

1

Hit and Run and Stuff

2

Brian Cohn May Szedlák

3

March 28, 2015

4

Abstract

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

²⁶ **1 Author Summary**

²⁷ **2 Introduction**

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

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

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

⁶⁶ 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 Linear approximation for degrees of freedom**

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
107 documents.

108 **3.2 Polytope representation of the feasible activation space**

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

113 **3.3 Hit-and-Run**

114 In this section we introduce the Hit-and-Run algorithm used for uniform sampling in a
115 convex body K , was introduced by Smith in 1984 [\[?\]](#). The mixing time is known to be
116 $\mathcal{O}^*(n^2 R^2/r^2)$, where R and r are the radii of the inscribed and circumscribed ball of K
117 respectively [\[?, ?\]](#). I.e., after $\mathcal{O}^*(n^2 R^2/r^2)$ steps of the Hit-and-Run algorithm we are at
118 a uniformly at random point in the convex body. In the case of the muscles of a limb,
119 we are interested in the polygon P that is given by the set of all possible activations
120 $\mathbf{a} \in \mathbb{R}^n$ 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 1×3 example.

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

121 the set of feasible activations is given by the shaded set in Figure ??.

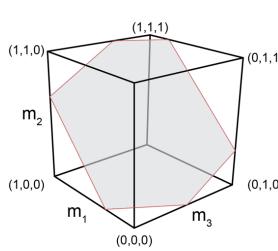


Figure 1: Feasible Activation

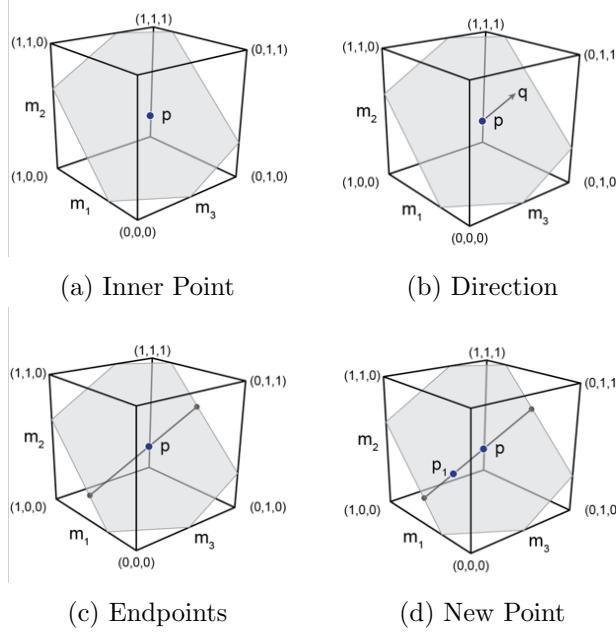


Figure 2: Hit-and-Run Step

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

- 124 1. Find a given starting point \mathbf{p} of P (Figure ??) .
- 125 2. Generate a random direction through \mathbf{p} (uniformly at random over all directions)
 126 (Figure ??).
- 127 3. Find the intersection points of the random direction with the n -unit cube (Figure
 128 ??).
- 129 4. Choose the next point of the sampling algorithm uniformly at random from the
 130 segment of the line in P (Figure ??).
- 131 5. Repeat from (b) the above steps with the new point as the starting point .

