

<sup>1</sup> Hit and Run and Stuff

<sup>2</sup> Brian Cohn May Szedlák

<sup>3</sup> March 28, 2015

<sup>4</sup> **Abstract**

The brain must select its control strategies among an infinite set of possibilities, thereby solving an optimization problem. While this set is infinite and lies in high dimensions, it is bounded by kinematic, neuromuscular, and anatomical constraints, within which the brain must select optimal solutions. We use data from a human index finger with 7 muscles, 4DOF, and 4 output dimensions. For a given force vector at the endpoint, the feasible activation space is a 3D convex polytope, embedded in the 7D unit cube. It is known that explicitly computing the volume of this polytope can become too computationally complex in many instances. We generated random points in the feasible activation space using the Hit-and-Run method, which converged to the uniform distribution. After generating enough points, we computed the distribution of activation across each muscle, shedding light onto the structure of these solution spaces- rather than simply exploring their maximal and minimal values. We also visualize the change in these activation distributions as we march toward maximal feasible force production in a given direction. Using the parallel coordinates method, we visualize the connection between the muscle activations. Once can then explore the feasible activation space, while constraining certain muscles. Although this paper presents a 7 dimensional case of the index finger, our methods extend to systems with up to at least 40 muscles. We challenge the community to map the shapes distributions of each variable in the solution space, thereby providing important contextual information into optimization of motor cortical function in future research.

<sup>26</sup> **1 Author Summary**

<sup>27</sup> **2 Introduction**

<sup>28</sup> Optimal control of a musculoskeletal system is intrinsically related to mechanical con-  
<sup>29</sup> straints. An endpoint's end effector forces are highly dependent upon tendon force  
<sup>30</sup> ranges, the leverage of each tendon insertion point across each joint, and the planes of  
<sup>31</sup> motion each degree of freedom (DOF), with these physical relationships defining the ca-  
<sup>32</sup> pabilities of the system. In spite of the complexity of alpha-gamma neuromuscular drive  
<sup>33</sup> models, every system exists under limitations intrinsic to physical mechanics, and as  
<sup>34</sup> such, limbs have been modeled to behave under these constraints with stunning realism  
<sup>35</sup> [cite]. With increasingly accurate and faceted models, a great body of research has been  
<sup>36</sup> tasked with predicting kinetics, while being sensitive to subtle changes in muscle activa-  
<sup>37</sup> tion [todorov's mujoco], skeletal weight distributions, neural synergies, and spatiotem-  
<sup>38</sup> poral variables[Kornelius and FVC, Racz FVC]. While many of these models highlight  
<sup>39</sup> their accuracy , and attribute it to nonlinear dynamic modeling, linear approximation  
<sup>40</sup> has long-remained a viable way to interpret the actions of physical limb systems, in the  
<sup>41</sup> context of a well-understood mathematical framework. As limbs exist under physical  
<sup>42</sup> constraints, neuromuscular control must strategize within the generic Newtonian laws  
<sup>43</sup> of physics, in the realm of linear statics and dynamics. While some would argue that  
<sup>44</sup> linear approximation of a musculoskeletal system is a blunt instrument in researching  
<sup>45</sup> what is considered a 'non-linear' system, linear approximation can offer a 'big picture  
<sup>46</sup> view' of the system. Some attention has been given to the constraints that physical  
<sup>47</sup> systems impart on control itself ['nice try' citations], with many placing emphasis on  
<sup>48</sup> non-linear synergies between motor units, for instance, between the *vastus lateralis* and  
<sup>49</sup> *vastus medialis* muscles of the leg. A breadth of modeling techniques have been applied  
<sup>50</sup> to physical systems to model and understand CNS control under the constants of a  
<sup>51</sup> given task, and many have been able to visualize some of the limitations animals must  
<sup>52</sup> abide by in optimization.

<sup>53</sup> Optimal control theory must be implemented in a way such that it is computationally  
<sup>54</sup> tractable. Control systems of designed (robotic) and evolved (neurophysiologic) origins  
<sup>55</sup> can afford only a small measure of latency. Identifying how optimal control works within  
<sup>56</sup> the framework of constraints could bring rise to more efficient algorithms, and this  
<sup>57</sup> contextual understanding could introduce new ways to visualize how neuromuscular  
<sup>58</sup> systems learn to improve over training. In dynamic systems we have seen jdo research  
<sup>59</sup> on this;[cite].

<sup>60</sup> In a static system, every possible combination of independent muscle activations  
<sup>61</sup> exists within the unit-n-cube, where N is set to however many independently-controlled  
<sup>62</sup> muscles a system has. Prior work has highlighted the relationship between the feasible  
<sup>63</sup> force space and the set of all activation solutions.[cite papers in the last 10 years] In  
<sup>64</sup> effect, adding constraints on the FFS (e.g. requiring only force in a given plane) adds  
<sup>65</sup> constraints to the FAS

<sup>66</sup> The effect of each muscle on each joint has been represented by the moment arm

67 matrix [citations], the relationship of each DOF on end-effector output directions . The  
68 feasible force set (described in detail in [cite]) is an M-dimensional polytope containing  
69 all possible force vectors an endpoint can output.

