

Distributed data analysis of Big Data and machine learning applied on a large number of detailed MOCCA numerical simulations

Arkadiusz Hypki, Warsaw, The Rubin-LSST Polish Consortium Annual Meeting 2024

2024.10.23

¹ Faculty of Mathematics and Computer Science of Adam Mickiewicz University

Globular clusters

Blue stragglers stars

Method

MOCCA code

BEANS code

Machine learning plugin for BEANS

Globular clusters

Globular clusters

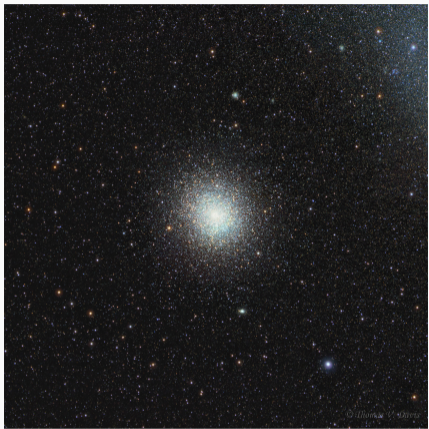


Figure 1: 47Tuc globular star cluster, one of the biggest and oldest in the Milky Way.

- very old (age comparable to the age of the Universe)
- size up to around 100 ly
- a core is clearly visible – best place for creating of many exotic objects: cataclysmic variables, X-ray binaries, black holes, intermediate-mass black holes, blue stragglers
- great laboratories for studying stellar evolution and dynamical interactions between stars
- Milky Way GCs: 50% GC within 5 kpc, the most distant 130 kpc

Dynamical modelling – importance

- may provide basic information to understand the **formation and then the evolution of exotic objects** within star clusters (e.g. hard binaries)
- dynamical interactions between stars may lead to **perturbations, disruptions, collisions and mass transfers** between stars
 - e.g. this may lead to decrease the semi-major axes and allow mergers which would now happen otherwise
 - may lead to formation of exotic binaries, supernova explosions (especially in the initial phase when many of massive stars are present), formation of black holes
- dynamical interactions in GCs may eject a lot of binaries that could be potential **sources of GWs**

What are blue stragglers?

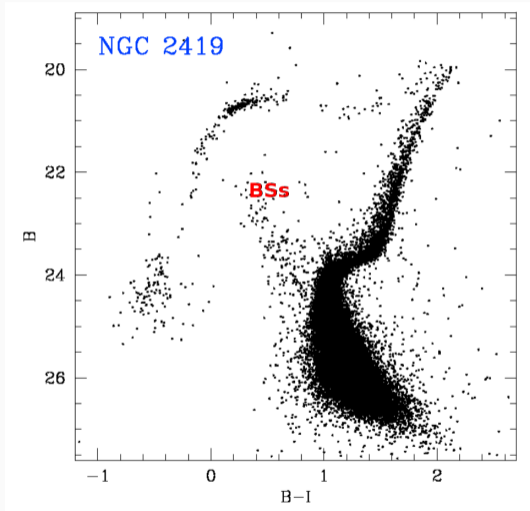


Figure 2: Example BSs in NGC2419

- BSs defined as stars which are brighter and bluer (hotter) than the main sequence turn-off point
- BSs lie along an extension of the main sequence in CMD
- it suggests that these objects got some additional mass
- BSs are present essentially in all star clusters

Two channels of formation: mass transfer and collisions

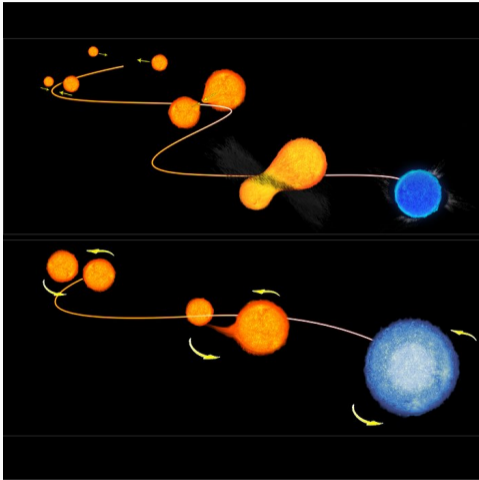


Figure 3: Mass transfer and collisional scenarios of BSs formation

- mass transfer (MT):
 - only for binaries (strong dependence on IMF)
 - BSs exceed only slightly turn-off (mostly)
 - MT leads to merger, which can create BSs too
- collisions (COLL):
 - dynamical interactions
 - important only for some star clusters

