['Material-1'].Density(table=((0.007, ), ))
mdb.models['Model-%d' %(count+1)].materials['Material-1'].Elastic(table=((1.1, 0.45), ))
mdb.models['Model-%d' %(count+1)].materials['Material-1'].Expansion(table=((8e-06, ), ))
mdb.models['Model-%d' %(count+1)].HomogeneousSolidSection(material='Material-1', name=
    'Section-1', thickness=None)
mdb.models['Model-%d' %(count+1)].parts['Humerus_%d' %(j-1)].SectionAssignment(offset=0.0,
```

```
    offsetField='', offsetType=MIDDLE_SURFACE, region=
    mdb.models['Model-%d' %(count+1)].parts['Humerus_%d' %(j-1)].sets['TOTAL'], sectionName=
    'Section-1', thicknessAssignment=FROM_SECTION)
mdb.models['Model-%d' %(count+1)].parts['Scapula_%d' %(j-1)].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=
    mdb.models['Model-%d' %(count+1)].parts['Scapula_%d' %(j-1)].sets['TOTAL'], sectionName=
    'Section-1', thicknessAssignment=FROM_SECTION)
mdb.models['Model-%d' %(count+1)].rootAssembly.DatumCsysByDefault(CARTESIAN)
mdb.models['Model-%d' %(count+1)].rootAssembly.translate(instanceList=('Humerus_%d-1' %(j-1), ), vector=(-(0.0), (-(0.0)), (4.0)))
#mdb.models['Model-%d' %(count+1)].rootAssembly.translate(instanceList=('Humerus_%d-1' %(j-1), ), vector=(-(0.5), (-(0.5)), 0.0))
if j == 2:
    mdb.models['Model-%d' %(count+1)].rootAssembly.translate(instanceList=('Humerus_%d-1' %(j-1), ), vector=(-0.0, (0.0),4.0))
if j%2 == 0:
    mdb.models['Model-%d' %(count+1)].rootAssembly.translate(instanceList=('Humerus_%d-1' %(j-1), ), vector=(-0.0, -(0.2),0.0))
dist_x = ((mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[min_nod_sc-1].coordinates[0]) - (mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[min_nod-1].coordinates[0]))
dist_y = ((mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[min_nod_sc-1].coordinates[1]) - (mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[min_nod-1].coordinates[1]))
dist_z = ((mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[min_nod_sc-1].coordinates[2]) - (mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[min_nod-1].coordinates[2]))
```

```
#mdb.models['Model-%d' %(count+1)].rootAssembly.translate(instanceList=('Humerus_%d-1' %(j-1), ), vector=(-(dist_x*2), -(dist_y*2), 0))
mdb.models['Model-%d' %(count+1)].StaticStep(initialInc=1.0, maxNumInc=1000,
    name='Step-1', nlgeom=ON, previous='Initial')
mdb.models['Model-%d' %(count+1)].fieldOutputRequests['F-Output-1'].setValues(variables=(
    'PRESSONLY', 'U'))
mdb.models['Model-%d' %(count+1)].ContactProperty('IntProp-1')
mdb.models['Model-%d' %(count+1)].interactionProperties['IntProp-1'].TangentialBehavior(
    formulation=FRICTIONLESS)
mdb.models['Model-%d' %(count+1)].interactionProperties['IntProp-1'].NormalBehavior(
    allowSeparation=ON, constraintEnforcementMethod=DEFAULT,
    pressureOverclosure=HARD)
mdb.models['Model-%d' %(count+1)].SurfaceToSurfaceContactStd(adjustMethod=NONE,
    clearanceRegion=None, createStepName='Step-1', datumAxis=None,
    initialClearance=OMIT, interactionProperty='IntProp-1', master=
    mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Scapula_%d-1' %(j-1)].surfaces['CONT_SCAP']
    , name='Int-1', slave=
    mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Humerus_%d-1' %(j-1)].surfaces['CONT_HUM']
    , sliding=FINITE, thickness=ON)
mdb.models['Model-%d' %(count+1)].EncastreBC(createStepName='Initial', localCsys=None,
    name='BC-1', region=
    mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Scapula_%d-1' %(j-1)].sets['FIX'])
mdb.models['Model-%d' %(count+1)].DisplacementBC(amplitude=UNSET, createStepName='Step-1',
    distributionType=UNIFORM, fieldName='', fixed=OFF, localCsys=None, name=
    'BC-2', region=
```

```
    mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Humerus_%d-1' %(j-1)].sets['DISP'],
    u1=(0.06), u2=(0.0), u3=-((-1*dist_z) + 0.08), ur1=0.0, ur2=0.0, ur3=0.0) #from muscle scaling opensim simulation a compressive force of 235 N was found. Joint stiffness of 0.0005N/mm was assumed for the simulation.
pos = 0
pos_m = [0] * 1
n = 0
import string
mdb.models['Model-%d' %(count+1)].keywordBlock.synchVersions()
for block in mdb.models['Model-%d' %(count+1)].keywordBlock.sieBlocks:
    if string.lower(block[0:len('*Output, field')])==string.lower('*Output, field'):
        pos_m[n] = pos
        n = n+1
    pos = pos +1
mdb.models['Model-%d' %(count+1)].keywordBlock.replace(pos_m[0] + 2, """
*Element Output,POSITION= NODES, directions=YES
PRESSONLY
""")
mdb.save()
mdb.Job(atTime=None, contactPrint=OFF, description='', echoPrint=OFF,
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF,
    memory=90, memoryUnits=PERCENTAGE, model='Model-%d' %(count+1), modelPrint=OFF,
    multiprocessingMode=DEFAULT, name='static_%d' %j, nodalOutputPrecision=SINGLE,
    numCpus=8, numDomains=8, numGPUs=0, queue=None, scratch='', type=ANALYSIS,
    userSubroutine='', waitHours=0, waitMinutes=0)
mdb.jobs['static_%d' %j].submit(consistencyChecking=OFF)
mdb.jobs['static_%d' %j].waitForCompletion()
```

```
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from optimization import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
from odbAccess import *
from abaqusConstants import *
# Opening the odb file
odbWritea=openOdb(path="C:\Nikhil_D\Research\Abaqus\Rat_bone_geometry\Asmbly_65\static_%d.odb" %j , readOnly = False)
#checking the steps in the file
#for stepName in odbWritea.steps.keys():
 #   print ' ', stepName
Phum = odbWritea.rootAssembly.instances['HUMERUS_%d-1' %(j-1)].nodeSets['TOTAL']
fieldValuesa_Phum = odbWritea.steps['Step-1'].frames[-1].fieldOutputs['PRESSONLY'].getSubset(region=Phum).values
# Creating a new step to write the PRESSONLY output as NODAL TEMPERATURE
newResultsStepa=odbWritea.Step(name='Test-a',description='User Defined Results',domain=TIME, timePeriod=1)
newResultsFramea=newResultsStepa.Frame(incrementNumber=1,frameValue=0.1)
```

```
newResultsFielda=newResultsFramea.FieldOutput(name='NT11',description='User Defined Results', type=SCALAR)
# Following is a code take the average value of all the assigned values to one particular node
import numpy
import math
noda = [0] * (len(fieldValuesa_Phum))
for k in range(len(fieldValuesa_Phum)):
    noda[k] = fieldValuesa_Phum[k].nodeLabel
hum_data = [0] * (len(fieldValuesa_Phum))
for n in range(len(fieldValuesa_Phum)):
    hum_data[k] = fieldValuesa_Phum[n].data
#max_hd = max(hum_data)
#expo_hd = (log10(max_hd) +2)
nda = [0] * (max(noda))
Data = [0] * (max(noda))
Dispdata = [0] * (max(noda))
k = 11000
for L in range(max(noda)):
    D = []
    Y_Cd = 1 #((mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[L].coordinates[1])-21.968)/5 # 5 is derived from uppermost z coordinate of vertex of humerus  minus lowermost z coordinate of vertex and 21.968 is the z coordinate of the lower point
    Cd = k *((0.14) -(0.87* Y_Cd) +(4.4* (Y_Cd ** 2)) - (2.66 * (Y_Cd ** 3)))
    for m in range(len(fieldValuesa_Phum)):
        if noda[m] == L+1 :
            if math.isnan(fieldValuesa_Phum[m].data) == False:
                D.append(fieldValuesa_Phum[m].data)
```

```
    Data[L] = -1*(numpy.mean(D))*Cd*mech_hum
    nda[L] = L+1
    Dispdata[L] = [Data[L]]
max_stress_hum = max(Data)
max_node_hum = nda[Data.index(max_stress_hum)]
## mdb.models['Model-%d' %count].rootAssembly.instances['Part-1-1'].nodes[0].coordinates[1]
# Compiling the nodal data and assigning the nodal temperature values in the new step
newResultsFielda.addData(position= NODAL , instance = openOdb(r'C:\Nikhil_D\Research\Abaqus\Rat_bone_geometry\Asmbly_65\static_%d.odb' %j).rootAssembly.instances
    ['HUMERUS_%d-1' %(j-1)] , labels = nda , data = Dispdata)
Pscap = odbWritea.rootAssembly.instances['SCAPULA_%d-1' %(j-1)].nodeSets['TOTAL']
fieldValuesa_Pscap = odbWritea.steps['Step-1'].frames[-1].fieldOutputs['PRESSONLY'].getSubset(region=Pscap).values
import numpy
import math
noda_S = [0] * (len(fieldValuesa_Pscap))
for k in range(len(fieldValuesa_Pscap)):
    noda_S[k] = fieldValuesa_Pscap[k].nodeLabel
scap_data = [0] * (len(fieldValuesa_Pscap))
for n in range(len(fieldValuesa_Pscap)):
    scap_data[k] = fieldValuesa_Pscap[n].data
#max_sd = max(scap_data)
#expo_sd = (log10(max_sd) +2)
nda_S = [0] * (max(noda_S))
Data_S = [0] * (max(noda_S))
Dispdata_S = [0] * (max(noda_S))
k = 11000
for L in range(max(noda_S)):
```

```
        D_S = []
        #Y_Cd_S = 1
        #Y_Cd = ((mdb.models['Model-%d' %count].rootAssembly.instances['Humerus_8_4_cut_nosurfnish-1'].nodes[L].coordinates[1])-21.968)/5 # 5 is derived from uppermost z coordinate of vertex of humerus  minus lowermost z coordinate of vertex and 21.968 is the z coordinate of the lower point
        Cd_S = k *((0.14) -(0.87* 1) +(4.4* (1 ** 2)) - (2.66 * (1 ** 3))) #Y_Cd is assumed 1 for time being
        for m in range(len(fieldValuesa_Pscap)):
            if noda_S[m] == L+1 :
                if math.isnan(fieldValuesa_Pscap[m].data) == False:
                    D_S.append(fieldValuesa_Pscap[m].data)
        Data_S[L] = -1*(numpy.mean(D_S))*Cd_S*mech_scap
        nda_S[L] = L+1
        Dispdata_S[L] = [Data_S[L]]
    max_stress_scap = max(Data_S)
    #max_node_scap = nda[Data_S.index(max_stress_scap)]
    newResultsFielda.addData(position= NODAL , instance = openOdb(r'C:\Nikhil_D\Research\Abaqus\Rat_bone_geometry\Asmbly_65\static_%d.odb' %j).rootAssembly.instances
        ['SCAPULA_%d-1' %(j-1)] , labels = nda_S , data = Dispdata_S)
    odbWritea.save()
    odbWritea.close()
    # Importing the first stress analysis cae model
    mdb.openAuxMdb(pathName=
        'C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_65/M1.cae')
    mdb.copyAuxMdbModel(fromName='Model-%d' %(count+1), toName='Model-%d' %(count+2)) # names of the new model can be changed
    mdb.copyAuxMdbModel(fromName='Model-%d' %(count+1), toName='Model-%d' %(count+2))
    mdb.closeAuxMdb()
```