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

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

136 and hence

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

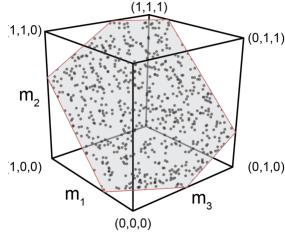


Figure 3: Uniform Distribution

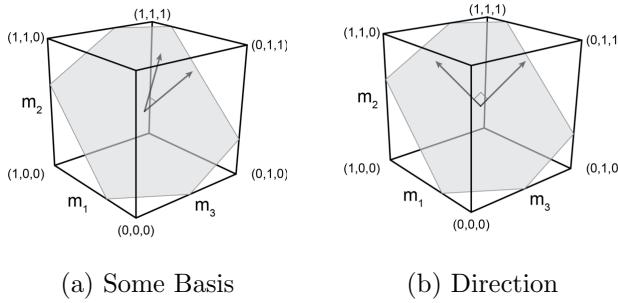


Figure 4: Find Orthonormal Basis

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

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

139 As shown by Marsaglia this can be done as follows [?].

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

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

150 **3.4 Mixing Time**

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

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

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

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

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

168 **3.5 Starting Point**

169 To find a starting point in

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

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

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

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

177 **3.6 Parallel Coordinates: Visualization of the Feasible Activation Space**

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

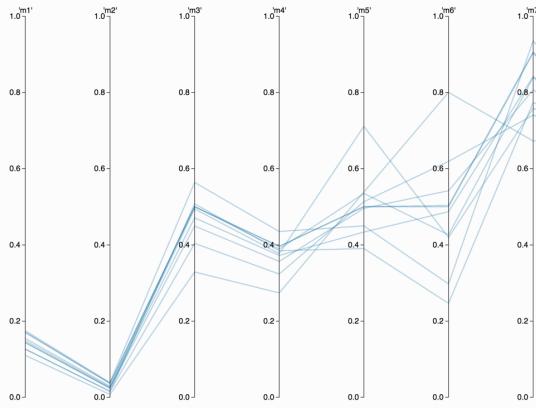


Figure 5: Feasible Activation

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

184 NICE FIGURE OF RESTRICTED PARALLEL COORDINATES

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

189 NICE PICTURE WITH WEIGHTS INCLUDED

190 4 Results

191 Many nice figures

192 1. Histograms

193 2. Histograms 3 directions

194 3. PC

195 4.1 Activation Distribution on a Fixed Force Vector

196 4.2 Changing Output Force in 3 Directions

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

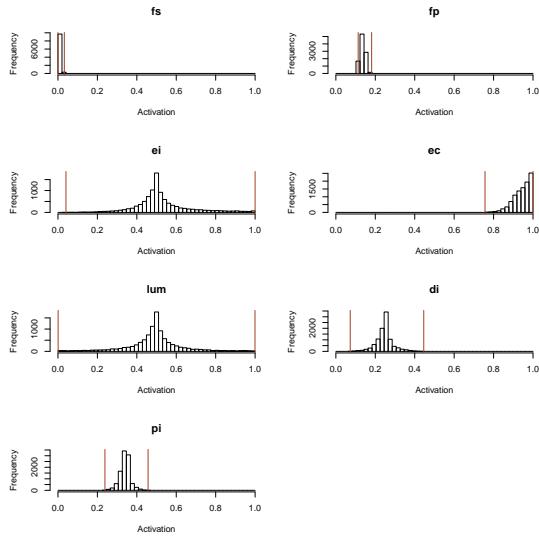


Figure 6: Histogram for Fixed Force

202 4.3 Parallel Coordinates

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

205 5 Discussion

206 Mostly to be written by Brian

207 5.1 Distributions

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

211 5.2 Parallel Coordinates

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

214 5.3 Running Time

215 The step of the algorithm which are time consuming are finding a starting point, which
216 solves a linear program and can take exponential running time in worst case. For each

217 fixed force vector we only have to find a starting point and an orthonormal basis once,
218 and are hence not of concern for the running time.

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

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

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

230 **6 Acknowledgments**

231 **References**

- 232 [1] M. Dyer, A. Frieze, and R. Kannan. A random polynomial time algorithm for ap-
233 proximating the volume of convex bodies. *Proc. of the 21st annual ACM Symposium
234 of Theory of Computing*, pages 375–381, 1989.
- 235 [2] M. Dyer and M. Frieze. On the complexity of computing the volume of a polyhedron.
236 *SIAM Journal on Computing*, 17:967–974, 1988.
- 237 [3] T. S. Blyth E. F. Robertson. *Basic Linear Algebra*. Springer, 2002.
- 238 [4] I.Z. Emiris and V. Fisikopoulos. Efficient random-walk methods for approximating
239 polytope volume. *Preprint: arXiv:1312.2873v2*, 2014.
- 240 [5] C. Ge, F. Ma, and J. Zhang. A fast and practical method to estimate volumes of
241 convex polytopes. *Preprint: arXiv:1401.0120*, 2013.
- 242 [6] L.G. Khachiyan. On the complexity of computing the volume of a polytope. *Izvestia
243 Akad. Nauk SSSR Tekhn. Kibernet*, 216217, 1988.
- 244 [7] L.G. Khachiyan. The problem of computing the volume of polytopes is np-hard.
245 *Uspekhi Mat. Nauk* 44, 179180, 1989.
- 246 [8] L. Lovász. Hit-and-run mixes fast. *Math. Prog.*, 86:443–461, 1998.
- 247 [9] G. Marsaglia. Choosing a point from the surface of a sphere. *Ann. Math. Statist.*,
248 43:645–646, 1972.
- 249 [10] R.L. Smith. Efficient monte-carlo procedures for generating points uniformly dis-
250 tributed over bounded regions. *Operations Res.*, 32:1296–1308, 1984.

