I have been working on the project Reassembly and Classification of Broken Bones and here are some summaries of my work:

I first worked on classification based on automatically collected data, using classical machine learning methods including random forest, supporting vector machine and K nearest neighbors, but the accuracy was only around 40%, so I temporarily paused my work waiting for more data. Then, I have been working on the manually collected data.

I first tried to use features from both fragment aspect and break curves aspect to predict break agent: Features of fragment aspect includes MaxDim, MLIntervals, Epiphysis, Breakquantity, PA\_NotchA\_Frag, PA\_NotchC\_Frag, PA\_NotchD\_Frag. Break curves aspect includes BKplane, Gon, VMfracAng. Extra tree method achieves best accuracy among all the classical methods I have tried, and it achieves 73% accuracy when classifying Effector (break agents) based on break curves without balancing number of break curves under each class. I have put slides of classification accuracy in the same folder, so that everyone could check on the accuracy of prediction on different labels other than Effector, such as Skelport, LPort, Species, SzCl.

Later, I made a voting algorithm to classify labels on fragment aspect, based on the voting result of break curves prediction. For example, if one fragment has 5 break curves, then we first predict labels on 5 break curves and then vote for the label of this one fragment. With voting algorithm, the prediction accuracy on fragment dropped down compare with the accuracy on break curves, it dropped to around 54%. I have made the script user friendly, and Alex could also try neural network without preprocessing manual data furthermore.

I also tried leave one out test, which is, using all the fragments as training data and leave one fragment as test data to predict on, however, the average accuracy over all fragments was still low, and remained around 54%.

Other than classical methods, I have also tried graph-based methods based on python packages provided by Jeff. The graph-based methods worked well on break curves aspect. I tried to predict effectors label. With balanced classes, 134 break curves in each class and 100 break curves labeled, the final prediction accuracy reached 79%. I haven’t tried voting algorithm on fragment aspect yet, but it seems to have a satisfying accuracy on break curves aspect.

I will try to simplify my code by implementing some pandas data frame code, and then work on the graph-based method.

The scripts can be found in the same folder, Data\_auto\_1out.py corresponds to leave one out test, Data\_auto4.py corresponds to classical method (Extra tree) on fragments, Data\_auto5.py corresponds to graph based methods on break curves, April 27 result summary.pptx corresponds to a summary of prediction over different labels.