70 neurons do alot of stuff, and much work has been put into understanding how neural  
71 drive results in force, motion, and kinetics. physical description of a musculoskeletal  
72 system

73 Functional performance is defined by the ability for a system to identify optimal  
74 solutions in a set of suboptimal solutions. {talk about local and global maxima and  
75 minima in neuro optimization control theory}

76 The feasible force set represents every possible output force an end effector can impart  
77 on an endpoint.

78 Described in a mathematical way the feasible activation set is expressed as follows.  
79 For a given force vector  $f \in \mathbb{R}^m$ , which are the activations that satisfy

$$\mathbf{f} = A\mathbf{a}, \mathbf{a} \in [0, 1]^n?$$

80 In our 7-dimensional example  $m = 4$  and  $n = 7$ , typically  $n$  is much larger than  $m$ .  
81 The constraint  $\mathbf{a} \in [0, 1]^n$  describes that the feasible activation space lies in the  $n$ -  
82 dimensional unit cube (also called the  $n$ -cube). Each row of the constraint  $\mathbf{f} = A\mathbf{a}$  is a  
83  $n - 1$  dimensional hyperplane. Assuming that the rows in  $A$  are linearly independent  
84 (which is a safe assumption in the muscle system case), the intersection of all  $m$  equality  
85 constraints constraints is a  $(n - m)$ -dimensional hyperplane. Hence the feasible activation  
86 set is the polytope given by the intersection of the  $n$ -cube and an  $(n - m)$ -dimensional  
87 hyperplane. Note that this intersection is empty in the case where the force  $f$  can not  
88 be generated.

89 Issues with volume computations: As realistic musculoskeletal systems has many  
90 more muscles, it's important for polytope calculation to be scalable to higher dimensions.

91 We first describe the stochastic method of hit-and-run, and illustrate its use on a  
92 fabricated 3-muscle, 1-DOF system with a desired force output of 1N. We designed  
93 this schematic (but mathematically viable) linear system of constraints to help readers  
94 understand the mechanics of hit-and-run mathematics. Our index-finger model has too  
95 many dimensions to show how the process works, so we hope this will help readers  
96 understand what is going on in n dimensions (7 in the case of the index-finger model).  
97 We also used this model to perform unit tests on our code in thoroughly validating our  
98 hit-and-run implementation.

99 We investigated the distributions of the feasible activation set across each muscle.  
100 State the purpose of the work in the form of the hypothesis, question, or problem you  
101 investigated; and, Briefly explain your rationale and approach and, whenever possible,  
102 the possible outcomes your study can reveal.

103 **3 Materials and Methods**

104 **3.1 Data and Samples**

105 We began with an index finger of a human (male) hand, which was taken from [].  
106 Dissected by []. Experimental forces from []. IUPAC licenses and other important facts.  
107 Measurements were made with a ruler, with  $\pm x$ .  $R$  Moment Arm Matrix  $J$  Endpoint  
108 Jacobian  $F_o$  MIC for each muscle

109 **3.2 Polytope representation of the feasible activation space**

110 Exact volume calculations for polygons can only be done in reasonable time in up to  
111 10 dimensions [2, 6, 7]. We therefore use the so called Hit-and-Run approach, which  
112 samples a series of points in a given polygon. Given the points for a feasible activation  
113 space, this method gives us a deeper understanding of its underlying structure.

114 **3.3 Hit-and-Run**

115 In this section we introduce the Hit-and-Run algorithm used for uniform sampling in a  
116 convex body  $K$ , was introduced by Smith in 1984 [10]. The mixing time is known to be  
117  $\mathcal{O}^*(n^2R^2/r^2)$ , where  $R$  and  $r$  are the radii of the inscribed and circumscribed ball of  $K$   
118 respectively [1, 8]. I.e., after  $\mathcal{O}^*(n^2R^2/r^2)$  steps of the Hit-and-Run algorithm we are at  
119 a uniformly at random point in the convex body. In the case of the muscles of a limb,  
120 we are interested in the polygon  $P$  that is given by the set of all possible activations  
121  $\mathbf{a} \in \mathbb{R}^n$  that satisfy

$$\mathbf{f} = A\mathbf{a}, \mathbf{a} \in [0, 1]^n,$$

where  $\mathbf{f} \in \mathbb{R}^m$  is a fixed force vector and  $A = J^{-T}RF_m \in \mathbb{R}^{m \times n}$ .  $P$  is bounded by the  
unit  $n$ -cube since all variables  $a_i, i \in [n]$  are bounded by 0 and 1 from below, above  
respectively. Consider the following  $1 \times 3$  example.

$$1 = \frac{10}{3}a_1 - \frac{53}{15}a_2 + 2a_3 \\ a_1, a_2, a_3 \in [0, 1],$$

122 the set of feasible activations is given by the shaded set in Figure 1.