```
mdb.models['Model-%d' %(count+2)].boundaryConditions['BC-2'].setValues(u1=0.0)
mdb.models['Model-%d' %(count+2)].boundaryConditions['BC-2'].setValues(u2=0.0)
mdb.models['Model-%d' %(count+2)].interactions['Int-1'].suppress()
mdb.models['Model-%d' %(count+2)].Temperature(absoluteExteriorTolerance=0.0,
    beginIncrement=None, beginStep=2, createStepName='Step-1',
    distributionType=FROM_FILE, endIncrement=None, endStep=2,
    exteriorTolerance=0.05, fileName=
    'C:\Nikhil_D\Research\Abaqus\Rat_bone_geometry\Asmbly_65\static_%d.odb' %j,
    interpolate=OFF, name='Predefined Field-1')
# Creating and submitting the job
mdb.Job(atTime=None, contactPrint=OFF, description='', echoPrint=OFF,
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF,
    memory=90, memoryUnits=PERCENTAGE, model='Model-%d' %(count+2), modelPrint=OFF,
    multiprocessingMode=DEFAULT, name='Mechgrowth_%d' %j, nodalOutputPrecision=SINGLE,
    numCpus=8, numDomains=8, numGPUs=0, queue=None, scratch='', type=ANALYSIS,
    userSubroutine='', waitHours=0, waitMinutes=0)
mdb.jobs['Mechgrowth_%d' %j].submit(consistencyChecking=OFF)
mdb.jobs['Mechgrowth_%d' %j].waitForCompletion()
# Importing the first stress analysis cae model
mdb.openAuxMdb(pathName=
    'C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_65/M1.cae')
mdb.copyAuxMdbModel(fromName='Model-%d' %(count+1), toName='Model-%d' %(count+3)) # names of the new model can be changed
mdb.copyAuxMdbModel(fromName='Model-%d' %(count+1), toName='Model-%d' %(count+3))
mdb.closeAuxMdb()
mdb.models['Model-%d' %(count+3)].boundaryConditions['BC-2'].setValues(u1=0.0)
mdb.models['Model-%d' %(count+3)].boundaryConditions['BC-2'].setValues(u2=0.0)
mdb.models['Model-%d' %(count+3)].interactions['Int-1'].suppress()
```

```
mdb.models['Model-%d' %(count+3)].rootAssembly.DatumCsysByThreePoints(coordSysType=
    CARTESIAN, name='scap_cs', origin=(9.156945,27.0467,22.118969), point1=(9.067071,25.469379,21.794357)
    , point2=(9.309071,25.873672,22.668428))
mdb.models['Model-%d' %(count+3)].rootAssembly.DatumCsysByThreePoints(coordSysType=
    CARTESIAN, name='HUm_cs', origin= mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[513].coordinates, point1= mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[520].coordinates,
    point2= mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[240].coordinates)
y_hum = sqrt((((mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[513].coordinates[0]) - (mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[240].coordinates[0])) **2) +
    (((mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[513].coordinates[1]) - (mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[240].coordinates[1]))**2) +
    (((mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[513].coordinates[2]) - (mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[240].coordinates[2]))**2))
mdb.models['Model-%d' %(count+3)].ExpressionField(description='', expression=
    '(11000*(0.14 -(0.87*((-Y/y_hum)**1))+(4.4*((-Y/y_hum)**2))-(2.66*((-Y/y_hum)**3))))'
    , localCsys=mdb.models['Model-%d' %(count+3)].rootAssembly.datums[mdb.models['Model-%d' %(count+3)].rootAssembly.features['HUm_cs'].id], name=
    'AnalyticalField-1')
mdb.models['Model-%d' %(count+3)].Temperature(createStepName='Step-1',
    crossSectionDistribution=CONSTANT_THROUGH_THICKNESS, distributionType=FIELD
    , field='AnalyticalField-1', magnitudes=(bio_hum, ), name='Hum_temp', region=
```

```
    mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].sets['TOTAL'])
mdb.models['Model-%d' %(count+3)].TabularAmplitude(data=((0.0, 0.0), (1.0, 1.0)), name=
'Amp-1', smooth=SOLVER_DEFAULT, timeSpan=STEP)
mdb.models['Model-%d' %(count+3)].predefinedFields['Hum_temp'].setValues(amplitude='Amp-1')
#mdb.models['Model-%d' %(count+3)].ExpressionField(description='', expression=
#   '(11000*(0.14 -(0.87*((-Z/3.0)**1))+(4.4*((-Z/3.0)**2))-(2.66*((-Z/3.0)**3))))'
#   , localCsys=mdb.models['Model-%d' %(count+3)].rootAssembly.datums[mdb.models['Model-%d' %(count+3)].rootAssembly.features['scap_cs'].id], name=
#   'AnalyticalField-2')
#mdb.models['Model-%d' %(count+3)].Temperature(createStepName='Step-1',
#   crossSectionDistribution=CONSTANT_THROUGH_THICKNESS, distributionType=FIELD
#   , field='AnalyticalField-2', magnitudes=(0.02, ), name='Scap_temp', region=
#   mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].sets['TOTAL'])
mdb.models['Model-%d' %(count+3)].Temperature(amplitude='Amp-1', createStepName='Step-1',
    crossSectionDistribution=CONSTANT_THROUGH_THICKNESS, distributionType=
    UNIFORM, magnitudes=(bio_scap, ), name='Scap_temp', region=
    mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].sets['TOTAL'])
mdb.Job(atTime=None, contactPrint=OFF, description='', echoPrint=OFF,
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF,
    memory=90, memoryUnits=PERCENTAGE, model='Model-%d' %(count+3), modelPrint=OFF,
    multiprocessingMode=DEFAULT, name='Biogrowth_%d' %j, nodalOutputPrecision=SINGLE,
    numCpus=8, numDomains=8, numGPUs=0, queue=None, scratch='', type=ANALYSIS,
    userSubroutine='', waitHours=0, waitMinutes=0)
mdb.jobs['Biogrowth_%d' %j].submit(consistencyChecking=OFF)
mdb.jobs['Biogrowth_%d' %j].waitForCompletion()
```

```
mdb.Model(modelType=STANDARD_EXPLICIT, name='Model-%d' %(count+4))
mdb.models['Model-%d' %(count+4)].PartFromOdb(frame=-1, instance='HUMERUS_%d-1' %(j-1), name=
    'Humerus_Mech_%d' %j, odb=session.openOdb(
    r'C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_65/Mechgrowth_%d.odb' %j), shape=
    DEFORMED, step=0)
mdb.models['Model-%d' %(count+4)].PartFromOdb(frame=-1, instance='SCAPULA_%d-1' %(j-1), name=
    'Scapula_Mech_%d' %j, odb=session.openOdb(
    r'C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_65/Mechgrowth_%d.odb' %j), shape=
    DEFORMED, step=0)
mdb.models['Model-%d' %(count+4)].PartFromOdb(instance='HUMERUS_%d-1' %(j-1), name='Humerus_%d' %(j-1),
    odb=session.openOdb(
    r'C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_65/Biogrowth_%d.odb' %j))
mdb.models['Model-%d' %(count+4)].PartFromOdb(instance='SCAPULA_%d-1' %(j-1), name='Scapula_%d' %(j-1),
    odb=session.openOdb(
    r'C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_65/Biogrowth_%d.odb' %j))
mdb.models['Model-%d' %(count+4)].PartFromOdb(frame=-1, instance='HUMERUS_%d-1' %(j-1), name='Humerus_Bio_%d' %j,
    odb=session.openOdb(
    r'C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_65/Biogrowth_%d.odb' %j),
    shape=DEFORMED, step=0)
mdb.models['Model-%d' %(count+4)].PartFromOdb(frame=-1, instance='SCAPULA_%d-1' %(j-1), name='Scapula_Bio_%d' %j,
    odb=session.openOdb(
    r'C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_65/Biogrowth_%d.odb' %j),
```

```
    shape=DEFORMED, step=0)
mdb.models['Model-%d' %(count+4)].Part(name='Humerus_%d' %j, objectToCopy=
    mdb.models['Model-%d' %(count+4)].parts['Humerus_%d' %(j-1)])
mdb.models['Model-%d' %(count+4)].Part(name='Scapula_%d' %j, objectToCopy=
    mdb.models['Model-%d' %(count+4)].parts['Scapula_%d' %(j-1)])
# trial loop
for i in range(len(mdb.models['Model-%d' %(count+4)].parts['Humerus_Mech_%d' %j].nodes)): ### for some weird reason +1 needs to be added only in this case
     mdb.models['Model-%d' %(count+4)].parts['Humerus_%d' %j].editNode(coordinate1=(mdb.models['Model-%d' %(count+4)].parts['Humerus_Mech_%d' %j].nodes[i].coordinates[0]+
        ((mdb.models['Model-%d' %(count+4)].parts['Humerus_Bio_%d' %j].nodes[i].coordinates[0])-
        (mdb.models['Model-%d' %(count+4)].parts['Humerus_%d' %(j-1)].nodes[i].coordinates[0]))),
        coordinate2 = (mdb.models['Model-%d' %(count+4)].parts['Humerus_Mech_%d' %j].nodes[i].coordinates[1]+
        ((mdb.models['Model-%d' %(count+4)].parts['Humerus_Bio_%d' %j].nodes[i].coordinates[1])-
        (mdb.models['Model-%d' %(count+4)].parts['Humerus_%d' %(j-1)].nodes[i].coordinates[1]))),
        coordinate3 = (mdb.models['Model-%d' %(count+4)].parts['Humerus_Mech_%d' %j].nodes[i].coordinates[2]+
        ((mdb.models['Model-%d' %(count+4)].parts['Humerus_Bio_%d' %j].nodes[i].coordinates[2])-
        (mdb.models['Model-%d' %(count+4)].parts['Humerus_%d' %(j-1)].nodes[i].coordinates[2]))),
        nodes=mdb.models['Model-%d' %(count+4)].parts['Humerus_%d' %j].nodes[i:(i+1)])
for i in range(len(mdb.models['Model-%d' %(count+4)].parts['Scapula_Mech_%d' %j].nodes)): ### check for the instances in the odb file and add the proper instance coordinate
```