Method

MOCCA – features

- one of the most advanced codes for simulations of real-size star clusters
- based on Monte Carlo method (a few simplifications in comparison to N-body codes, e.g. one radial position)
- agrees very well with N-body codes (Wang et al. 2016)
- provides almost as much details about stars as N-body codes
- simulating the real clusters (M22, M4, 47Tuc etc.)
- exotic objects: blue stragglers, IMBHs, CVs...
- “observations” of simulations vs. real observations (COCOA)
- MOCCA can now handle **dynamical evolution of multiple population**
- very fast, which allows to test whole range of possible initial conditions (MOCCA-SURVEYs)
- data analysis with BEANS

BEANS code

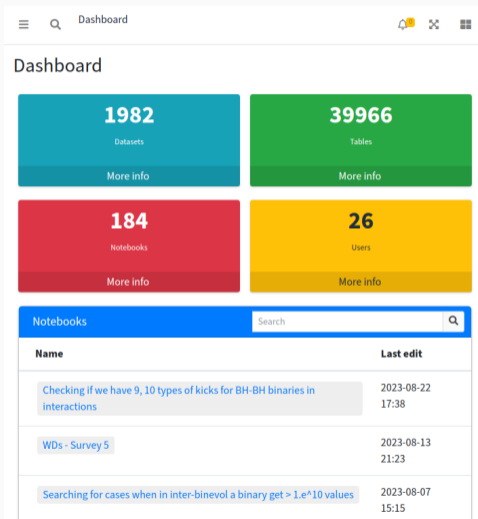


Figure 4: <http://BEANScode.net>

- interactive, distributed data analysis
- web-based
- open source
- data analysis in a form of notebooks (like Jupyter)
- Apache Pig (Apache Hadoop)
- connectors to MOCCA, NBODY codes
- Python, AWK, Gaia plugins
- **access to all simulations from all different mocca-survey from BEANS**
- motivation: ML plugin

BEANS code

The screenshot displays the BEANS MOCCA interface. On the left is a dark sidebar with navigation options: Dashboard, Notebooks (selected), New notebook, Datasets, Extras, Account, and a NOTEBOOK section with Edit and View options. Below these are ADMINISTRATION options: Administration and Diagnostics. The main area shows a notebook titled "Histories for all WDs (Survey5)". The notebook content includes a title "Collecting separate histories into one table", a toolbar, and a code editor with the following Pig code:

```
Collecting separate histories
Pig mode local
6 WDs6 = load 'datasets="Histories for all WDs Survey5" tables="Histories of all WDs escape Survey5 par
7
8 WDsAll = union ONSCHEMA WDs1, WDs2, WDs3, WDs4, WDs5, WDs6;
9
10 WDsAll = order WDsAll by dsid, id, outputId;
11
12 WDsAllGr = group WDsAll by id;
13 WDsAllCount = foreach WDsAllGr generate
14     group as id,
15     COUNT(WDsAll.id) as c;
16
17 store WDsAll into 'NAME "Histories of all WDs (Survey5 joined sorted)" ' using BeansTable();
```

Below the code editor is a section titled "Test plot with mass of the star id == 5746". The plot is titled "Mass of the object id 1063095" and shows a horizontal line at a mass of approximately 0.8, with a sharp drop-off at the end of the x-axis.

Figure 5: BEANS example notebook (computing histories for all WDs from all MOCCA simulations).

MOCCA-SURVEYs (Survey1, Survey2, and more models in progress)

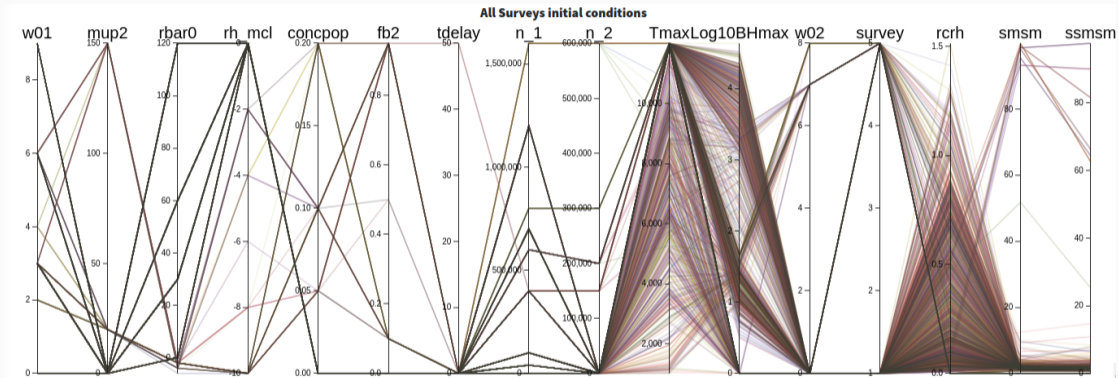


Figure 6: Grid of all MOCCA models from different MOCCA-SURVEY. All accessible from BEANS (<http://beans.moccode.net/>)

