Bayesian Mapping of Functional Muscle MRE Data

Eric Barnhill

January 19, 2018

Introduction

Magnetic Resonance Elastography (MRE) generates maps of tissue viscoelastic properties (for more details see [1]). In this study, such maps were acquired for a cohort of subjects both before and after eccentric muscle damage. This study aimed to analyse the location and extent of the impact of muscle damage on muscle viscoelastic properties.

As elastography is an experimental measurement method, it was unknown what the stability and certainty levels of the data would be, and consequently what models would capture effects most robustly. We sought to generate a voxel-wise map of effects, but also investigated muscle-wise statistical models as they were expected to show increased robustness and statistical power due to pooling of data points.

Data Exploration

Data distributions

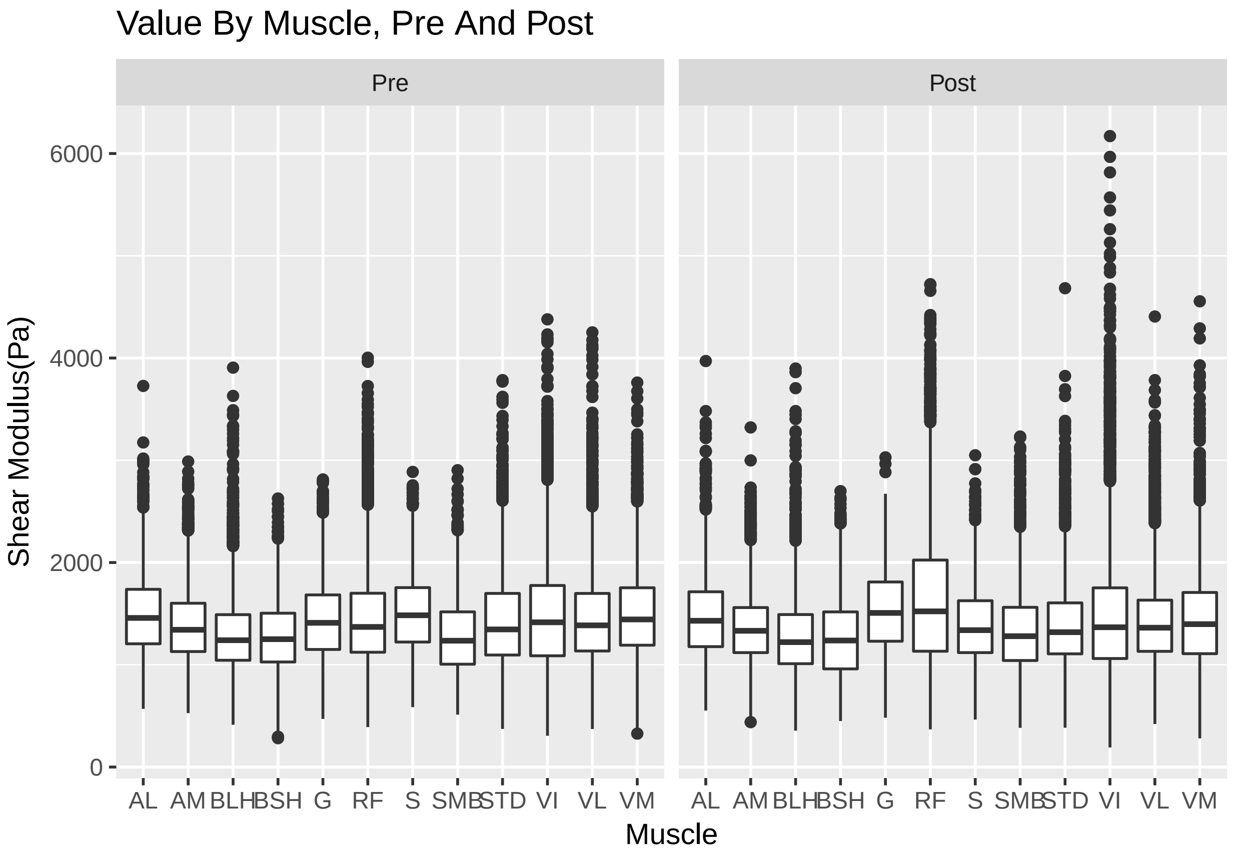
Data was assembled into tidy format from separate spreadsheets by subject and muscle.

```
source('muscle_load_clean_data.R')
data_tall <- load_clean_data(file.path(proj_path, data_path))
```