```
        mdb.models['Model-%d' %(count+4)].parts['Scapula_%d' %j].editNode(coordinate1=(mdb.models['Model-%d' %(count+4)].parts['Scapula_Mech_%d' %j].nodes[i].coordinates[0]+
        ((mdb.models['Model-%d' %(count+4)].parts['Scapula_Bio_%d' %j].nodes[i].coordinates[0])-
        (mdb.models['Model-%d' %(count+4)].parts['Scapula_%d' %(j-1)].nodes[i].coordinates[0]))),
        coordinate2 = (mdb.models['Model-%d' %(count+4)].parts['Scapula_Mech_%d' %j].nodes[i].coordinates[1]+
        ((mdb.models['Model-%d' %(count+4)].parts['Scapula_Bio_%d' %j].nodes[i].coordinates[1])-
        (mdb.models['Model-%d' %(count+4)].parts['Scapula_%d' %(j-1)].nodes[i].coordinates[1]))),
        coordinate3 = (mdb.models['Model-%d' %(count+4)].parts['Scapula_Mech_%d' %j].nodes[i].coordinates[2]+
        ((mdb.models['Model-%d' %(count+4)].parts['Scapula_Bio_%d' %j].nodes[i].coordinates[2])-
        (mdb.models['Model-%d' %(count+4)].parts['Scapula_%d' %(j-1)].nodes[i].coordinates[2]))),
        nodes=mdb.models['Model-%d' %(count+4)].parts['Scapula_%d' %j].nodes[i:(i+1)])
count = count+4
mdb.Model(name='Model-%d' %(count+1), objectToCopy=mdb.models['Model-%d' %count])
#mdb.Model(name='Model-%d' %(count+2), objectToCopy=mdb.models['Model-%d' %count])
mdb.models['Model-%d' %count].Material(name='Material-1')
mdb.models['Model-%d' %count].materials['Material-1'].Density(table=((0.007, ), ))
mdb.models['Model-%d' %count].materials['Material-1'].Elastic(table=((1.1, 0.45), ))
mdb.models['Model-%d' %count].materials['Material-1'].Expansion(table=((8e-06, ), ))
mdb.models['Model-%d' %count].HomogeneousSolidSection(material='Material-1', name=
    'Section-1', thickness=None)
mdb.models['Model-%d' %count].parts['Humerus_%d' %j].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=
    mdb.models['Model-%d' %count].parts['Humerus_%d' %j].sets['TOTAL'], sectionName=
    'Section-1', thicknessAssignment=FROM_SECTION)
mdb.models['Model-%d' %count].parts['Scapula_%d' %j].SectionAssignment(offset=0.0,
```

```
    offsetField='', offsetType=MIDDLE_SURFACE, region=
    mdb.models['Model-%d' %count].parts['Scapula_%d' %j].sets['TOTAL'], sectionName=
    'Section-1', thicknessAssignment=FROM_SECTION)
mdb.models['Model-%d' %count].rootAssembly.Instance(dependent=ON, name='Humerus_%d-1' %j,
    part=mdb.models['Model-%d' %count].parts['Humerus_%d' %j])
mdb.models['Model-%d' %count].rootAssembly.Instance(dependent=ON, name='Scapula_%d-1' %j,
    part=mdb.models['Model-%d' %count].parts['Scapula_%d' %j])
#mdb.models['Model-%d' %count].rootAssembly.translate(instanceList=('Humerus_%d-1' %j, ), vector=((0.0), (0.0),(j+0.5)))
#mdb.models['Model-%d' %count].rootAssembly.translate(instanceList=('Humerus_%d-1' %j, ), vector=((0.5), (0.5),0.0))
mdb.save()
mdb.saveAs(pathName='C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_65/M1_%d.cae' %j)
mdb.saveAs(pathName='C:/Nikhil_D/Research/Abaqus/Rat_bone_geometry/Asmbly_65/M1.cae')
```

Appendix G: I2M2 matlab script

This appendix includes a matlab script used for I2M2 simulations. This script performs musculoskeletal simulations in OpenSIm. Depending upon the preganglionic or postganglionic simulations, the appropriate muscle mass and length parameters were implemented. This script calls for one other python script used to perform finite element simulations.

Matlab script:

```
clear, close all, clc;
incr = 1
for ii = 1:15

    %Variables
    GridSpaceX=1;
    GridSpaceY=1;
    %mo='noGUI';
    mo='script';

    %Make python file with variables
    delete('Var.py');
    fid = fopen('Var.py', 'w');
    fprintf(fid,'GridSpaceX = %0.12f\n',GridSpaceX);
    fprintf(fid,'GridSpaceY = %0.12f\n',GridSpaceY);
    fclose(fid);

    %Make part(run Abaqus)
    %unix(['abaqus cae ',mo,'=Abq_mat_conn.py']);   %Unix system
    status = system(['abaqus cae ',mo,'=Script_abaqus.py']); %Windows system?



    % Scale,Forward dynamic simulation and joint reaction force analysis
script
    import org.opensim.modeling.*

    fileID=fopen('scale.txt');
    scale_data=textscan(fileID, '%s %s %f32');
    scale_data=scale_data(1,3);
    scale_data=scale_data{1};
    row=1
    for j=1:max(size(scale_data))
        col=mod(j,3);
        if mod(j,3)==0
            col=3;
        end
```

```
        coord(row,col)=scale_data(j);
        if mod(j,3)==0
            row=row+1;
        end
    end
    fclose(fileID);


    % %Open Model
    model=Model('MoBL_ARMS.osim');
    % mus=model.getMuscles();
    % Post Mass
    %
mass_muscles={'DELT1','DELT2','DELT3','PECM1','PECM2','PECM3','BIClong','SUBS
C','TMAJ','TRIlong'};
    % mass_factors={(1-((1-0.62)*(ii/15))),(1-((1-0.6025)*(ii/15))),(1-((1-
(0.585/0.854))*(ii/15))),(1-((1-(0.747/0.889))*(ii/15))),(1-((1-
(0.747/0.889))*(ii/15))),(1-((1-(0.747/0.889))*(ii/15))),(1-((1-
0.457)*(ii/15))),(1-((1-(0.76/0.91))*(ii/15))),(1-((1-0.631)*(ii/15))),(1-
((1-0.5788)*(ii/15)))};
    % Pre Mass

mass_muscles={'DELT1','DELT2','DELT3','BIClong','SUBSC','TMAJ','TRIlong','SUP
SP','INFSP'};
      mass_factors={(1-((1-0.294)*(ii/15))),(1-((1-0.324)*(ii/15))),(1-((1-
(0.354/0.777))*(ii/15))),(1-((1-(0.242/0.77))*(ii/15))),(1-((1-
0.301)*(ii/15))),(1-((1-(0.455/0.707))*(ii/15))),(1-((1-0.7648)*(ii/15))),(1-
((1-(0.164/0.675))*(ii/15))),(1-((1-0.212)*(ii/15)))};
    % Post Length
    %  length_muscles={'DELT3','PECM1','PECM2','PECM3','SUBSC','BIClong'};
    %  length_factors={(1-((1-0.8543)*(ii/15))),(1-((1-0.8887)*(ii/15))),(1-
((1-0.8887)*(ii/15))),(1-((1-0.8887)*(ii/15))),(1-((1-0.9103)*(ii/15))),(1-
((1-1.057)*(ii/15)))};
    % Pre Length
      length_muscles={'DELT3','BIClong','BICshort','SUPSP','TMAJ'};
      length_factors={(1-((1-0.777)*(ii/15))),(1-((1-0.77)*(ii/15))),(1-((1-
0.8016)*(ii/15))),(1-((1-0.6755)*(ii/15))),(1-((1-0.707)*(ii/15)))};

    %Get Original optimal fiber length for previous simulations
    % for i=0:model.getMuscles.getSize()-1
    %     for j=1:length(noscale)
    %         if strcmp(mus.get(i).getName(),noscale(j))==1
    %
origFibLeng.(noscale{j}).fiberlength=model.getMuscles().get(i).getOptimalFibe
rLength();
    %             break
    %         end
    %     end
    % end
    % Scale model


    BS=model.getBodySet();
    ST=ScaleTool();
    ST.setName('scaling_operation');
    ST.setSubjectMass(4.8);
```

```
MS=ST.getModelScaler();
MS.setOutputModelFileName('scaled_model.osim');

AS=ArrayStr();
AS.setSize(1);
AS.setitem(0,'manualScale');
MS.setScalingOrder(AS);

SZ=BS.getSize();
SS=ScaleSet();
SS.setSize(SZ);
vec=Vec3();
vec.set(0,coord(1,1));
vec.set(1,coord(1,2));
vec.set(2,coord(1,3));
vec2=Vec3();
vec2.set(0,coord(2,1));
vec2.set(1,coord(2,2));
vec2.set(2,coord(2,3));



for i=1:SZ
  name=BS.get(i-1).getName();
  S(i)=Scale();
  S(i).setSegmentName(name);
    if i<7
    S(i).setScaleFactors(vec);
    else
    S(i).setScaleFactors(vec2);
    end

    SS.cloneAndAppend(S(i));
%   MS.addScale(S(i));
end
SS.print('scale_set.xml');

MS.setScaleSetFile('scale_set.xml');


% ST.print('scale.xml')

ST.getModelScaler().setOutputModelFileName('scaled_model.osim');

ST.getModelScaler().processModel(model.initSystem(),model,ST.getPathToSubject
(),ST.getSubjectMass());

% Set model for Forward Dynamics

modelScaled=Model('scaled_model.osim');

muscle=modelScaled.getMuscles();
```

```
    % Increasing peak force
     for i=0:muscle.getSize()-1

                 MAX_ISO=muscle.get(i).getMaxIsometricForce()
                 muscle.get(i).setMaxIsometricForce(MAX_ISO*(((vec.get(1))+
(vec.get(2))+ (vec.get(3))/3)));
                 muscle.get(i).getMaxIsometricForce()

     end



    % Mass scaling
      for i=0:muscle.getSize()-1
          for j=1:length(mass_muscles)
              if strcmp(muscle.get(i).getName(),mass_muscles(j))==1
                 MAX_ISO=muscle.get(i).getMaxIsometricForce()
                 muscle.get(i).setMaxIsometricForce(MAX_ISO*mass_factors{j});
                 muscle.get(i).getMaxIsometricForce()
                 break
              end
          end
      end


    % Length scaling
     for i=0:muscle.getSize()-1
         for j=1:length(length_muscles)
             if strcmp(muscle.get(i).getName(),length_muscles(j))==1
                OPT_LEN=muscle.get(i).getMaxIsometricForce()

muscle.get(i).setMaxIsometricForce(OPT_LEN*length_factors{j});
                muscle.get(i).getMaxIsometricForce()
                break
             end
         end
     end

    modelScaled.print('scaled_model.osim')


    modelScaled=Model('scaled_model.osim');
    modelScaled.initSystem();
    % % Add controller
    muscleController=PrescribedController('static.sto');
    modelScaled.addController(muscleController);


    % Run Forward dynamics
    FT=ForwardTool();
    FT.setModel(modelScaled);
    FT.setStartTime(0);
    FT.setFinalTime(1);
    FT.setOutputPrecision(20);
```

```
    FT.setMaximumNumberOfSteps(20000);
    FT.setErrorTolerance(0.00001);
    FT.setControlsFileName('static.sto');
    FT.setStatesFileName('init_states.sto')
    FT.setSolveForEquilibrium(true);
    FT.setName('static_sim');
    FT.print('FD_setup.xml');
    FT.run();