Milky Way coverage of initial conditions (mocca-survey-2)

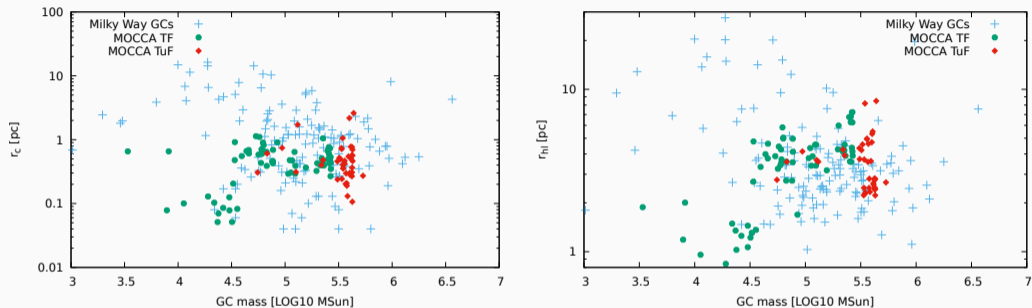


Figure 7: MOCCA simulation r_c , and r_{hl} coverage of Milky Way GCs. MOCCA simulations cover proper ranges of values of Milky Way GCs – it gives some confidence that the results of our work are well representing Milky Way GCs