123 The Hit-and-Run walk on  $P$  is defined as follows (it works analogously for any convex  
124 body).

- 125 1. Find a given starting point  $\mathbf{p}$  of  $P$  (Figure 2a) .  
126 2. Generate a random direction through  $\mathbf{p}$  (uniformly at random over all directions)  
127 (Figure 2b).  
128 3. Find the intersection points of the random direction with the  $n$ -unit cube (Figure  
129 2c).

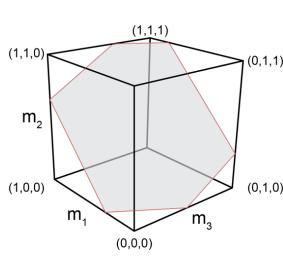


Figure 1: Feasible Activation

- 130     4. Choose the next point of the sampling algorithm uniformly at random from the  
 131       segment of the line in  $P$  (Figure 2d).
- 132     5. Repeat from (b) the above steps with the new point as the starting point .

133     The implementation of this algorithm is straight forward except for the choice of the  
 134       random direction. How do we sample uniformly at random (u.a.r.) from all directions  
 135       in  $P$ ? Suppose that  $\mathbf{q}$  is a direction in  $P$  and  $p \in P$ . Then by definition of  $P$ ,  $\mathbf{q}$  must  
 136       satisfy  $\mathbf{f} = A(\mathbf{p} + \mathbf{q})$ . Since  $\mathbf{p} \in P$ , we know that  $\mathbf{f} = A\mathbf{p}$  and therefore

$$\mathbf{f} = A(\mathbf{p} + \mathbf{q}) = \mathbf{f} + A\mathbf{q}$$

137     and hence

$$A\mathbf{q} = 0.$$

138     We therefore need to choose directions uniformly at random from all directions in  
 139       the vectorspace

$$V = \{\mathbf{q} \in \mathbb{R}^n | A\mathbf{q} = 0\}.$$

140     As shown by Marsaglia this can be done as follows [9].

- 141     1. Find an orthonormal basis  $b_1, \dots, b_r \in \mathbb{R}^n$  of  $A\mathbf{q} = 0$ .
- 142     2. Choose  $(\lambda_1, \dots, \lambda_r) \in \mathcal{N}(0, 1)^n$  (from the Gaussian distribution).
- 143     3.  $\sum_{i=1}^r \lambda_i b_i$  is a u.a.r. direction.

144     A basis of a vectorspace  $V$  is a minimal set of vectors that generate  $V$ , and it is  
 145       orthonormal if the vectors are pairwise orthogonal (perpendicular) and have unit length.  
 146       Using basic linear algebra one can find a basis for  $V = \{A\mathbf{q} = 0\}$  and orthogonalize it  
 147       with the well known Gram-Schmidt method (for details see e.g. [3]). Note that in order  
 148       to get the desired u.a.r. sample the basis needs to be orthonormal. For the limb case we  
 149       can safely assume that the rows of  $A$  are linearly independent and hence the number of  
 150       basis vectors is  $n - m$ .

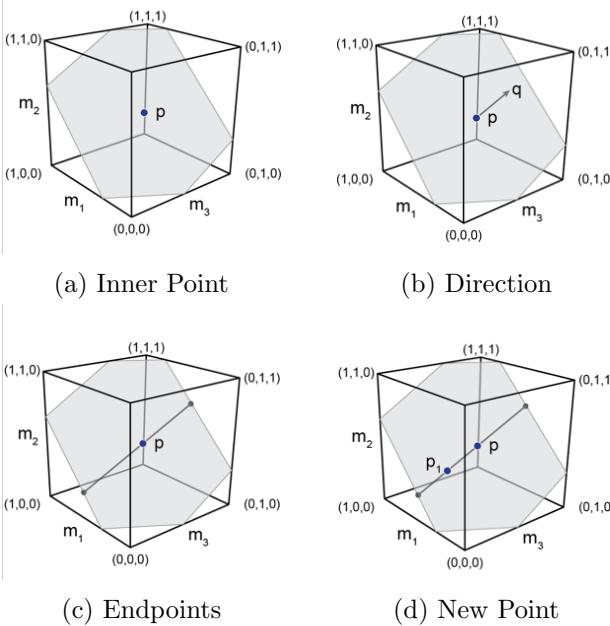


Figure 2: Hit-and-Run Step

### 151 3.4 Mixing Time

152 How many steps are necessary to reach a uniformly at random point in the polytope?  
 153 The theoretical bound  $\mathcal{O}^*(n^2 R^2 / r^2)$  given in [8] has a very large hidden coefficient ( $10^{30}$ )  
 154 which makes the algorithm almost infeasible in lower dimensions.

155 These bounds hold for general convex sets. For convex polygons in higher dimensions,  
 156 experimental results suggest that  $\mathcal{O}(n)$  steps of the Hit-and-Run algorithm are sufficient.  
 157 In particular Emiris and Fisikopoulos paper suggest that  $(10 + 10\frac{n}{r})n$  steps are enough  
 158 to have a close to uniform distribution [4]. In all cases tested, sampling more point did  
 159 not make accuracy significantly higher.

160 Ge et al. showed experimentally that up to about 40 dimensions, ??? random points  
 161 seem to suffice to get a close to uniform distribution [5].