    % % Run JRF analysis
    analyzeTool=AnalyzeTool('jrf_setup.xml');
    analyzeTool.setModel(modelScaled);
    analyzeTool.setName('JRF')
    analyzeTool.run();


    data = Storage('JRF_JointReaction_ReactionLoads.sto');
    labels = data.getColumnLabels();
    X = ArrayDouble();
    Y = ArrayDouble();
    Z = ArrayDouble();
    rot=[cosd(30) 0 -sind(30);0 1 0;sind(30) 0 cosd(30)];
    data.getDataColumn(labels.getitem(1),X);
    data.getDataColumn(labels.getitem(2),Y);
    data.getDataColumn(labels.getitem(3),Z);

    glenoid=zeros(X.getSize(),3);
    for i=1:X.getSize()
        vecf=[X.get(i-1);Y.get(i-1);Z.get(i-1)];
        vecf=rot*vecf;
        glenoid(i,:)=transpose(vecf);
    end


    xSum=0;
    ySum=0;
    zSum=0;
    SZ1=X.getSize();
    for i=1:X.getSize()-1
        xSum=xSum+glenoid(i,1);
        ySum=ySum+glenoid(i,2);
        zSum=zSum+glenoid(i,3);
    end

    xAVG=xSum/X.getSize();
    yAVG=ySum/Y.getSize();
    zAVG=zSum/Z.getSize();

    avg_jrf=[xAVG, yAVG, zAVG];
    fileID=fopen(sprintf('JRF%i.py',incr),'w');
    fprintf(fileID, 'JRF_x = %5d\nJRF_y = %5d\nJRF_z = %5d\n', avg_jrf);
    fclose(fileID);
    incr = incr +1;
end
```

Appendix H: I2M2 python script

This appendix includes the python script used for I2M2 simulations. The python script performs the bone morphological simulations in Abaqus following preganglionic and postganglionic injuries discussed in Aim 4.

Python script:

```
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from optimization import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *

#execfile('JRF.py')
execfile('count_data.py')
openMdb(pathName='C:/Nikhil_D/Research/Abaqus/I2M2/postgang_trial_22/M1')
mdb.saveAs(pathName='C:\Nikhil_D\Research\Abaqus\I2M2\postgang_trial_22\M1')

#increased mech growth compared to AMC 12 and 13
mech_hum = 20
```

```
mech_scap = 20
bio_hum = 9350
bio_scap = 9350

for j in range( cyc ,(cyc+1)):
    execfile('JRF%d.py'%(cyc-1))##Note: Scouting operation for explicit dynamic analysis to determine the nodes on the two geometries which comes in the contact for the first time
    #if (j-1)%3 == 0:
    #   zmove = 0.1
    #else:
    #   zmove = 0.0
    mdb.openAuxMdb(pathName=
        'C:/Nikhil_D/Research/Abaqus/I2M2/postgang_trial_22/M1.cae')
    mdb.copyAuxMdbModel(fromName='Model-%d' %count, toName='Model-%d' %(count+1)) # names of the new model can be changed
    mdb.copyAuxMdbModel(fromName='Model-%d' %count, toName='Model-%d' %(count+1))
    mdb.closeAuxMdb()
    mdb.models['Model-%d' %count].Material(name='Material-1')
    mdb.models['Model-%d' %count].materials['Material-1'].Density(table=((0.007, ), ))
    mdb.models['Model-%d' %count].materials['Material-1'].Elastic(table=((1.1, 0.45), ))
    mdb.models['Model-%d' %count].materials['Material-1'].Expansion(table=((8e-06, ), ))
    mdb.models['Model-%d' %count].HomogeneousSolidSection(material='Material-1', name=
        'Section-1', thickness=None)
    mdb.models['Model-%d' %count].parts['Humerus_%d' %(j-1)].SectionAssignment(offset=0.0,
        offsetField='', offsetType=MIDDLE_SURFACE, region=
        mdb.models['Model-%d' %count].parts['Humerus_%d' %(j-1)].sets['TOTAL'], sectionName=
        'Section-1', thicknessAssignment=FROM_SECTION)
```

```
mdb.models['Model-%d' %count].parts['Scapula_%d' %(j-1)].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=
    mdb.models['Model-%d' %count].parts['Scapula_%d' %(j-1)].sets['TOTAL'], sectionName=
    'Section-1', thicknessAssignment=FROM_SECTION)
mdb.models['Model-%d' %count].rootAssembly.translate(instanceList=('Humerus_%d-1' %(j-
1), ), vector=(-1.2, (0.0),0.0))
if j%1 == 0:
    mdb.models['Model-%d' %count].rootAssembly.translate(instanceList=('Humerus_%d-1' %(j-
1), ), vector=(-0.0, (-0.01*cyc),(-0.028*cyc)))
mdb.models['Model-%d' %count].ExplicitDynamicsStep(name='Step-
1', previous='Initial',timePeriod=10.0)
mdb.models['Model-%d' %count].OperatorFilter(halt=ON, limit=0.01, name='Filter-1',
    operation=MAX)
mdb.models['Model-%d' %count].fieldOutputRequests['F-Output-1'].setValues(filter=
    'Filter-1', variables=('S', ))
mdb.models['Model-%d' %count].ContactProperty('IntProp-1')
mdb.models['Model-%d' %count].interactionProperties['IntProp-1'].TangentialBehavior(
    formulation=FRICTIONLESS)
mdb.models['Model-%d' %count].interactionProperties['IntProp-1'].NormalBehavior(
    allowSeparation=ON, constraintEnforcementMethod=DEFAULT,
    pressureOverclosure=HARD)
mdb.models['Model-%d' %count].SurfaceToSurfaceContactExp(clearanceRegion=None,
    createStepName='Step-1', datumAxis=None, initialClearance=OMIT,
    interactionProperty='IntProp-1', master=
    mdb.models['Model-%d' %count].rootAssembly.instances['Scapula_%d-1' %(j-
1)].surfaces['CONT_SCAP']
    , mechanicalConstraint=KINEMATIC, name='Int-1', slave=
```

```
    mdb.models['Model-%d' %count].rootAssembly.instances['Humerus_%d-1' %(j-1)].surfaces['CONT_HUM']
    , sliding=FINITE)
mdb.models['Model-%d' %count].BodyForce(comp1=0.1, comp2=0.0, createStepName='Step-1', name=
    'Load-1', region=
    mdb.models['Model-%d' %count].rootAssembly.instances['Humerus_%d-1' %(j-1)].sets['TOTAL'])
mdb.models['Model-%d' %count].EncastreBC(createStepName='Initial', localCsys=None,
    name='BC-1', region=
    mdb.models['Model-%d' %count].rootAssembly.instances['Scapula_%d-1' %(j-1)].sets['FIX'])
mdb.models['Model-%d' %count].DisplacementBC(amplitude=UNSET, createStepName='Step-1',
    distributionType=UNIFORM, fieldName='', fixed=OFF, localCsys=None, name=
    'BC-2', region=
    mdb.models['Model-%d' %count].rootAssembly.instances['Humerus_%d-1' %(j-1)].sets['DISP'],
    u1=UNSET, u2=0.0, u3=0.0, ur1=0.0, ur2=0.0, ur3=0.0)
pos = 0 ##Note: Following commands are to request the output information on the nodes and not on gaussian points (elements)
pos_m = [0] * 1
n = 0
import string
mdb.models['Model-%d' %count].keywordBlock.synchVersions()
for block in mdb.models['Model-%d' %count].keywordBlock.sieBlocks:
    if string.lower(block[0:len('*Output, field')])==string.lower('*Output, field'):
        pos_m[n] = pos
        n = n+1
    pos = pos +1
mdb.models['Model-%d' %count].keywordBlock.replace(pos_m[0] + 1, """
*Element Output,POSITION= NODES, directions=YES
```

```
S,
""")
mdb.Job(activateLoadBalancing=False, atTime=None, contactPrint=OFF,
    description='', echoPrint=OFF, explicitPrecision=SINGLE, historyPrint=OFF,
    memory=90, memoryUnits=PERCENTAGE, model='Model-%d' %count, modelPrint=OFF,
    multiprocessingMode=DEFAULT, name='dyn_%d' %j, nodalOutputPrecision=SINGLE,
    numCpus=1, numDomains=1, parallelizationMethodExplicit=DOMAIN, queue=None,
    scratch='', type=ANALYSIS, userSubroutine='', waitHours=0, waitMinutes=0)
mdb.jobs['dyn_%d' %j].submit(consistencyChecking=OFF)
mdb.jobs['dyn_%d' %j].waitForCompletion()
odbWritea=openOdb(path="C:\Nikhil_D\Research\Abaqus\I2M2\postgang_trial_22\dyn_%d.odb" %j)
## Note: Deterining the max values of stress on the nodes
Phum = odbWritea.rootAssembly.instances['HUMERUS_%d-1' %(j-1)].nodeSets['TOTAL']
fieldValuesa_Phum = odbWritea.steps['Step-1'].frames[-1].fieldOutputs['S_FILTER-1'].getSubset(region=Phum).values
noda = [0] * (len(fieldValuesa_Phum))
for k in range(len(fieldValuesa_Phum)):
    noda[k] = fieldValuesa_Phum[k].nodeLabel
hum_data = [0] * (len(fieldValuesa_Phum))
for n in range(len(fieldValuesa_Phum)):
    hum_data[n] = fieldValuesa_Phum[n].data[1]
min_hd = min(hum_data)
min_index_hd = hum_data.index(min_hd)
min_nod = noda[min_index_hd]
Pscap = odbWritea.rootAssembly.instances['SCAPULA_%d-1' %(j-1)].nodeSets['TOTAL']
fieldValuesa_scap = odbWritea.steps['Step-1'].frames[-1].fieldOutputs['S_FILTER-1'].getSubset(region=Pscap).values
noda = [0] * (len(fieldValuesa_scap))
```

```
for k in range(len(fieldValuesa_scap)):
    noda[k] = fieldValuesa_scap[k].nodeLabel
scap_data = [0] * (len(fieldValuesa_scap))
for n in range(len(fieldValuesa_scap)):
    scap_data[n] = fieldValuesa_scap[n].data[1]
min_sc = min(scap_data)
min_index_sc = scap_data.index(min_sc)
min_nod_sc = noda[min_index_sc]
odbWritea.save()
odbWritea.close()
mdb.models['Model-%d' %(count+1)].Material(name='Material-1') ## Note: Model 2 for the implicit simulations
mdb.models['Model-%d' %(count+1)].materials['Material-1'].Density(table=((0.007, ), ))
mdb.models['Model-%d' %(count+1)].materials['Material-1'].Elastic(table=((1.1, 0.45), ))
mdb.models['Model-%d' %(count+1)].materials['Material-1'].Expansion(table=((8e-06, ), ))
mdb.models['Model-%d' %(count+1)].HomogeneousSolidSection(material='Material-1', name=
    'Section-1', thickness=None)
mdb.models['Model-%d' %(count+1)].parts['Humerus_%d' %(j-1)].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=
    mdb.models['Model-%d' %(count+1)].parts['Humerus_%d' %(j-1)].sets['TOTAL'], sectionName=
    'Section-1', thicknessAssignment=FROM_SECTION)
mdb.models['Model-%d' %(count+1)].parts['Scapula_%d' %(j-1)].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=
    mdb.models['Model-%d' %(count+1)].parts['Scapula_%d' %(j-1)].sets['TOTAL'], sectionName=
    'Section-1', thicknessAssignment=FROM_SECTION)
mdb.models['Model-%d' %(count+1)].rootAssembly.DatumCsysByDefault(CARTESIAN)
mdb.models['Model-%d' %(count+1)].rootAssembly.translate(instanceList=('Humerus_%d-1' %(j-1), ), vector=(-(j+1.25), (0.0), 0.0))
```