A boxplot of stiffness values by muscle group, shows heavy right skew, classic signs of a gamma or lognormal distribution. Comparing the groups pre- and post- shows that while the medians of the distributions remain relatively similar, for some groups (including the Vastus muscles) the skew increases greatly. This suggests that some of the effect mays be in the dispersion function of the distribution and not just the central tendency. We therefore probably want to model the shape function by muscle and condition, rather than just the scaling function (or its inverse, the rate function).

tab.1_cap <- fig_nums(name = "tab_1", caption = "German Bundesliga: Final Table 2015/16, Position 7-12") Fig. 1: Boxplots for stiffness (Pa) by muscle.

```
ggplot(data_tall) +
    geom_boxplot(aes(x=musc_id, y=value)) +
    facet_grid(. ~ cond_id) +
    labs(title="Value By Muscle, Pre And Post",x="Muscle",y="Shear Modulus(Pa)")
```



To evaluate whether a lognormal or gamma model would be more suitable, we look at the same boxplot of the log values: Fig. 2: Boxplots for log stiffness (Pa) by muscle.

```
ggplot(data_tall) +
    geom_boxplot(aes(x=musc_id, y=log(value))) +
    facet_grid(. ~ cond_id) +
    labs(title="Log Value By Muscle, Pre And Post",x="Muscle",y="Shear Modulus(Pa)")
```

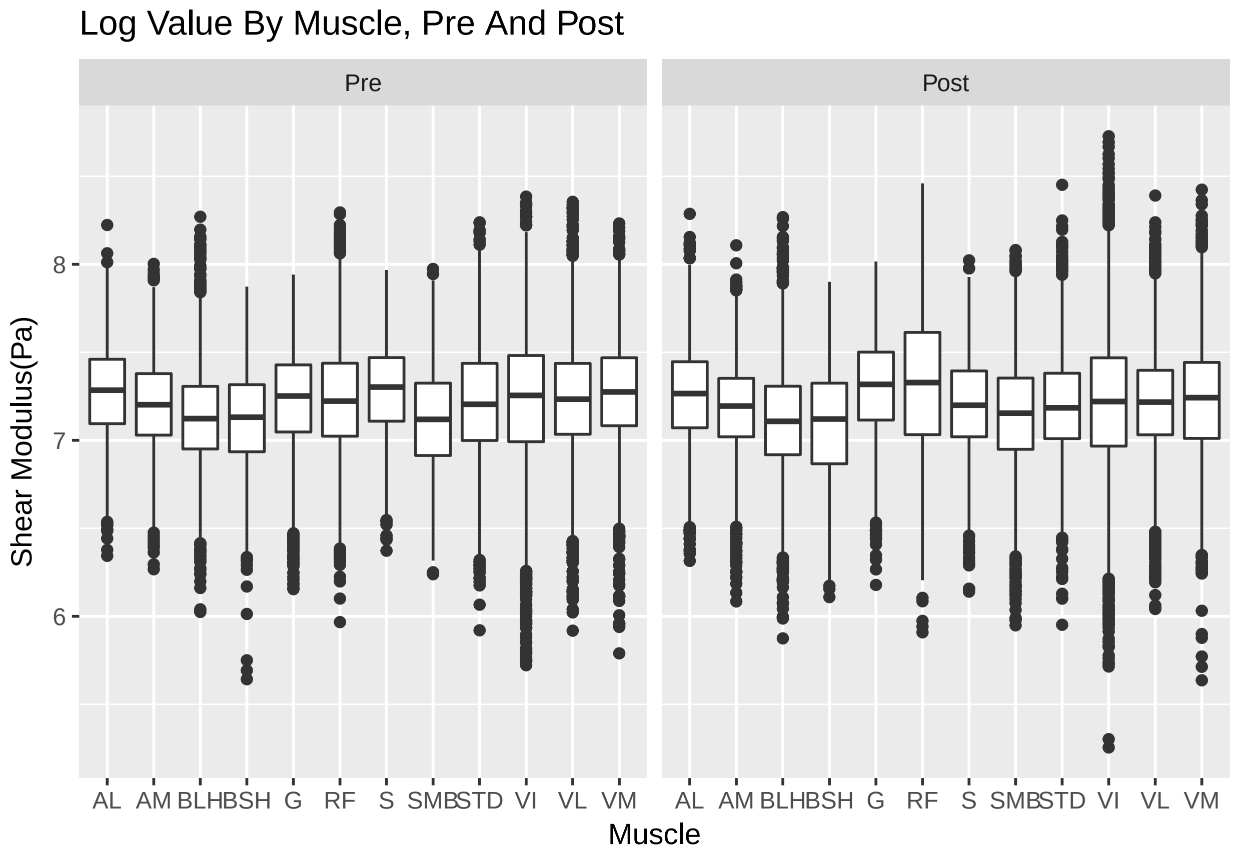


Fig. 2 shows that the skew ranges from roughly even (normal) but also many distributions skew left, with more extreme values on bottom than on top. As the log of a gamma skews left, gamma distributions will be used for the best fit. Normal models will also be run as a "reality check", because the immediate outputs are more intuitive, and we expect predicted values to be similar to the gamma outputs but have less overall goodness of fit.