162 Therfore for given output force we execute the Hit-and-Run algorithm 1000 times  
 163 on 100 points. The experimental results propose that those 1000 points are uniformly  
 164 distributed on the polygon.

165 As a additional control, for each muscle we observe that the theoretical upper and  
 166 lower bound of the feasible activation match the observed corresponding bounds (dif-  
 167 ference max ??). To find the theoretical upperbound (lowerbound) of a given muscle  
 168 activation we solve two linear programs maximizing (minimizing)  $a_i$  over the polytope.

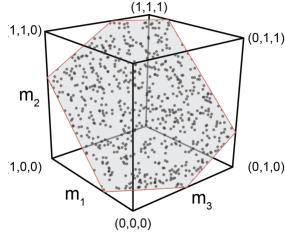


Figure 3: Uniform Distribution

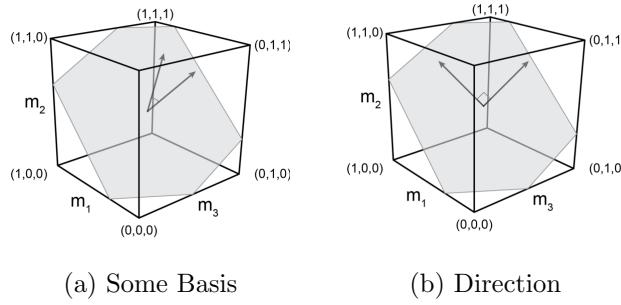


Figure 4: Find Orthonormal Basis

### 169 3.5 Starting Point

170 To find a starting point in

$$\mathbf{f} = A\mathbf{a}, \mathbf{a} \in [0, 1]^n,$$

171 we only need to find a feasible activation vector. For the hit and run algorithm to mix  
172 faster, we do not want the starting point to be in a vertex of the activation space. We  
173 use the following standard trick using slack variables  $\epsilon_i$ .

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^n \epsilon_i \\ & \text{subject to} && \mathbf{f} = A\mathbf{a} \\ & && a_i \in [\epsilon_i, 1 - \epsilon_i], \quad \forall i \in \{1, \dots, n\} \\ & && \epsilon_i \geq 0, \quad \forall i \in \{1, \dots, n\}. \end{aligned} \tag{1}$$

174 This approach can still fail in theory, but this method has the choose  $\epsilon_i > 0$  and there-  
175 fore  $a_i \neq 0$  or  $1$ . Since for all vertices of the feasible activation space lie on the boundary  
176 of the  $n$ -cube, at least  $n - m$  muscles must have activation  $0$  or  $1$ . Documentation is  
177 included in our supplementary information.

178 **3.6 Parallel Coordinates: Visualization of the Feasible Activation Space**

179 Citation A common way to visualize higher dimensional data is using parallel coordinates.  
180 To show our sample set of points in the feasible activation space we draw  $n$   
181 parallel lines, which representing the activations of the  $n$  muscles. Each point is then  
182 represented by connecting their coordinates by  $n - 1$  lines.

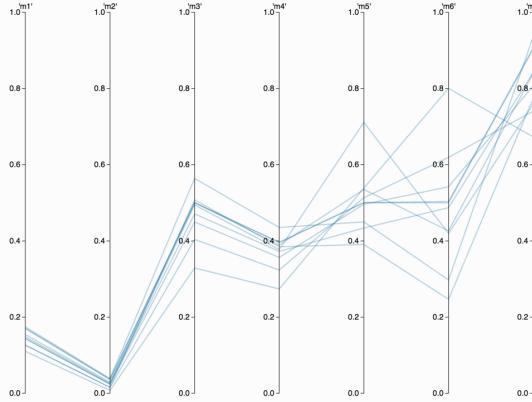


Figure 5: Feasible Activation

183 Using an interactive surface one can now restrict each muscle function to any desired  
184 interval, e.g., figure ??.

185 **NICE FIGURE OF RESTRICTED PARALLEL COORDINATES**

186 For the  $l_1$ ,  $l_2$  and  $l_3$  norm respectively, we added an additional line to represent  
187 the corresponding weight. E.g. for a given point  $\mathbf{a} \in \mathbb{R}^n$  we are interested in  $\sum_{i=1}^n a_i$ ,  
188  $\sqrt{\sum_{i=1}^n a_i^2}$  and  $\sqrt[3]{\sum_{i=1}^n a_i^3}$ . As for the muscles one can restrict the intervals of the weight  
189 functions, to explore the corresponding feasible activation space.