```
#mdb.models['Model-%d' %(count+1)].rootAssembly.translate(instanceList=('Humerus_%d-1' %(j-1), ), vector=(-(0.5), (-(0.5)), 0.0))
if j%1 == 0:
    mdb.models['Model-%d' %(count+1)].rootAssembly.translate(instanceList=('Humerus_%d-1' %(j-1), ), vector=(-0.0, (-0.01*cyc),(-0.028*cyc)))
dist_x = ((mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[min_nod_sc-1].coordinates[0]) - (mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[min_nod-1].coordinates[0]))
dist_y = ((mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[min_nod_sc-1].coordinates[1]) - (mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[min_nod-1].coordinates[1]))
#mdb.models['Model-%d' %(count+1)].rootAssembly.translate(instanceList=('Humerus_%d-1' %(j-1), ), vector=(-(dist_x*2), -(dist_y*2), 0))
mdb.models['Model-%d' %(count+1)].StaticStep(initialInc=1.0, maxNumInc=1000,
    name='Step-1', nlgeom=ON, previous='Initial')
mdb.models['Model-%d' %(count+1)].fieldOutputRequests['F-Output-1'].setValues(variables=(
    'PRESSONLY', 'U'))
mdb.models['Model-%d' %(count+1)].ContactProperty('IntProp-1')
mdb.models['Model-%d' %(count+1)].interactionProperties['IntProp-1'].TangentialBehavior(
    formulation=FRICTIONLESS)
mdb.models['Model-%d' %(count+1)].interactionProperties['IntProp-1'].NormalBehavior(
    allowSeparation=ON, constraintEnforcementMethod=DEFAULT,
    pressureOverclosure=HARD)
mdb.models['Model-%d' %(count+1)].SurfaceToSurfaceContactStd(adjustMethod=NONE,
    clearanceRegion=None, createStepName='Step-1', datumAxis=None,
    initialClearance=OMIT, interactionProperty='IntProp-1', master=
    mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Scapula_%d-1' %(j-1)].surfaces['CONT_SCAP']
```

```
    , name='Int-1', slave=
    mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Humerus_%d-1' %(j-1)].surfaces['CONT_HUM']
    , sliding=FINITE, thickness=ON)
mdb.models['Model-%d' %(count+1)].EncastreBC(createStepName='Initial', localCsys=None,
    name='BC-1', region=
    mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Scapula_%d-1' %(j-1)].sets['FIX'])
mdb.models['Model-%d' %(count+1)].DisplacementBC(amplitude=UNSET, createStepName='Step-1',
    distributionType=UNIFORM, fieldName='', fixed=OFF, localCsys=None, name=
    'BC-2', region=
    mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Humerus_%d-1' %(j-1)].sets['DISP'],
    u1=((dist_x) + (-JRF_z*0.00011/0.7)), u2=(-JRF_x*0.00011/0.7), u3=( JRF_y*0.00011/0.7), ur1=0.0, ur2=0.0, ur3=0.0) #from muscle scaling opensim simulation a compressive force of 235 N was found. Joint stiffness of 0.0005N/mm was assumed for the simulation.
pos = 0
pos_m = [0] * 1
n = 0
import string
mdb.models['Model-%d' %(count+1)].keywordBlock.synchVersions()
for block in mdb.models['Model-%d' %(count+1)].keywordBlock.sieBlocks:
    if string.lower(block[0:len('*Output, field')])==string.lower('*Output, field'):
        pos_m[n] = pos
        n = n+1
    pos = pos +1
mdb.models['Model-%d' %(count+1)].keywordBlock.replace(pos_m[0] + 2, """
*Element Output,POSITION= NODES, directions=YES
```

```
PRESSONLY
""")
mdb.save()
mdb.Job(atTime=None, contactPrint=OFF, description='', echoPrint=OFF,
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF,
    memory=90, memoryUnits=PERCENTAGE, model='Model-%d' %(count+1), modelPrint=OFF,
    multiprocessingMode=DEFAULT, name='static_%d' %j, nodalOutputPrecision=SINGLE,
    numCpus=8, numDomains=8, numGPUs=0, queue=None, scratch='', type=ANALYSIS,
    userSubroutine='', waitHours=0, waitMinutes=0)
mdb.jobs['static_%d' %j].submit(consistencyChecking=OFF)
mdb.jobs['static_%d' %j].waitForCompletion()
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from optimization import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
from odbAccess import *
from abaqusConstants import *
# Opening the odb file
```

```
odbWritea=openOdb(path="C:\Nikhil_D\Research\Abaqus\I2M2\postgang_trial_22\static_%d.odb" %j, readOnly = False)
#checking the steps in the file
#for stepName in odbWritea.steps.keys():
 #   print ' ', stepName
Phum = odbWritea.rootAssembly.instances['HUMERUS_%d-1' %(j-1)].nodeSets['TOTAL']
Pnodisph = odbWritea.rootAssembly.instances['HUMERUS_%d-1' %(j-1)].nodeSets['NODISPH']
fieldValuesa_Phum = odbWritea.steps['Step-1'].frames[-1].fieldOutputs['PRESSONLY'].getSubset(region=Phum).values
# Creating a new step to write the PRESSONLY output as NODAL TEMPERATURE
newResultsStepa=odbWritea.Step(name='Test-a',description='User Defined Results',domain=TIME, timePeriod=1)
newResultsFramea=newResultsStepa.Frame(incrementNumber=1,frameValue=0.1)
newResultsFielda=newResultsFramea.FieldOutput(name='NT11',description='User Defined Results', type=SCALAR)
# Following is a code take the average value of all the assigned values to one particular node
import numpy
import math
noda = [0] * (len(fieldValuesa_Phum))
for k in range(len(fieldValuesa_Phum)):
    noda[k] = fieldValuesa_Phum[k].nodeLabel
noda_nodisph = [0] * len(Pnodisph.nodes)
for k in range(len(Pnodisph.nodes)):
    noda_nodisph[k] = Pnodisph.nodes[k].label
hum_data = [0] * (len(fieldValuesa_Phum))
for n in range(len(fieldValuesa_Phum)):
    hum_data[k] = fieldValuesa_Phum[n].data
#max_hd = max(hum_data)
```

```
#expo_hd = (log10(max_hd) +2)
nda = [0] * (max(noda))
Data = [0] * (max(noda))
Dispdata = [0] * (max(noda))
k = 11000
for L in range(max(noda)):
    D = []
    Y_Cd = 1 #((mdb.models['Model-%d' %(count+1)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[L].coordinates[1])-21.968)/5 # 5 is derived from uppermost z coordinate of vertex of humerus  minus lowermost z coordinate of vertex and 21.968 is the z coordinate of the lower point
    Cd = k *((0.14) -(0.87* Y_Cd) +(4.4* (Y_Cd ** 2)) - (2.66 * (Y_Cd ** 3)))
    for m in range(len(fieldValuesa_Phum)):
        if noda[m] == L+1 :
            for pp in range(len(Pnodisph.nodes)):
                if noda_nodisph[pp] == noda[m]:
                    mult = 0
                    break
                else:
                    mult = 1
            if math.isnan(fieldValuesa_Phum[m].data) == False:
                D.append(fieldValuesa_Phum[m].data)
    Data[L] = -1*(numpy.mean(D))*Cd*mech_hum*mult
    nda[L] = L+1
    Dispdata[L] = [Data[L]]
max_stress_hum = max(Data)
max_node_hum = nda[Data.index(max_stress_hum)]
## mdb.models['Model-%d' %count].rootAssembly.instances['Part-1-1'].nodes[0].coordinates[1]
```

```
# Compiling the nodal data and assigning the nodal temperature values in the new step
newResultsFielda.addData(position= NODAL , instance = openOdb(r'C:\Nikhil_D\Research\Abaqus\I2M2\postgang_trial_22\static_%d.odb' %j).rootAssembly.instances
    ['HUMERUS_%d-1' %(j-1)] , labels = nda , data = Dispdata)
Pscap = odbWritea.rootAssembly.instances['SCAPULA_%d-1' %(j-1)].nodeSets['TOTAL']
Pnodisps = odbWritea.rootAssembly.instances['SCAPULA_%d-1' %(j-1)].nodeSets['NODISPS']
fieldValuesa_Pscap = odbWritea.steps['Step-1'].frames[-1].fieldOutputs['PRESSONLY'].getSubset(region=Pscap).values
import numpy
import math
noda_S = [0] * (len(fieldValuesa_Pscap))
for k in range(len(fieldValuesa_Pscap)):
    noda_S[k] = fieldValuesa_Pscap[k].nodeLabel
noda_nodisps = [0] * len(Pnodisps.nodes)
for k in range(len(Pnodisps.nodes)):
    noda_nodisps[k] = Pnodisps.nodes[k].label
scap_data = [0] * (len(fieldValuesa_Pscap))
for n in range(len(fieldValuesa_Pscap)):
    scap_data[k] = fieldValuesa_Pscap[n].data
#max_sd = max(scap_data)
#expo_sd = (log10(max_sd) +2)
nda_S = [0] * (max(noda_S))
Data_S = [0] * (max(noda_S))
Dispdata_S = [0] * (max(noda_S))
k = 11000
for L in range(max(noda_S)):
    D_S = []
    #Y_Cd_S = 1
```

```
    #Y_Cd = ((mdb.models['Model-%d' %count].rootAssembly.instances['Humerus_8_4_cut_nosurfnish-1'].nodes[L].coordinates[1])-21.968)/5 # 5 is derived from uppermost z coordinate of vertex of humerus  minus lowermost z coordinate of vertex and 21.968 is the z coordinate of the lower point
    Cd_S = k *((0.14) -(0.87* 1) +(4.4* (1 ** 2)) - (2.66 * (1 ** 3))) #Y_Cd is assumed 1 for time being
    for m in range(len(fieldValuesa_Pscap)):
        if noda_S[m] == L+1 :
            for pp in range(len(Pnodisps.nodes)):
                if noda_nodisps[pp] == noda_S[m]:
                    mult = 0
                    break
                else:
                    mult = 1
            if math.isnan(fieldValuesa_Pscap[m].data) == False:
                D_S.append(fieldValuesa_Pscap[m].data)
    Data_S[L] = -1*(numpy.mean(D_S))*Cd_S*mech_scap*mult
    nda_S[L] = L+1
    Dispdata_S[L] = [Data_S[L]]
max_stress_scap = max(Data_S)
#max_node_scap = nda[Data_S.index(max_stress_scap)]
newResultsFielda.addData(position= NODAL , instance = openOdb(r'C:\Nikhil_D\Research\Abaqus\I2M2\postgang_trial_22\static_%d.odb' %j).rootAssembly.instances
    ['SCAPULA_%d-1' %(j-1)] , labels = nda_S , data = Dispdata_S)
odbWritea.save()
odbWritea.close()
# Importing the first stress analysis cae model
mdb.openAuxMdb(pathName=
    'C:/Nikhil_D/Research/Abaqus/I2M2/postgang_trial_22/M1.cae')
```