Machine learning plugin for BEANS

Core collapse excess of blue stragglers number – 1 Gyr

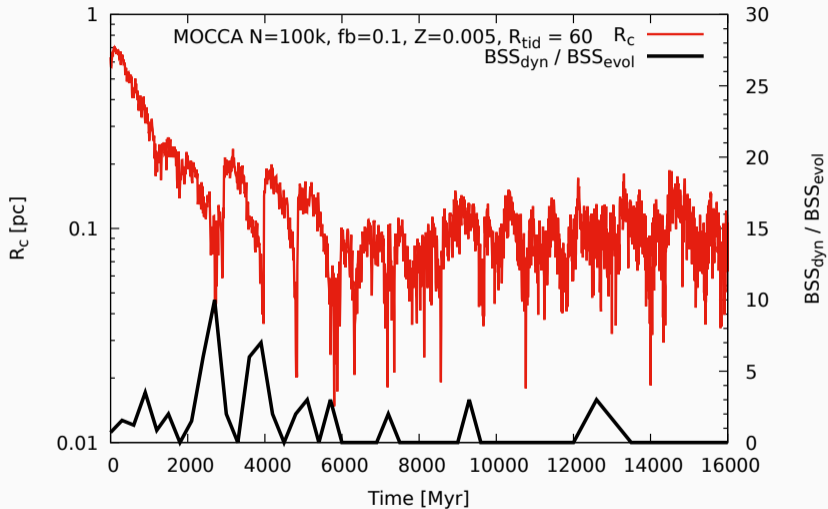


Figure 8: Core collapse vs. dynamical blue straggler excess

Core collapse excess of blue stragglers number – 3 Gyr

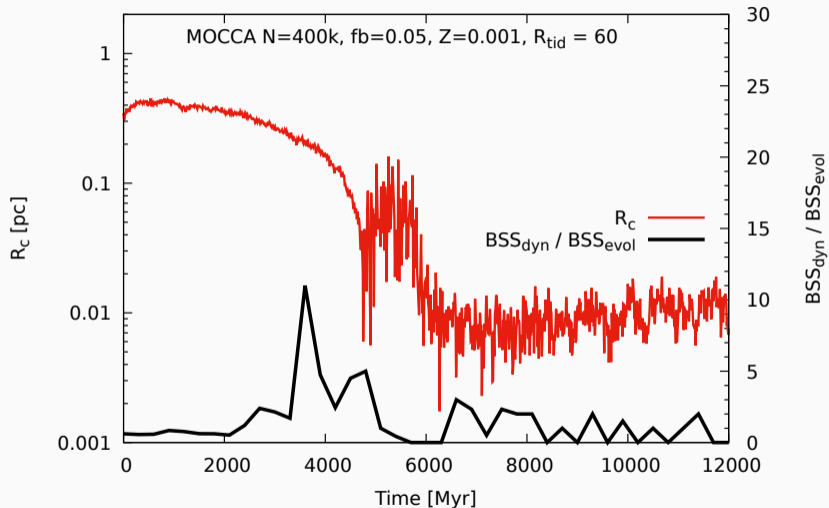


Figure 9: Core collapse vs. dynamical blue straggler excess

Core collapse excess of blue stragglers number – 6 Gyr

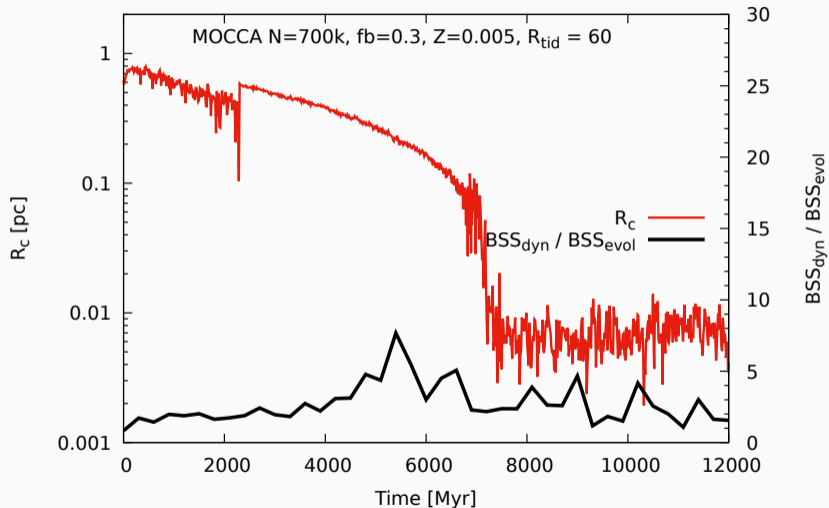


Figure 10: Core collapse vs. dynamical blue straggler excess

Core collapse excess of blue stragglers number – 11 Gyr

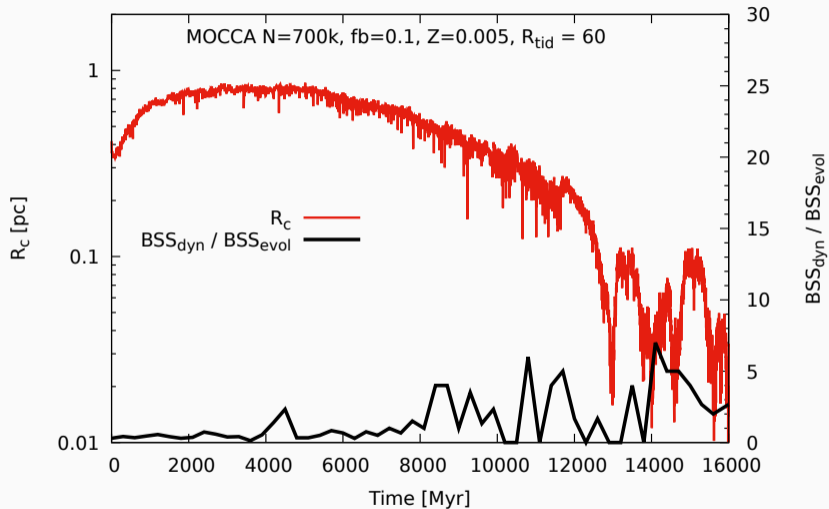


Figure 11: Core collapse vs. dynamical blue straggler excess

Core collapse excess of blue stragglers number for Milky Way

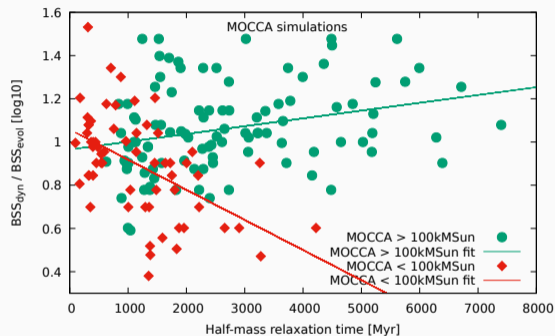


Figure 12: Dynamical BSSs to evolution BSSs fractions function of the half-mass relaxation time

- MOCCA simulations divided into two groups: more massive clusters (green points, $> 100kM_{\odot}$), and less massive clusters (red points, $< 100kM_{\odot}$)
- **low and high mass GCs have clearly different slopes for the excess of dynamical BSSs**
- motivation 1: ML to find the core collapse automatically

BEANS ML plugin

The screenshot shows the BEANS ML plugin interface. The left sidebar contains navigation options: Dashboard, Notebooks, Datasets, Extras, Account, and Administration. The main area is titled "Machine learning tests" and shows a configuration form for a new test. The form includes a title field, a "Training table(s)" section with "Data from datasets" (mach) and "from Tables" (train), a "Test table(s)" section with "Data from datasets" (machine) and "from Tables" (system d8cb7), and an "Output table" section. The "Column names for training" field is circled in orange and contains the text "tphys, smt, r1, rchut2, rhob, vc".