Relationships between groups

Each data entry has a pixel-wise location, a muscle location, and a location within a muscle group. The above boxplots show differences between muscle medians that are credibly worth exploring. Below we look at the data by muscle group, again against the log value:

Fig. 3: Boxplots for log stiffness (Pa) by muscle group.

```
ggplot(data_tall) +
    geom_boxplot(aes(x=grp_id, y=log(value))) +
    facet_grid(. ~ cond_id) +
    labs(title="Log Value By Muscle, Pre And Post",x="Muscle",y="Shear Modulus(Pa)")
```

Fig. 3 shows similar trends. Medians, in particular of adductors and extensors, are quite similar, indicating that modelling an intercept by muscle group will likely produce uniqueness problems. However, the externsors show a clear change in dispersion that the adductors and flexors do not.

All three muscle groups show some left skew in the log plot, supporting use of the gamma distribution. Finally, we see slight change in dispersion of the function in adductors and hamstrings, but large increase in dispersion of the quadriceps.

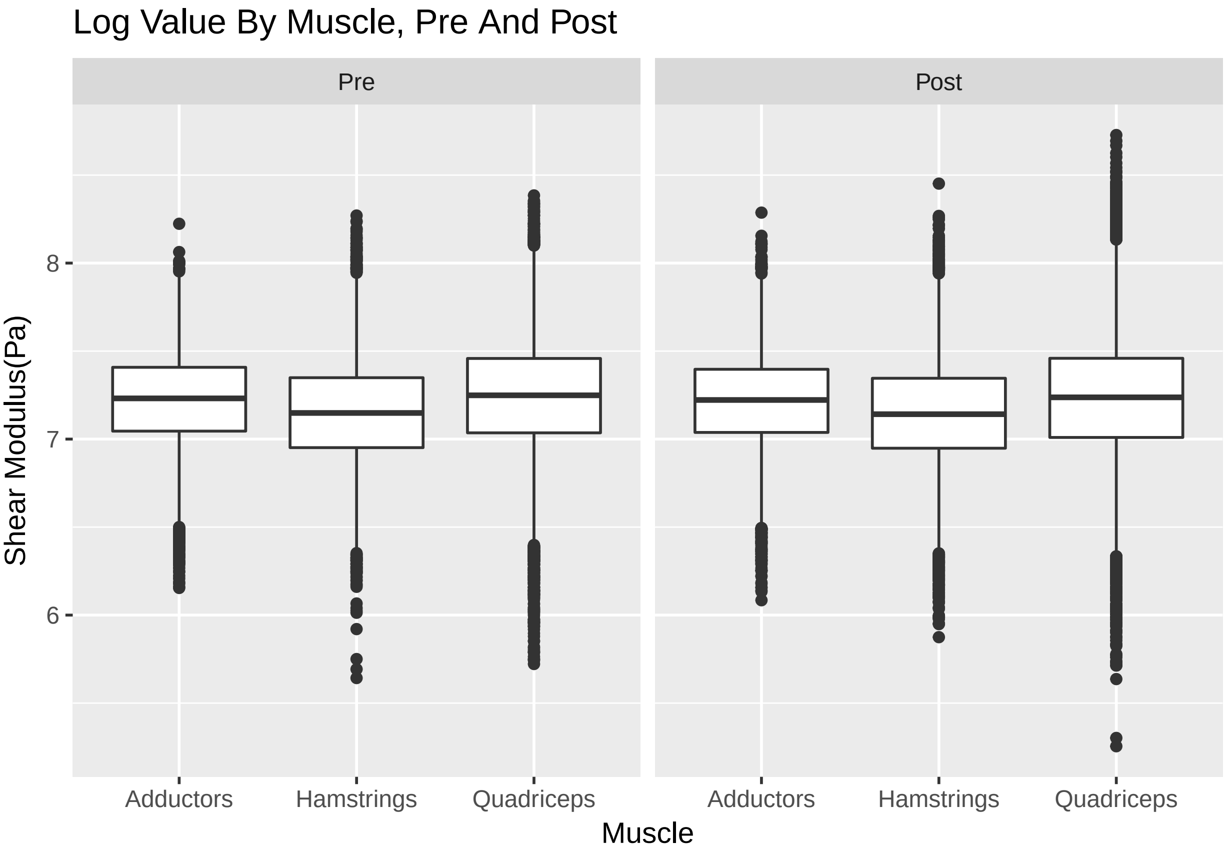


Figure 1: Fig. 3: Boxplots for log stiffness (Pa) by muscle group.