```
mdb.copyAuxMdbModel(fromName='Model-%d' %(count+1), toName='Model-%d' %(count+2)) # names of the new model can be changed
mdb.copyAuxMdbModel(fromName='Model-%d' %(count+1), toName='Model-%d' %(count+2))## Note: Mechanical growth simulation
mdb.closeAuxMdb()
mdb.models['Model-%d' %(count+2)].boundaryConditions['BC-2'].setValues(u1=0.0)
mdb.models['Model-%d' %(count+2)].boundaryConditions['BC-2'].setValues(u2=0.0)
mdb.models['Model-%d' %(count+2)].interactions['Int-1'].suppress()
mdb.models['Model-%d' %(count+2)].Temperature(absoluteExteriorTolerance=0.0,
    beginIncrement=None, beginStep=2, createStepName='Step-1',
    distributionType=FROM_FILE, endIncrement=None, endStep=2,
    exteriorTolerance=0.05, fileName=
    'C:\Nikhil_D\Research\Abaqus\I2M2\postgang_trial_22\static_%d.odb' %j,
    interpolate=OFF, name='Predefined Field-1')
# Creating and submitting the job
mdb.Job(atTime=None, contactPrint=OFF, description='', echoPrint=OFF,
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF,
    memory=90, memoryUnits=PERCENTAGE, model='Model-%d' %(count+2), modelPrint=OFF,
    multiprocessingMode=DEFAULT, name='Mechgrowth_%d' %j, nodalOutputPrecision=SINGLE,
    numCpus=8, numDomains=8, numGPUs=0, queue=None, scratch='', type=ANALYSIS,
    userSubroutine='', waitHours=0, waitMinutes=0)
mdb.jobs['Mechgrowth_%d' %j].submit(consistencyChecking=OFF)
mdb.jobs['Mechgrowth_%d' %j].waitForCompletion()
# Importing the first stress analysis cae model
mdb.openAuxMdb(pathName=
    'C:/Nikhil_D/Research/Abaqus/I2M2/postgang_trial_22/M1.cae')
mdb.copyAuxMdbModel(fromName='Model-%d' %(count+1), toName='Model-%d' %(count+3)) # names of the new model can be changed
```

```
mdb.copyAuxMdbModel(fromName='Model-%d' %(count+1), toName='Model-%d' %(count+3))
mdb.closeAuxMdb()
mdb.models['Model-%d' %(count+3)].boundaryConditions['BC-1'].suppress()
mdb.models['Model-%d' %(count+3)].boundaryConditions['BC-2'].suppress()
#mdb.models['Model-%d' %(count+3)].boundaryConditions['BC-2'].setValues(u1=0.0)
#mdb.models['Model-%d' %(count+3)].boundaryConditions['BC-2'].setValues(u2=0.0)
mdb.models['Model-%d' %(count+3)].EncastreBC(createStepName='Initial', localCsys=None,
    name='BC-3', region=
    mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].sets['BIO_SCAP_BC_%d' %j])
mdb.models['Model-%d' %(count+3)].EncastreBC(createStepName='Initial', localCsys=None,
    name='BC-4', region=
    mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].sets['BIO_HUM_BC_%d' %j])
mdb.models['Model-%d' %(count+3)].interactions['Int-1'].suppress()
mdb.models['Model-%d' %(count+3)].rootAssembly.DatumCsysByThreePoints(coordSysType=
    CARTESIAN, name='scap_cs', origin=(9.156945,27.0467,22.118969), point1=(9.067071,25.469379,21.794357)
    , point2=(9.309071,25.873672,22.668428))
mdb.models['Model-%d' %(count+3)].rootAssembly.DatumCsysByThreePoints(coordSysType=
    CARTESIAN, name='HUm_cs', origin= mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[513].coordinates, point1= mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[520].coordinates,
    point2= mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[240].coordinates)
```

```
#y_hum = sqrt((((mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[513].coordinates[0]) - (mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[240].coordinates[0])) **2) +
#   (((mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[513].coordinates[1]) - (mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[240].coordinates[1]))**2) +
#   (((mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[513].coordinates[2]) - (mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[240].coordinates[2]))**2))
#mdb.models['Model-%d' %(count+3)].ExpressionField(description='', expression=
#   '(11000*(0.14 -(0.87*((-Y/y_hum)**1))+(4.4*((-Y/y_hum)**2))-(2.66*((-Y/y_hum)**3))))'
#   , localCsys=mdb.models['Model-%d' %(count+3)].rootAssembly.datums[mdb.models['Model-%d' %(count+3)].rootAssembly.features['HUm_cs'].id], name=
#   'AnalyticalField-1')
#mdb.models['Model-%d' %(count+3)].Temperature(createStepName='Step-1',
#   crossSectionDistribution=CONSTANT_THROUGH_THICKNESS, distributionType=FIELD
#   , field='AnalyticalField-1', magnitudes=(bio_hum, ), name='Hum_temp', region=
#   mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].sets['TOTAL'])
mdb.models['Model-%d' %(count+3)].TabularAmplitude(data=((0.0, 0.0), (1.0, 1.0)), name=
'Amp-1', smooth=SOLVER_DEFAULT, timeSpan=STEP)
#mdb.models['Model-%d' %(count+3)].predefinedFields['Hum_temp'].setValues(amplitude='Amp-1')
#mdb.models['Model-%d' %(count+3)].ExpressionField(description='', expression=
#   '(11000*(0.14 -(0.87*((-Z/3.0)**1))+(4.4*((-Z/3.0)**2))-(2.66*((-Z/3.0)**3))))'
#   , localCsys=mdb.models['Model-%d' %(count+3)].rootAssembly.datums[mdb.models['Model-%d' %(count+3)].rootAssembly.features['scap_cs'].id], name=
#   'AnalyticalField-2')
#mdb.models['Model-%d' %(count+3)].Temperature(createStepName='Step-1',
```

```
#   crossSectionDistribution=CONSTANT_THROUGH_THICKNESS, distributionType=FIELD
#   , field='AnalyticalField-2', magnitudes=(0.02, ), name='Scap_temp', region=
#   mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].sets['TOTAL'])
mdb.models['Model-%d' %(count+3)].Temperature(amplitude='Amp-1', createStepName='Step-1',
    crossSectionDistribution=CONSTANT_THROUGH_THICKNESS, distributionType=
    UNIFORM, magnitudes=(bio_scap, ), name='Scap_temp', region=
    mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].sets['TOTAL'])
mdb.models['Model-%d' %(count+3)].Temperature(amplitude='Amp-1', createStepName='Step-1',
    crossSectionDistribution=CONSTANT_THROUGH_THICKNESS, distributionType=
    UNIFORM, magnitudes=(bio_hum, ), name='Hum_temp', region=
    mdb.models['Model-%d' %(count+3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].sets['TOTAL'])
mdb.Job(atTime=None, contactPrint=OFF, description='', echoPrint=OFF,
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True, historyPrint=OFF,
    memory=90, memoryUnits=PERCENTAGE, model='Model-%d' %(count+3), modelPrint=OFF,
    multiprocessingMode=DEFAULT, name='Biogrowth_%d' %j, nodalOutputPrecision=SINGLE,
    numCpus=8, numDomains=8, numGPUs=0, queue=None, scratch='', type=ANALYSIS,
    userSubroutine='', waitHours=0, waitMinutes=0)
mdb.jobs['Biogrowth_%d' %j].submit(consistencyChecking=OFF)
mdb.jobs['Biogrowth_%d' %j].waitForCompletion()
mdb.Model(modelType=STANDARD_EXPLICIT, name='Model-%d' %(count+4))
mdb.models['Model-%d' %(count+4)].PartFromOdb(frame=-1, instance='HUMERUS_%d-1' %(j-1), name=
    'Humerus_Mech_%d' %j, odb=session.openOdb(
    r'C:/Nikhil_D/Research/Abaqus/I2M2/postgang_trial_22/Mechgrowth_%d.odb' %j), shape=
    DEFORMED, step=0)
```

```
mdb.models['Model-%d' %(count+4)].PartFromOdb(frame=-1, instance='SCAPULA_%d-1' %(j-1), name=
    'Scapula_Mech_%d' %j, odb=session.openOdb(
    r'C:/Nikhil_D/Research/Abaqus/I2M2/postgang_trial_22/Mechgrowth_%d.odb' %j), shape=
    DEFORMED, step=0)
mdb.models['Model-%d' %(count+4)].PartFromOdb(instance='HUMERUS_%d-1' %(j-1), name='Humerus_%d' %(j-1),
    odb=session.openOdb(
    r'C:/Nikhil_D/Research/Abaqus/I2M2/postgang_trial_22/Biogrowth_%d.odb' %j))
mdb.models['Model-%d' %(count+4)].PartFromOdb(instance='SCAPULA_%d-1' %(j-1), name='Scapula_%d' %(j-1),
    odb=session.openOdb(
    r'C:/Nikhil_D/Research/Abaqus/I2M2/postgang_trial_22/Biogrowth_%d.odb' %j))
mdb.models['Model-%d' %(count+4)].PartFromOdb(frame=-1, instance='HUMERUS_%d-1' %(j-1), name='Humerus_Bio_%d' %j,
    odb=session.openOdb(
    r'C:/Nikhil_D/Research/Abaqus/I2M2/postgang_trial_22/Biogrowth_%d.odb' %j),
    shape=DEFORMED, step=0)
mdb.models['Model-%d' %(count+4)].PartFromOdb(frame=-1, instance='SCAPULA_%d-1' %(j-1), name='Scapula_Bio_%d' %j,
    odb=session.openOdb(
    r'C:/Nikhil_D/Research/Abaqus/I2M2/postgang_trial_22/Biogrowth_%d.odb' %j),
    shape=DEFORMED, step=0)
mdb.models['Model-%d' %(count+4)].Part(name='Humerus_%d' %j, objectToCopy=
    mdb.models['Model-%d' %(count+4)].parts['Humerus_%d' %(j-1)])
mdb.models['Model-%d' %(count+4)].Part(name='Scapula_%d' %j, objectToCopy=
    mdb.models['Model-%d' %(count+4)].parts['Scapula_%d' %(j-1)])
# trial loop
```