190 **NICE PICTURE WITH WEIGHTS INCLUDED**

191 **4 Results**

192 Many nice figures

193 1. Histograms

194 2. Histograms 3 directions

195 3. PC

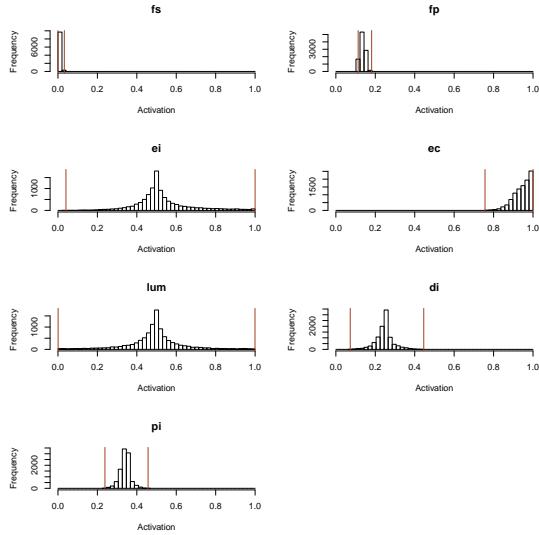


Figure 6: Histogram for Fixed Force

#### 196 4.1 Activation Distribution on a Fixed Force Vector

#### 197 4.2 Changing Output Force in 3 Directions

198 We discuss different forces into three different directions, which are given by the palmar  
 199 direction ( $x$ -direction), the distal direction ( $y$ -direction) and the sum of them. The  
 200 maximal forces into each direction are given by ??, ?? and ?? respectively. For  $\alpha =$   
 201  $0.1, 0.2, \dots, 0.9$ , we give the histograms where the force is  $\alpha \cdot F_{\max}$ , where  $F_{\max}$  is the  
 202 maximum output force in the corresponding direction.

#### 203 4.3 Parallel Coordinates

204 Muscle 5 and 6 same direction and same strength  $\Rightarrow$  Does not matter which one we  
 205 activate for low cost

## 206 5 Discussion

207 Mostly to be written by Brian

#### 208 5.1 Distributions

- 209 • Bounding box away from 0 and 1 means muscle is really needed  $\rightarrow$  Already known  
   210 from the bounding boxes
- 211 • High density  $\rightarrow$  most solutions in that area

212    **5.2 Parallel Coordinates**

- 213    • Parallel lines in PC indicate opposite direction of muscles  
214    • Crossing lines indicate similar direction

215    **5.3 Running Time**

216    The step of the algorithm which are time consuming are finding a starting point, which  
217    solves a linear program and can take exponential running time in worst case. For each  
218    fixed force vector we only have to find a starting point and an orthonormal basis once,  
219    and are hence not of concern for the running time.

220    Running one loop of the hit and run algorithm only needs linear time, therefore the  
221    method will extend to higher dimesions with only linear factor of additinal running time  
222    needed.

223    TODO: add part on how we didn't deal with a dynamical system, but the feasible  
224    force space still exists, just with constraints on muscle activations instantaneously, and  
225    the momentums present in each of the limbs about their joints.

226    TODO: discuss how the FAS is an overestimate of the total number of possible  
227    activation solutions, because of synergies. Our mathematical approach considers 100  
228    percent independent control of every muscle, with no correlations or inverse correlations  
229    between muscle activations. Then slightly discuss how synergies can limit activation  
230    capabilities.

231    **6 Acknowledgments**

232    **References**

- 233    [1] M. Dyer, A. Frieze, and R. Kannan. A random polynomial time algorithm for ap-  
234    proximating the volume of convex bodies. *Proc. of the 21st annual ACM Symposium  
of Theory of Computing*, pages 375–381, 1989.  
235  
236    [2] M. Dyer and M. Frieze. On the complexity of computing the volume of a polyhedron.  
237    *SIAM Journal on Computing*, 17:967–974, 1988.  
238    [3] T. S. Blyth E. F. Robertson. *Basic Linear Algebra*. Springer, 2002.  
239    [4] I.Z. Emiris and V. Fisikopoulos. Efficient random-walk methods for approximating  
240    polytope volume. *Preprint: arXiv:1312.2873v2*, 2014.  
241    [5] C. Ge, F. Ma, and J. Zhang. A fast and practical method to estimate volumes of  
242    convex polytopes. *Preprint: arXiv:1401.0120*, 2013.  
243    [6] L.G. Khachiyan. On the complexity of computing the volume of a polytope. *Izvestia  
Akad. Nauk SSSR Tekhn. Kibernet*, 216217, 1988.  
244

- 245 [7] L.G. Khachiyan. The problem of computing the volume of polytopes is np-hard.  
246        *Uspekhi Mat. Nauk* 44, 179180, 1989.
- 247 [8] L. Lovász. Hit-and-run mixes fast. *Math. Prog.*, 86:443–461, 1998.
- 248 [9] G. Marsaglia. Choosing a point from the surface of a sphere. *Ann. Math. Statist.*,  
249        43:645–646, 1972.
- 250 [10] R.L. Smith. Efficient monte-carlo procedures for generating points uniformly dis-  
251        tributed over bounded regions. *Operations Res.*, 32:1296–1308, 1984.