Data Modeling

The data exploration yielded the following conclusions:

1. Model the distributions with gamma functions
2. Intercept by muscle group is likely to be indeterminate. Intercept by muscle will be modeled but, as central tendency is small relative to dispersion, the model may be very uncertain
3. In the muscle-wise models, shape of distribution should be modeled by muscle 4.However, the pixel-wise maps should not model by muscle, to better catch focused local changes that drive the effect in terms of dispersion rather than central tendency

The third condition is slightly unusual, as the shape parameter of the Gamma distribution is often modeled as constant, but Bayesian analysis makes it straightforward to take this nuance into account. Essentially both models are attempts to model the dispersion effect, one with the shape of the value distribution within the muscle, one by modeling only on the pixel-wise level.

Our final analysis thus investigated two models:

- By Muscle:

1. Gamma, Rate By Muscle, Shape By Muscle

   $y_{ij} \sim Gamma(\alpha_j, \alpha_j/\gamma_j) log(\gamma_j) = \beta_j^0 + \beta_j^1 * I_{cond=POST} \alpha_j = \delta_j^0 + \delta_j^1 * I_{cond=POST} \beta^{0\ldots1} \sim \mathcal{N}(\mu^{0\ldots1}, (\sigma^{0\ldots1})^2) \beta^{0\ldots1} \sim exp(\lambda$

2. Gamma, Rate By Pixel, Rate By Muscle, Shape By Muscle

   $y_{ij} \sim Gamma(\alpha, \alpha/\gamma_j) log(\gamma_j) = \beta_j^0 + \beta_j^1 * I_{cond=POST} \alpha \sim \mathcal{U}(0.001, 100) \beta^{0\ldots1} \sim \mathcal{N}(\mu^{0\ldots1}, (\sigma^{0\ldots1})^2) card(j) = card(pixels)$

   Where y_{ij} is a draw from the distribution, α is the shape parameter, γ is the mean (and therefore $\frac{\alpha}{\gamma}$ is the rate parameter), the β variables are linear predictors with mean μ and variance σ^2, and I is an indicator function. Model 1 by muscle makes an unusual choice in modelling the shape function α, which needed to be modelled as the addition of two nonzero distributions. A sum of two uniform distributions would concentrate weight at the center of the prior distribution; consequently exponential priors were chosen as these would concentrate more weight in the lower, more probable values of the shape function prior.

Results - Model By Muscle - Iteration 1

Diagnostics

Gelman-Rubin diagnostics for variables β_0, β_1 and δ_0 were all less than the typical convergence standards of 1.03. The δ_1 variable was less stable, with diagnostic range of $1 - 04 - 1.12$. This suggests that the most challenging aspect of the data to model, the change of the shape function across conditions, requires either more samples or a different prior to capture the behavior. It is well known that thin tails can require many samples to accurately model.

Autocorrelation diagnostics were, similarly, low with β_0, β_1 and δ_0, with no correlation above ± 0.01 at lag 50; however the δ_1 variable had correlations from $0.07 - 0.12$ at lag 50.