```
    for i in range(len(mdb.models['Model-%d' %(count+4)].parts['Humerus_Mech_%d' %j].nodes)): ### for some weird reason +1 needs to be added only in this case
        mdb.models['Model-%d' %(count+4)].parts['Humerus_%d' %j].editNode(coordinate1=(mdb.models['Model-%d' %(count+4)].parts['Humerus_Mech_%d' %j].nodes[i].coordinates[0]+
            ((mdb.models['Model-%d' %(count+4)].parts['Humerus_Bio_%d' %j].nodes[i].coordinates[0])-
            (mdb.models['Model-%d' %(count+4)].parts['Humerus_%d' %(j-1)].nodes[i].coordinates[0]))),
            coordinate2 = (mdb.models['Model-%d' %(count+4)].parts['Humerus_Mech_%d' %j].nodes[i].coordinates[1]+
            ((mdb.models['Model-%d' %(count+4)].parts['Humerus_Bio_%d' %j].nodes[i].coordinates[1])-
            (mdb.models['Model-%d' %(count+4)].parts['Humerus_%d' %(j-1)].nodes[i].coordinates[1]))),
            coordinate3 = (mdb.models['Model-%d' %(count+4)].parts['Humerus_Mech_%d' %j].nodes[i].coordinates[2]+
            ((mdb.models['Model-%d' %(count+4)].parts['Humerus_Bio_%d' %j].nodes[i].coordinates[2])-
            (mdb.models['Model-%d' %(count+4)].parts['Humerus_%d' %(j-1)].nodes[i].coordinates[2]))),
            nodes=mdb.models['Model-%d' %(count+4)].parts['Humerus_%d' %j].nodes[i:(i+1)])
    for i in range(len(mdb.models['Model-%d' %(count+4)].parts['Scapula_Mech_%d' %j].nodes)): ### check for the instances in the odb file and add the proper instance coordinate
        mdb.models['Model-%d' %(count+4)].parts['Scapula_%d' %j].editNode(coordinate1=(mdb.models['Model-%d' %(count+4)].parts['Scapula_Mech_%d' %j].nodes[i].coordinates[0]+
            ((mdb.models['Model-%d' %(count+4)].parts['Scapula_Bio_%d' %j].nodes[i].coordinates[0])-
            (mdb.models['Model-%d' %(count+4)].parts['Scapula_%d' %(j-1)].nodes[i].coordinates[0]))),
            coordinate2 = (mdb.models['Model-%d' %(count+4)].parts['Scapula_Mech_%d' %j].nodes[i].coordinates[1]+
            ((mdb.models['Model-%d' %(count+4)].parts['Scapula_Bio_%d' %j].nodes[i].coordinates[1])-
```

```
        (mdb.models['Model-%d' %(count+4)].parts['Scapula_%d' %(j-1)].nodes[i].coordinates[1]))),
        coordinate3 = (mdb.models['Model-%d' %(count+4)].parts['Scapula_Mech_%d' %j].nodes[i].coordinates[2]+
        ((mdb.models['Model-%d' %(count+4)].parts['Scapula_Bio_%d' %j].nodes[i].coordinates[2])-
        (mdb.models['Model-%d' %(count+4)].parts['Scapula_%d' %(j-1)].nodes[i].coordinates[2]))),
        nodes=mdb.models['Model-%d' %(count+4)].parts['Scapula_%d' %j].nodes[i:(i+1)])
count = count+4
cyc = cyc+1
mdb.Model(name='Model-%d' %(count+1), objectToCopy=mdb.models['Model-%d' %count])
#mdb.Model(name='Model-%d' %(count+2), objectToCopy=mdb.models['Model-%d' %count])
mdb.models['Model-%d' %count].Material(name='Material-1')
mdb.models['Model-%d' %count].materials['Material-1'].Density(table=((0.007, ), ))
mdb.models['Model-%d' %count].materials['Material-1'].Elastic(table=((1.1, 0.45), ))
mdb.models['Model-%d' %count].materials['Material-1'].Expansion(table=((8e-06, ), ))
mdb.models['Model-%d' %count].HomogeneousSolidSection(material='Material-1', name=
    'Section-1', thickness=None)
mdb.models['Model-%d' %count].parts['Humerus_%d' %j].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=
    mdb.models['Model-%d' %count].parts['Humerus_%d' %j].sets['TOTAL'], sectionName=
    'Section-1', thicknessAssignment=FROM_SECTION)
mdb.models['Model-%d' %count].parts['Scapula_%d' %j].SectionAssignment(offset=0.0,
    offsetField='', offsetType=MIDDLE_SURFACE, region=
    mdb.models['Model-%d' %count].parts['Scapula_%d' %j].sets['TOTAL'], sectionName=
    'Section-1', thicknessAssignment=FROM_SECTION)
mdb.models['Model-%d' %count].rootAssembly.Instance(dependent=ON, name='Humerus_%d-1' %j,
    part=mdb.models['Model-%d' %count].parts['Humerus_%d' %j])
mdb.models['Model-%d' %count].rootAssembly.Instance(dependent=ON, name='Scapula_%d-1' %j,
    part=mdb.models['Model-%d' %count].parts['Scapula_%d' %j])
```

```
mdb.models['Model-%d' %count].rootAssembly.translate(instanceList=('Humerus_%d-1' %j, ), vector=((j+0.5), 0.0,0.0))
#mdb.models['Model-%d' %count].rootAssembly.translate(instanceList=('Humerus_%d-1' %j, ), vector=((j+0.5)+(JRF_z*0.00011/0.7), (JRF_x*0.00011/0.7),(-JRF_y*0.00011/0.7)))
#mdb.models['Model-%d' %count].rootAssembly.translate(instanceList=('Humerus_%d-1' %j, ), vector=((0.5), (0.5),0.0))
dist_hum_x1_original = sqrt((((mdb.models['Model-2'].rootAssembly.instances['Humerus_0-1'].nodes[443].coordinates[0]) - (mdb.models['Model-2'].rootAssembly.instances['Humerus_0-1'].nodes[343].coordinates[0])) **2) +
    (((mdb.models['Model-2'].rootAssembly.instances['Humerus_0-1'].nodes[443].coordinates[1]) - (mdb.models['Model-2'].rootAssembly.instances['Humerus_0-1'].nodes[343].coordinates[1]))**2) +
    (((mdb.models['Model-2'].rootAssembly.instances['Humerus_0-1'].nodes[443].coordinates[2]) - (mdb.models['Model-2'].rootAssembly.instances['Humerus_0-1'].nodes[343].coordinates[2]))**2))
dist_hum_x1_new = sqrt((((mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[443].coordinates[0]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[343].coordinates[0])) **2) +
    (((mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[443].coordinates[1]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[343].coordinates[1]))**2) +
    (((mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[443].coordinates[2]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[343].coordinates[2]))**2))
#dist_hum_x1_new = ((mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[1].coordinates[0]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[54].coordinates[0]))
```

```
dist_hum_x1 = sqrt((((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[443].coordinates[0]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[343].coordinates[0])) **2) +
    (((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[443].coordinates[1]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[343].coordinates[1]))**2) +
    (((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[443].coordinates[2]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[343].coordinates[2]))**2))
#dist_hum_x1 = ((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %j].nodes[1].coordinates[0]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %j].nodes[54].coordinates[0]))
dist_hum_y1_original = sqrt((((mdb.models['Model-2'].rootAssembly.instances['Humerus_0-1'].nodes[243].coordinates[0]) - (mdb.models['Model-2'].rootAssembly.instances['Humerus_0-1'].nodes[518].coordinates[0])) **2) +
    (((mdb.models['Model-2'].rootAssembly.instances['Humerus_0-1'].nodes[243].coordinates[1]) - (mdb.models['Model-2'].rootAssembly.instances['Humerus_0-1'].nodes[518].coordinates[1]))**2) +
    (((mdb.models['Model-2'].rootAssembly.instances['Humerus_0-1'].nodes[243].coordinates[2]) - (mdb.models['Model-2'].rootAssembly.instances['Humerus_0-1'].nodes[518].coordinates[2]))**2))
dist_hum_y1_new = sqrt((((mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[243].coordinates[0]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[518].coordinates[0])) **2) +
    (((mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[243].coordinates[1]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[518].coordinates[1]))**2) +
```

```python
    (((mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[243].coordinates[2]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[518].coordinates[2]))**2))
#dist_hum_y1_new = ((mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[414].coordinates[1]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[134].coordinates[1]))
dist_hum_y1 = sqrt((((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[243].coordinates[0]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[518].coordinates[0])) **2) +
    (((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[243].coordinates[1]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[518].coordinates[1]))**2) +
    (((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[243].coordinates[2]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[518].coordinates[2]))**2))
#dist_hum_y1 = ((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %j].nodes[414].coordinates[1]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %j].nodes[134].coordinates[1]))
dist_hum_z1_original = sqrt((((mdb.models['Model-2'].rootAssembly.instances['Humerus_0-1'].nodes[365].coordinates[0]) - (mdb.models['Model-2'].rootAssembly.instances['Humerus_0-1'].nodes[23].coordinates[0])) **2) +
    (((mdb.models['Model-2'].rootAssembly.instances['Humerus_0-1'].nodes[365].coordinates[1]) - (mdb.models['Model-2'].rootAssembly.instances['Humerus_0-1'].nodes[23].coordinates[1]))**2) +
    (((mdb.models['Model-2'].rootAssembly.instances['Humerus_0-1'].nodes[365].coordinates[2]) - (mdb.models['Model-2'].rootAssembly.instances['Humerus_0-1'].nodes[23].coordinates[2]))**2))
```

```
dist_hum_z1_new = sqrt((((mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[365].coordinates[0]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[23].coordinates[0])) **2) +
    (((mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[365].coordinates[1]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[23].coordinates[1]))**2) +
    (((mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[365].coordinates[2]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[23].coordinates[2]))**2))
#dist_hum_z1_new = ((mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[46].coordinates[2]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Humerus_%d-1' %j].nodes[61].coordinates[2]))
dist_hum_z1 = sqrt((((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[365].coordinates[0]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[23].coordinates[0])) **2) +
    (((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[365].coordinates[1]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[23].coordinates[1]))**2) +
    (((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[365].coordinates[2]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[23].coordinates[2]))**2))
#dist_hum_z1 = ((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[46].coordinates[2]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[61].coordinates[2]))
#dist_hum_z2 = ((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[27].coordinates[2]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[100].coordinates[2]))
```

```
#dist_hum_z3 = ((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[29].coordinates[2]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Humerus_%d-1' %(j-1)].nodes[98].coordinates[2]))
#avg_hum_x = (dist_hum_x1 + dist_hum_x2 + dist_hum_x3)/3
#avg_hum_y = (dist_hum_y1 + dist_hum_y2 + dist_hum_y3)/3
#avg_hum_z = (dist_hum_z1 + dist_hum_z2 + dist_hum_z3)/3
#avg_hum_x_new = (dist_hum_x1_new + dist_hum_x2_new + dist_hum_x3_new)/3
#avg_hum_y_new = (dist_hum_y1_new + dist_hum_y2_new + dist_hum_y3_new)/3
#avg_hum_z_new = (dist_hum_z1_new + dist_hum_z2_new + dist_hum_z3_new)/3
#scale_hum_x = (avg_hum_x_new/avg_hum_x)
#scale_hum_y = (avg_hum_y_new/avg_hum_y)
#scale_hum_z = (avg_hum_z_new/avg_hum_z)
scale_hum_z = (dist_hum_z1_new/dist_hum_z1)
scale_hum_y = (dist_hum_y1_new/dist_hum_y1)
scale_hum_x = (dist_hum_x1_new/dist_hum_x1)
scale_hum_tot_z = (dist_hum_z1_new/dist_hum_z1_original)
scale_hum_tot_y = (dist_hum_y1_new/dist_hum_y1_original)
scale_hum_tot_x = (dist_hum_x1_new/dist_hum_x1_original)
#dist_scap_x1_new = ((mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[4].coordinates[0]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[25].coordinates[0]))
#dist_scap_x2_new = ((mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[59].coordinates[0]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[264].coordinates[0]))
#dist_scap_x3_new = ((mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[8].coordinates[0]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[47].coordinates[0]))
```

