Questions to answer in order to FINISH this manuscript:

\*Note: These questions are roughly listed in the order in which I will go about answering them.

1. What models do I currently have in my GIT repository?
   1. HINCHcalcium\_component.cellml
   2. Isometric\_Ca2.cellml
   3. Isometric\_protocol.py
   4. MeganModel\_workloop\_dynamicCai.cellml
   5. MeganModel\_workloop\_fixedCai.cellml
   6. MeganModel\_workloop\_isometricCai.cellml
   7. QuickReleaseProtocol.cellml
   8. RiceCaiTransient.cellml
   9. RiceTran\_workloop\_cycling\_dynamicCai.cellml
   10. RiceTran\_workloop\_cycling\_fixedCai.cellml
   11. Workloop\_protocol\_with\_passive\_cycling2.py
   12. Workloop\_protocol\_with\_passive\_cycling\_dynamicCai.py
   13. comparison\_figures.m
   14. main.py
   15. main\_dynamicCai.py
2. What NEW documents do I need to add to the above GIT repository to get the WL with isometric calcium to run to steady state? (17-5-2017)
   1. A new .py protocol that sets the value of “t” in RiceCaiTransient.cellml.
   2. A new version of RiceCaiTransient.cellml that has “t” as a parameter that can be SET instead of a value that is calculated by executing an equation. This will involve creating the necessary connections between python—MeganModel—CalciumModel. I will have to make changes to MeganModel\_workloop\_isometricCai.cellml—but these changes will be documented in GIT.

\*\*The most recent model version in this repository will be tailored to recreate the data in the manuscript. Thus, the MeganModel\_workloop\_isometricCai.cellml will be able to run with RiceCaiTransient.cellml ONLY. It will no longer be in the correct form use a piecewise calcium transient (which is inefficient anyway). To get BACK to a version that could handle a piecewise function I would need to look through the GIT history before 17-5-2017.

1. Figure out what version of the python protocol each contraction scenario currently uses (think about renaming the python protocol).
   1. The quick-release protocol is run in CellML not in OpenCMISS on hpc… so there is no python protocol that is associated with recreating the Kurihara quick-release results.
   2. The isometric contraction uses Isometric\_protocol.py
   3. The Work-loop with dynamic calcium uses main\_dynamicCai.py and Workloop\_protocol\_with\_passive\_cycling\_dynamicCai.py
   4. The Work-loop with isometric calcium uses main.py and Workloop\_protocol\_with\_passive\_cycling2.py
2. Can you create a map that shows how each of the models relate to each other? (17-5-2017)
3. Have you renamed the models so that their purpose is easy to understand?
4. How are you going to get the fitted RiceTran calcium transient to be cyclical so that is can run to steady-state?
5. Have you included the python optimisation code in the GIT repository, explaining how you got the parameters
6. Have you tried to get hpc to run the MATLAB plotting? (To make plotting faster)?
7. Have you renamed the Calcium component for the model that uses the Rice Calcium transient (the fitted isometric transient)
8. Have you verified that the XB model has the expected parameter values… and can you explain the parameter changes that you have made throughout your model coupling process? (ex. kn\_p and kp\_n, sigmap and sigman, TmpC, mass and viscosity, species param,
9. Have you run the WL simulation with the “matching” isometric calcium transient?
10. Which MeganModel version do you need to run in order to replicate the Kurihara quick-release data?
11. Do the models DO/ PRODUCE the results I expect them to?
12. Are your results repeatable?
13. Do you have ONE python interface from which you can recreate all data included in the Article?
14. I will also need a results folder in the GIT repository where data can be saved in an identifiable way and then retrieved by a code that will plot the figures.
15. Have you updated the other python protocol codes (other than Workloop\_protocol\_with\_passive\_cycling2.py) so that the criteria for starting passive restretch is Ca\_i[-1] <= 0.1 and time[-1] >= 500 ?