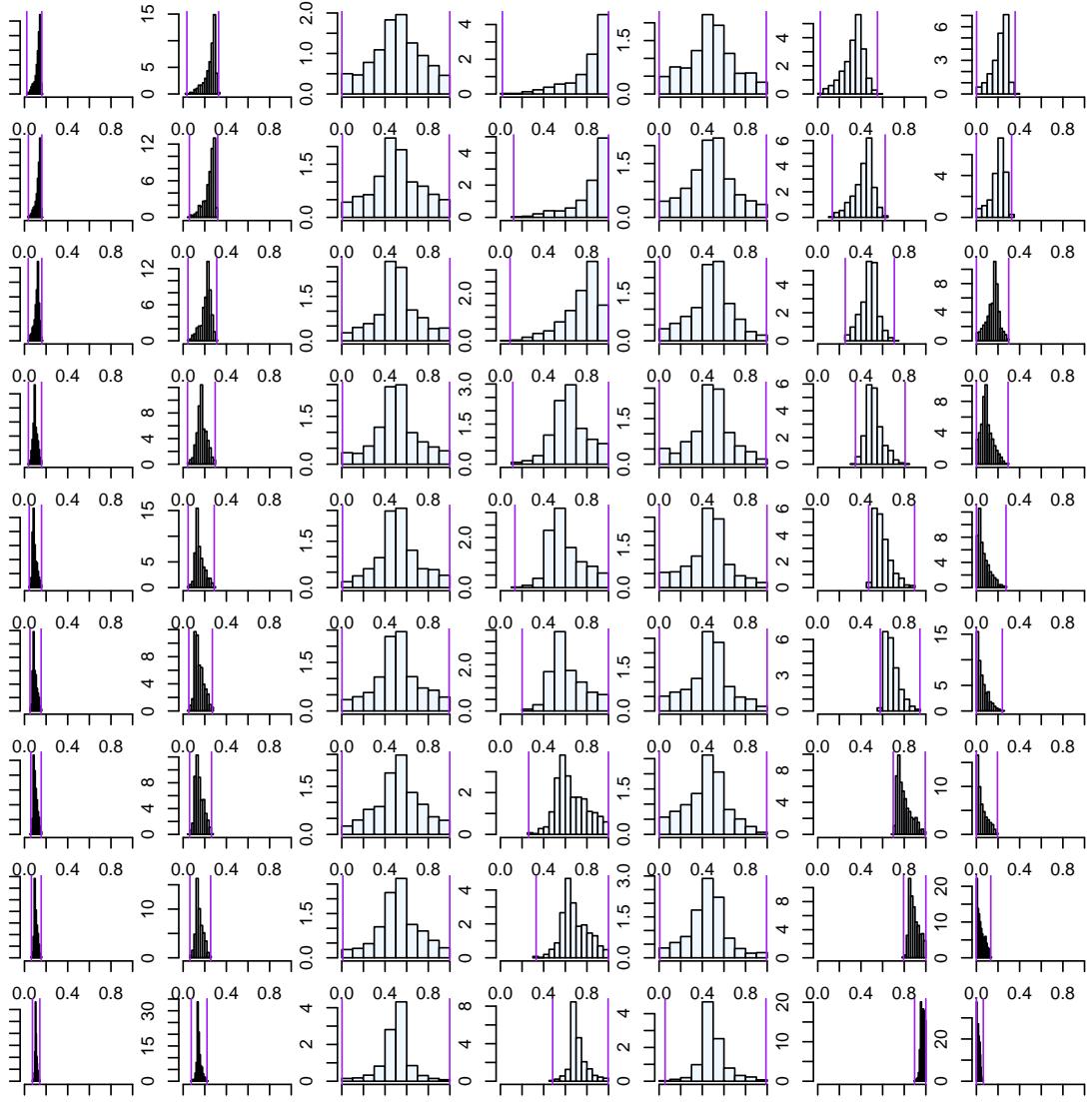


Figure 7: Histogram for x -Direction

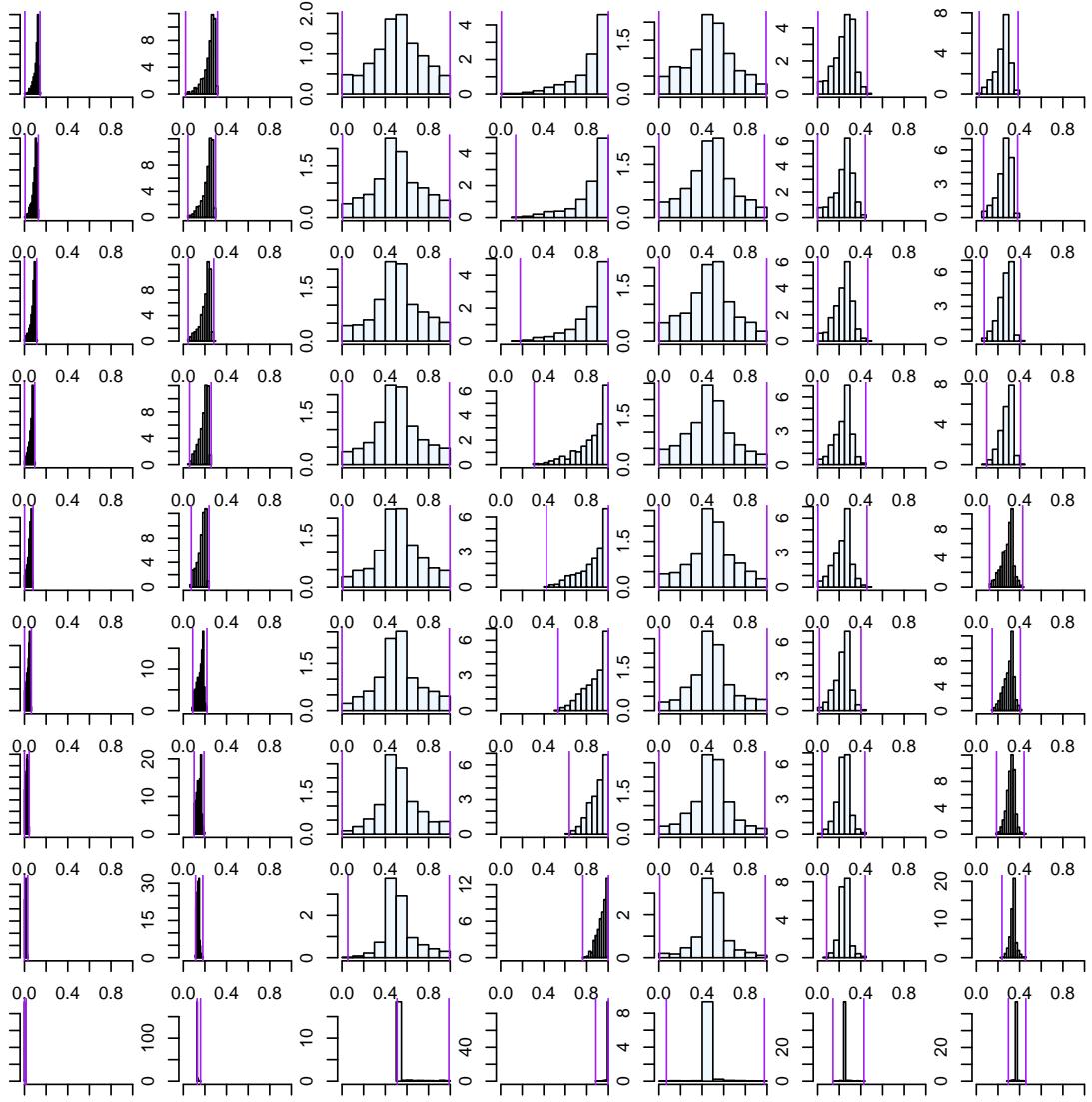