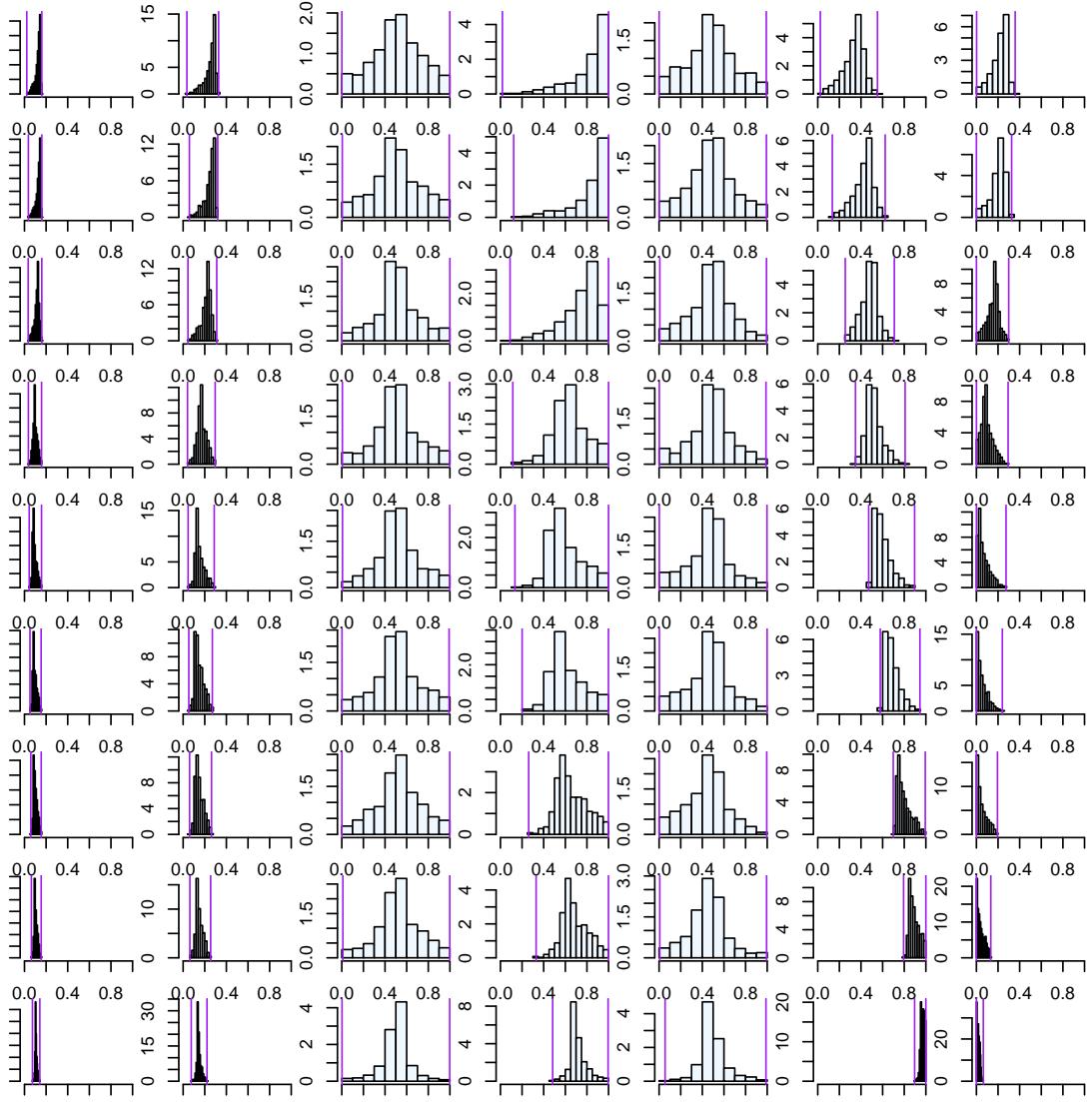


Figure 7: Histogram for  $x$ -Direction

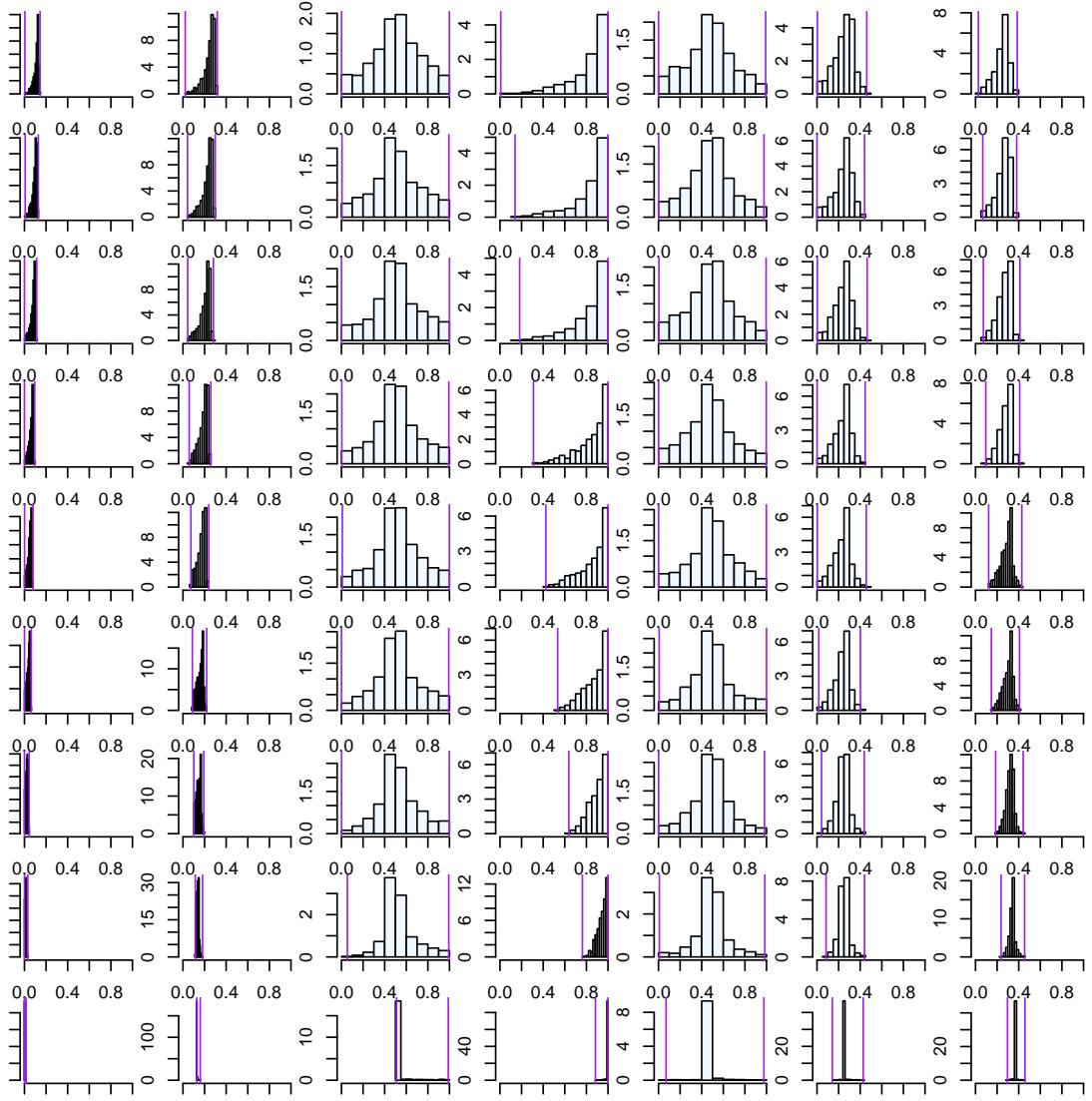


Figure 8: Histogram for  $xy$ -Direction