```
#dist_scap_x1 = ((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[4].coordinates[0]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[25].coordinates[0]))
#dist_scap_x2 = ((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[59].coordinates[0]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[264].coordinates[0]))
#dist_scap_x3 = ((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[8].coordinates[0]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[47].coordinates[0]))
dist_scap_x1_original = sqrt((((mdb.models['Model-2'].rootAssembly.instances['Scapula_0-1'].nodes[661].coordinates[0]) - (mdb.models['Model-2'].rootAssembly.instances['Scapula_0-1'].nodes[537].coordinates[0])) **2) +
    (((mdb.models['Model-2'].rootAssembly.instances['Scapula_0-1'].nodes[661].coordinates[1]) - (mdb.models['Model-2'].rootAssembly.instances['Scapula_0-1'].nodes[537].coordinates[1]))**2) +
    (((mdb.models['Model-2'].rootAssembly.instances['Scapula_0-1'].nodes[661].coordinates[2]) - (mdb.models['Model-2'].rootAssembly.instances['Scapula_0-1'].nodes[537].coordinates[2]))**2))
dist_scap_x1_new = sqrt((((mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[661].coordinates[0]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[537].coordinates[0])) **2) +
    (((mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[661].coordinates[1]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[537].coordinates[1]))**2) +
    (((mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[661].coordinates[2]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[537].coordinates[2]))**2))
```

```
#dist_hum_x1_new = ((mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[1].coordinates[0]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[54].coordinates[0]))
dist_scap_x1 = sqrt((((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[661].coordinates[0]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[537].coordinates[0])) **2) +
    (((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[661].coordinates[1]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[537].coordinates[1]))**2) +
    (((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[661].coordinates[2]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[537].coordinates[2]))**2))
#dist_hum_x1 = ((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %j].nodes[1].coordinates[0]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %j].nodes[54].coordinates[0]))
dist_scap_y1_original = sqrt((((mdb.models['Model-2'].rootAssembly.instances['Scapula_0-1'].nodes[88].coordinates[0]) - (mdb.models['Model-2'].rootAssembly.instances['Scapula_0-1'].nodes[568].coordinates[0])) **2) +
    (((mdb.models['Model-2'].rootAssembly.instances['Scapula_0-1'].nodes[88].coordinates[1]) - (mdb.models['Model-2'].rootAssembly.instances['Scapula_0-1'].nodes[568].coordinates[1]))**2) +
    (((mdb.models['Model-2'].rootAssembly.instances['Scapula_0-1'].nodes[88].coordinates[2]) - (mdb.models['Model-2'].rootAssembly.instances['Scapula_0-1'].nodes[568].coordinates[2]))**2))
dist_scap_y1_new = sqrt((((mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[88].coordinates[0]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[568].coordinates[0])) **2) +
```

```
    (((mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[88].coordinates[1]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[568].coordinates[1]))**2) +
    (((mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[88].coordinates[2]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[568].coordinates[2]))**2))
#dist_hum_y1_new = ((mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[414].coordinates[1]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[134].coordinates[1]))
dist_scap_y1 = sqrt((((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[88].coordinates[0]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[568].coordinates[0])) **2) +
    (((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[88].coordinates[1]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[568].coordinates[1]))**2) +
    (((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[88].coordinates[2]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[568].coordinates[2]))**2))
#dist_hum_y1 = ((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %j].nodes[414].coordinates[1]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %j].nodes[134].coordinates[1]))
dist_scap_z1_original = sqrt((((mdb.models['Model-2'].rootAssembly.instances['Scapula_0-1'].nodes[467].coordinates[0]) - (mdb.models['Model-2'].rootAssembly.instances['Scapula_0-1'].nodes[568].coordinates[0])) **2) +
    (((mdb.models['Model-2'].rootAssembly.instances['Scapula_0-1'].nodes[467].coordinates[1]) - (mdb.models['Model-2'].rootAssembly.instances['Scapula_0-1'].nodes[568].coordinates[1]))**2) +
```

```
    (((mdb.models['Model-2'].rootAssembly.instances['Scapula_0-1'].nodes[467].coordinates[2]) - (mdb.models['Model-2'].rootAssembly.instances['Scapula_0-1'].nodes[568].coordinates[2]))**2))
dist_scap_z1_new = sqrt((((mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[467].coordinates[0]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[568].coordinates[0])) **2) +
    (((mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[467].coordinates[1]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[568].coordinates[1]))**2) +
    (((mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[467].coordinates[2]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[568].coordinates[2]))**2))
#dist_hum_z1_new = ((mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[46].coordinates[2]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[61].coordinates[2]))
dist_scap_z1 = sqrt((((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[467].coordinates[0]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[568].coordinates[0])) **2) +
    (((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[467].coordinates[1]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[568].coordinates[1]))**2) +
    (((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[467].coordinates[2]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[568].coordinates[2]))**2))
#dist_scap_y1_new = ((mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[0].coordinates[1]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[17].coordinates[1]))
```

```
#dist_scap_y2_new = ((mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[35].coordinates[1]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[57].coordinates[1]))
#dist_scap_y3_new = ((mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[31].coordinates[1]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[18].coordinates[1]))
#dist_scap_y1 = ((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[0].coordinates[1]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[17].coordinates[1]))
#dist_scap_y2 = ((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[35].coordinates[1]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[57].coordinates[1]))
#dist_scap_y3 = ((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[31].coordinates[1]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[18].coordinates[1]))
#dist_scap_z1_new = ((mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[30].coordinates[2]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[42].coordinates[2]))
#dist_scap_z2_new = ((mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[25].coordinates[2]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[47].coordinates[2]))
#dist_scap_z3_new = ((mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[19].coordinates[2]) - (mdb.models['Model-%d' %(count)].rootAssembly.instances['Scapula_%d-1' %j].nodes[52].coordinates[2]))
#dist_scap_z1 = ((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[30].coordinates[2]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[42].coordinates[2]))
```

```
#dist_scap_z2 = ((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[25].coordinates[2]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[47].coordinates[2]))
#dist_scap_z3 = ((mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[19].coordinates[2]) - (mdb.models['Model-%d' %(count-3)].rootAssembly.instances['Scapula_%d-1' %(j-1)].nodes[52].coordinates[2]))
#avg_scap_x = (dist_scap_x1 + dist_scap_x2 + dist_scap_x3)/3
#avg_scap_y = (dist_scap_y1 + dist_scap_y2 + dist_scap_y3)/3
#avg_scap_z = (dist_scap_z1 + dist_scap_z2 + dist_scap_z3)/3
#avg_scap_x_new = (dist_scap_x1_new + dist_scap_x2_new + dist_scap_x3_new)/3
#avg_scap_y_new = (dist_scap_y1_new + dist_scap_y2_new + dist_scap_y3_new)/3
#avg_scap_z_new = (dist_scap_z1_new + dist_scap_z2_new + dist_scap_z3_new)/3
#scale_scap_x = scale_hum_x #(avg_scap_x_new/avg_scap_x)
scale_scap_y = (dist_scap_y1_new/dist_scap_y1) #(avg_scap_y_new/avg_scap_y)
scale_scap_z = (dist_scap_z1_new/dist_scap_z1) #(avg_scap_z_new/avg_scap_z)
scale_scap_x = (dist_scap_x1_new/dist_scap_x1)
scale_scap_tot_y = (dist_scap_y1_new/dist_scap_y1_original) #(avg_scap_y_new/avg_scap_y)
scale_scap_tot_z = (dist_scap_z1_new/dist_scap_z1_original) #(avg_scap_z_new/avg_scap_z)
scale_scap_tot_x = (dist_scap_x1_new/dist_scap_x1_original)
scale_tot_x = ((scale_hum_tot_x)+ (scale_scap_tot_x))/2
scale_tot_y = ((scale_hum_tot_y)+ (scale_scap_tot_y))/2
scale_tot_z = ((scale_hum_tot_z)+ (scale_scap_tot_z))/2
new_file = open("scale.txt","w")
new_file.write("scale_tot_x =    " + str(scale_tot_x) + "\n")
new_file.write("scale_tot_y =    " + str(scale_tot_y) + "\n")
new_file.write("scale_tot_z =    " + str(scale_tot_z) + "\n")
new_file.write("scale_hum_x =    " + str(scale_hum_x) + "\n")
new_file.write("scale_hum_y =    " + str(scale_hum_y) + "\n")
```

```
new_file.write("scale_hum_z =    " + str(scale_hum_z) + "\n")
new_file.write("scale_scap_x =    " + str(scale_scap_x) + "\n")
new_file.write("scale_scap_y =    " + str(scale_scap_y) + "\n")
new_file.write("scale_scap_z =    " + str(scale_scap_z) + "\n")
new_file.write("scale_hum_overall_x =    " + str(scale_hum_tot_x) + "\n")
new_file.write("scale_hum_overall_y =    " + str(scale_hum_tot_y) + "\n")
new_file.write("scale_hum_overall_z =    " + str(scale_hum_tot_z) + "\n")
new_file.write("scale_scap_overall_x =    " + str(scale_scap_tot_x) + "\n")
new_file.write("scale_scap_overall_y =    " + str(scale_scap_tot_y) + "\n")
new_file.write("scale_scap_overall_z =    " + str(scale_scap_tot_z) + "\n")
#new_file.write("node_dist_x_new =    " + str(dist_x_new) + "\n")
#new_file.write("node_dist_y_new =    " + str(dist_y_new) + "\n")
#new_file.write("node_dist_z_new =    " + str(dist_y_new) + "\n")
new_file.close()
new_file = open("scale_%d.txt" %j,"w")
new_file.write("scale_tot_x =    " + str(scale_tot_x) + "\n")
new_file.write("scale_tot_y =    " + str(scale_tot_y) + "\n")
new_file.write("scale_tot_z =    " + str(scale_tot_z) + "\n")
new_file.write("scale_hum_x =    " + str(scale_hum_x) + "\n")
new_file.write("scale_hum_y =    " + str(scale_hum_y) + "\n")
new_file.write("scale_hum_z =    " + str(scale_hum_z) + "\n")
new_file.write("scale_scap_x =    " + str(scale_scap_x) + "\n")
new_file.write("scale_scap_y =    " + str(scale_scap_y) + "\n")
new_file.write("scale_scap_z =    " + str(scale_scap_z) + "\n")
new_file.write("scale_hum_overall_x =    " + str(scale_hum_tot_x) + "\n")
new_file.write("scale_hum_overall_y =    " + str(scale_hum_tot_y) + "\n")
new_file.write("scale_hum_overall_z =    " + str(scale_hum_tot_z) + "\n")
new_file.write("scale_scap_overall_x =    " + str(scale_scap_tot_x) + "\n")
```

```
    new_file.write("scale_scap_overall_y =    " + str(scale_scap_tot_y) + "\n")
    new_file.write("scale_scap_overall_z =    " + str(scale_scap_tot_z) + "\n")
    #new_file.write("node_dist_x_new =    " + str(dist_x_new) + "\n")
    #new_file.write("node_dist_y_new =    " + str(dist_y_new) + "\n")
    #new_file.write("node_dist_z_new =    " + str(dist_y_new) + "\n")
    new_file.close()
    count_file = open("count_data.py","w")
    count_file.write("count =  " + str(count) + "\n")
    count_file.write("cyc =  " + str(cyc) + "\n")
    count_file.close()
    mdb.saveAs(pathName='C:/Nikhil_D/Research/Abaqus/I2M2/postgang_trial_22/M1_%d.cae' %j)
    mdb.saveAs(pathName='C:/Nikhil_D/Research/Abaqus/I2M2/postgang_trial_22/M1.cae')

import sys
sys.exit()
```