Figure 8: Histogram for xy -Direction

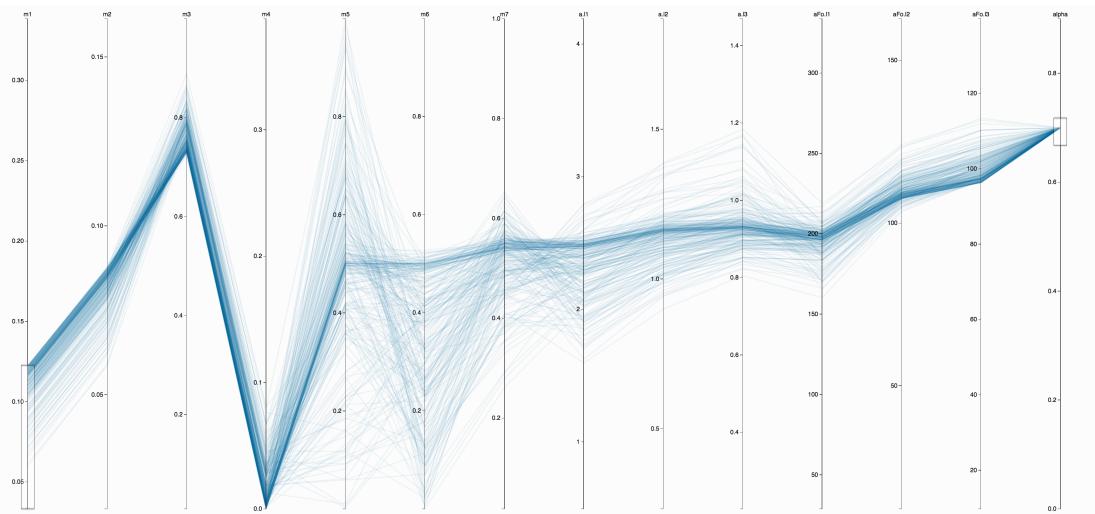


Figure 9: Low for Muscle 1