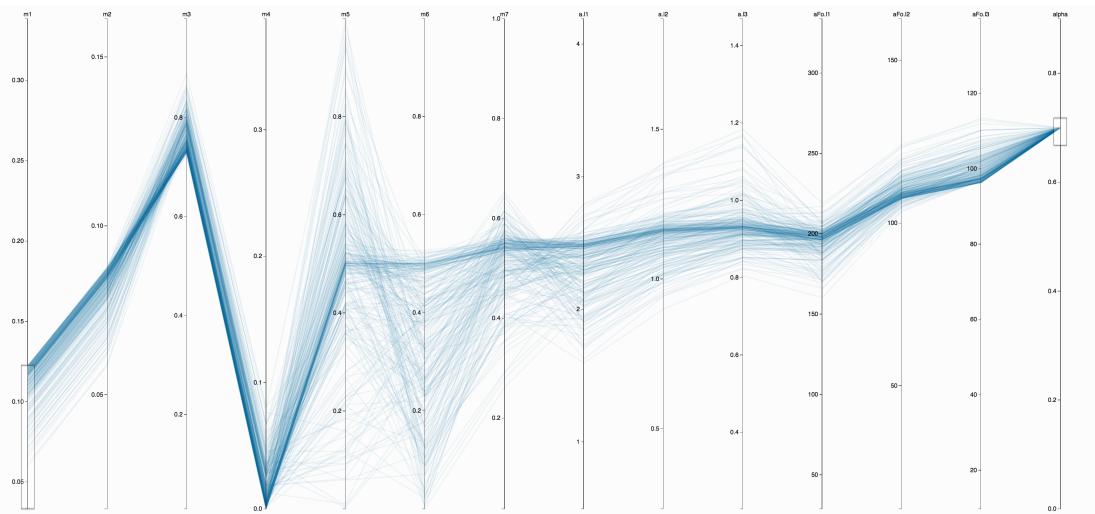


Figure 9: Low for Muscle 1

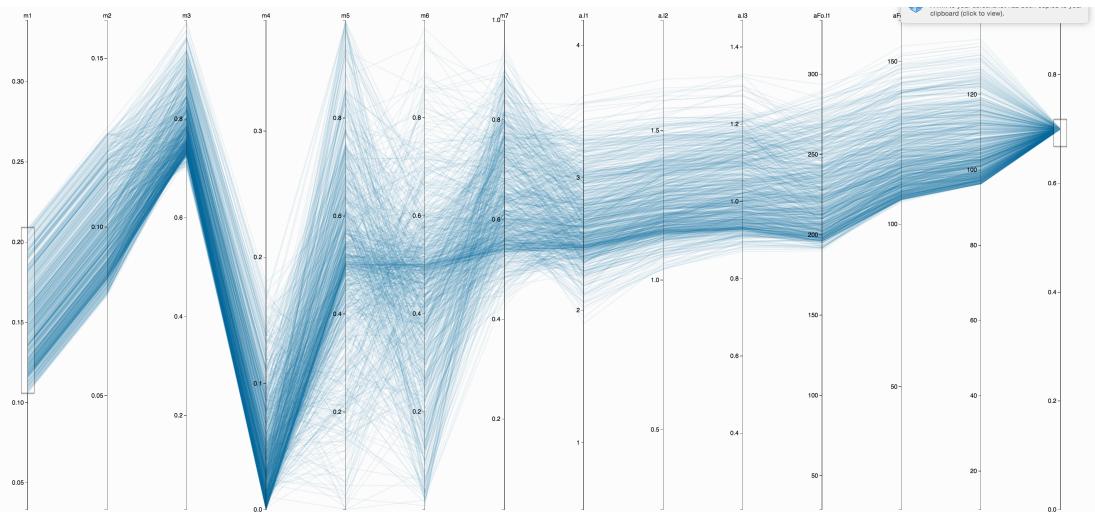


Figure 10: Middle for Muscle 1