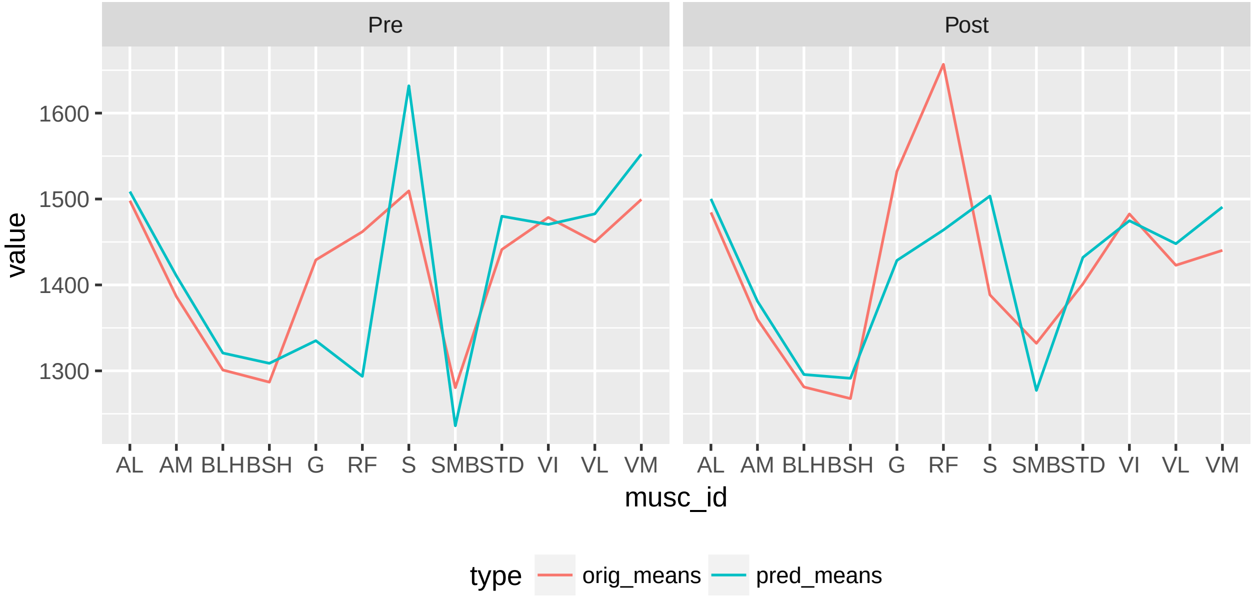
Conclusion for Iteration 1

This model struggled to converge or model the change in shape of the original data. Two likely ways to improve the model are to increase the number of samples, or to alter the prior on the change in shape function. For the second iteration, this second idea was tried first. The prior on δ_0 was kept as an exponential, as it is known to be positive and not high, but δ_1, as a change between two shapes, does not need this constraint. For the second iteration δ_1 was modeled as normal.

Iteration 2

Our second iteration, modelling δ_1 as normal, showed good convergence and diagnostics. Predictive modelling results are shown below:

```
source('mre_muscle_predict.R')
load("models_temp.Rdata")
means <- muscle_predict_original_means(data_tall, model_results$csim, T)
```



Here the modelling appeared to be a good predictor for all muscles except G, RF, and S, which I believe are the three smallest, so perhaps their distriibutions are the least stable.

Particularly of note are the values for the δ_1 functions that represent the change in shape. Here δ_0 and δ_1 are contrasted:

```
del0 <- model_results$csim[,which(grepl("delta0", colnames(model_results$csim)))]
print("Delta 0 means:")
```

```
## [1] "Delta 0 means:"
```

```
print(colMeans(del0))
```

```
##  delta0[1]  delta0[2]  delta0[3]  delta0[4]  delta0[5]  delta0[6]
##  14.791336  16.716234  14.014413  14.190466   9.413218  13.451758
##  delta0[7]  delta0[8]  delta0[9] delta0[10] delta0[11] delta0[12]
##  16.021566  15.632721   7.555830   8.812069  10.579620  14.125768
## delta0[13]
##  13.064298
```

```
del1 <- model_results$csim[,which(grepl("delta1", colnames(model_results$csim)))]
print("Delta 1 means:")
```

```
## [1] "Delta 1 means:"
```

```
print(colMeans(del1))
```

```
##  delta1[1]  delta1[2]  delta1[3]  delta1[4]  delta1[5]  delta1[6]
## -1.4629876 -0.1027771 -1.7533196 -2.1188435  1.8679864 -3.7824765
##  delta1[7]  delta1[8]  delta1[9] delta1[10] delta1[11] delta1[12]
## -1.0267528 -2.8502891  1.7088782 -1.2233085  0.3962779 -2.0974936
## delta1[13]
## -1.0387196
```

```
print("Pct change:")
```

```
## [1] "Pct change:"
```

```
print(colMeans(del1) / colMeans(del0))
```

```
##    delta1[1]    delta1[2]    delta1[3]    delta1[4]    delta1[5]
## -0.098908416 -0.006148339 -0.125108308 -0.149314582  0.198442914
##    delta1[6]    delta1[7]    delta1[8]    delta1[9]   delta1[10]
## -0.281188264 -0.064085669 -0.182328407  0.226166843 -0.138821941
##   delta1[11]   delta1[12]   delta1[13]
##  0.037456726 -0.148487043 -0.079508265
```

What this shows, is that the delta 1 values in the converged model are negative by a substantial percentage of the original shape function. That means that the shape is getting wider (as a higher shape value is a more sharply defined curve, and a low shape value is a very broad curve). And so I think this is a first sign of our finding.

To be continued…

1. Hirsch, S., Braun, J., and Sack, I. (2016). Magnetic resonance elastography: Physical background and medical applications (John Wiley & Sons).