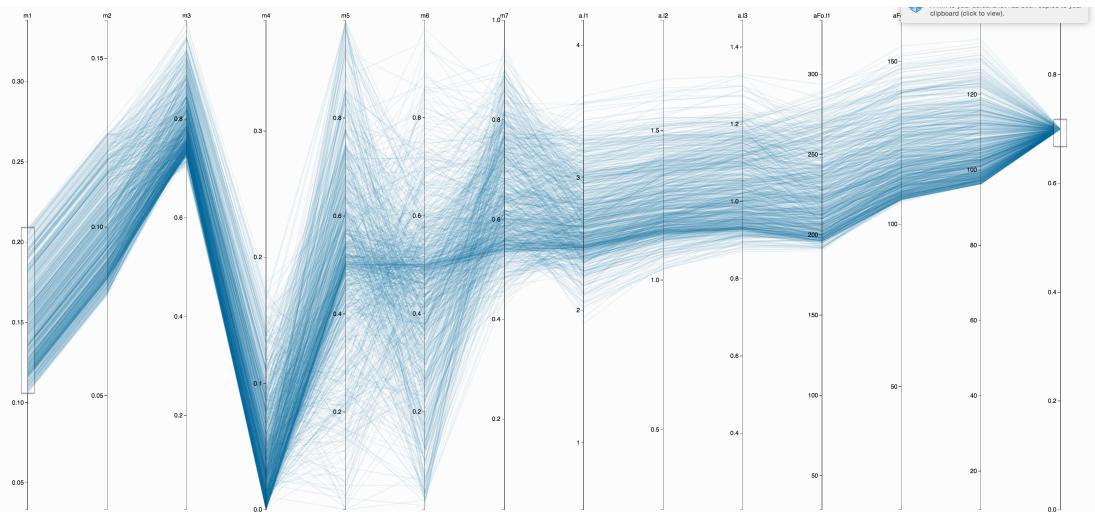


Figure 10: Middle for Muscle 1

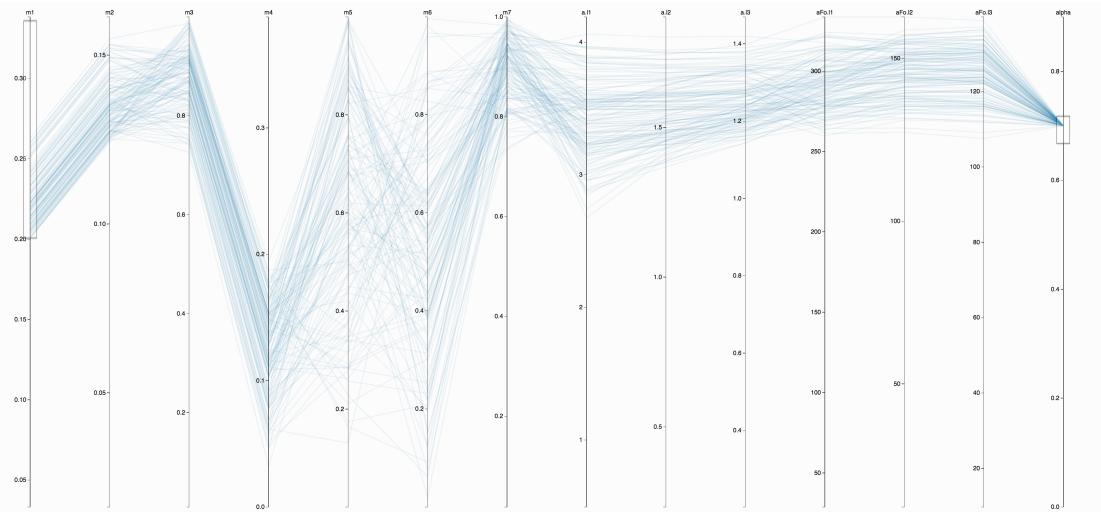