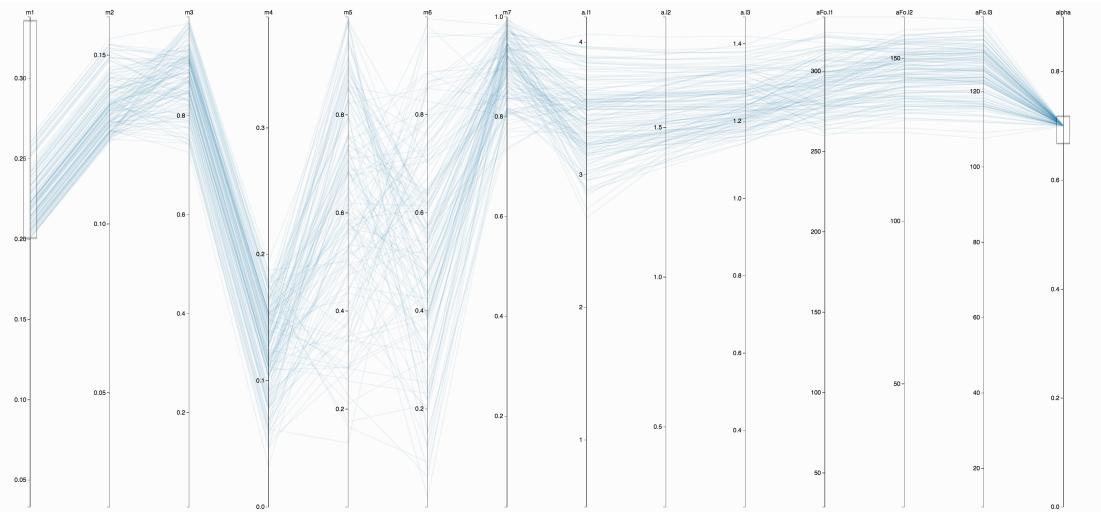


Figure 11: Upper for Muscle 1

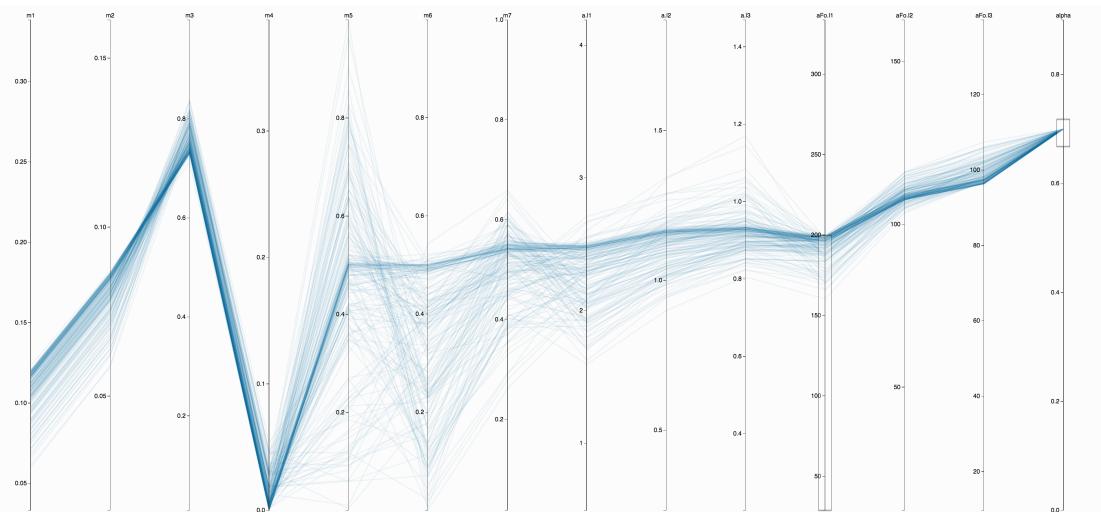


Figure 12: Weighted Cost