Figure 11: Upper for Muscle 1

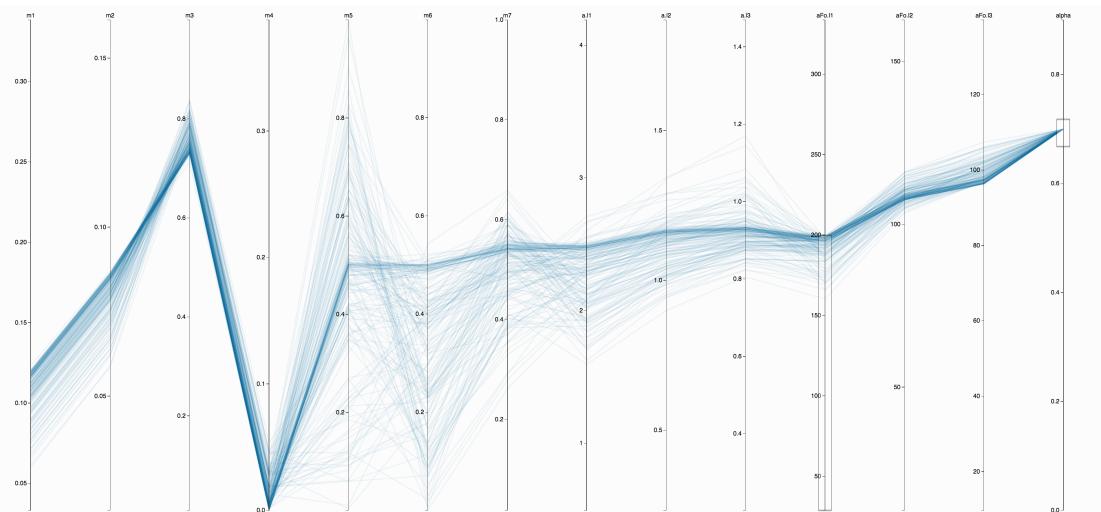


Figure 12: Weighted Cost