BEANS release
arkadiusz@hypki.net

Dashboard

Notebooks

Notebooks

New notebook

Datasets

Extras

Account

NOTEBOOK

Edit

View

Insert entry

ADMINISTRATION

Administration

Machine learning tests

New

Title

Predicting core collapse for one Survey1 mocca simulation

Training table(s)

Data from datasets

from Tables

mach

train

Column names for training (separated by comma)

tphys, smt, r1, rchut2, rhob, vc

Column name to predict

collapsed

Test table(s)

Data from datasets

from Tables

machine

system d8cb7

Column names for testing should be the same as for learning and should exist in the test tables

Output table

Output table name with predictions

collapsed predicted full system survey1

Column name with predictions

collapsedpred

- ML plugin added to BEANS
 - now we have access to our > 2000 MOCCA simulations
- currently we are using SCIKIT-LEARN
 - e.g. Random Forest Classifier
 - APACHE MAHOUT in plans
- one can easily define which column to use for learning which helps non-technical users
- the output are immediately accessible in BEANS for further analysis

Figure 13: BEANS ML plugin

Finding the core collapse time - 1. Using ML for predictions

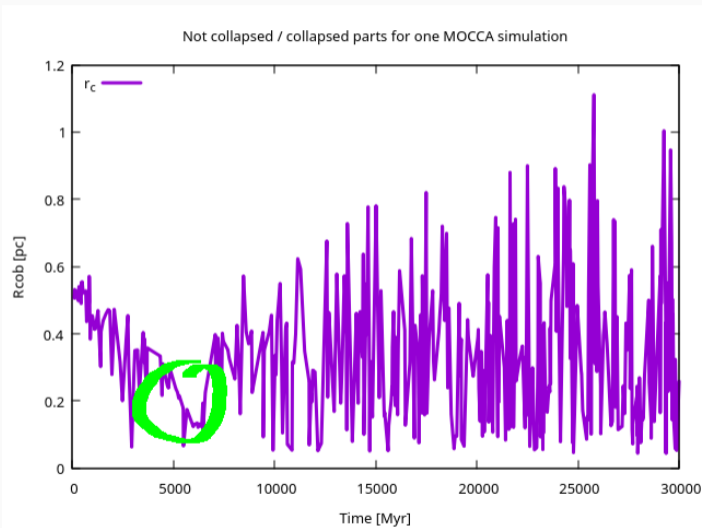
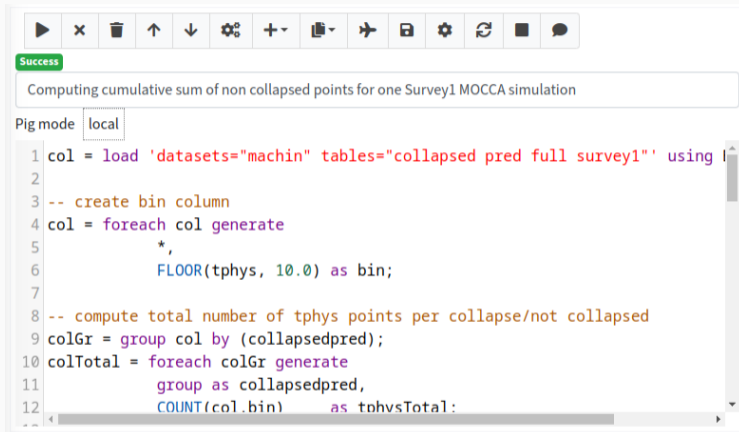


Figure 14: Finding core collapse in automatic/ML way

2. Computing cumulative distributions for predictions



```
Success
Computing cumulative sum of non collapsed points for one Survey1 MOCCA simulation
Pig mode local
1 col = load 'datasets="machin" tables="collapsed pred full survey1"' using l
2
3 -- create bin column
4 col = foreach col generate
5     *,
6     FLOOR(tphys, 10.0) as bin;
7
8 -- compute total number of tphys points per collapse/not collapsed
9 colGr = group col by (collapsedpred);
10 colTotal = foreach colGr generate
11     group as collapsedpred,
12     COUNT(col.bin) as tphysTotal;
```

Figure 15: Apache Pig computes cumulative distributions for all MOCCA simulations for collapsed and not collapsed points.

2. Computing cumulative distributions for predictions

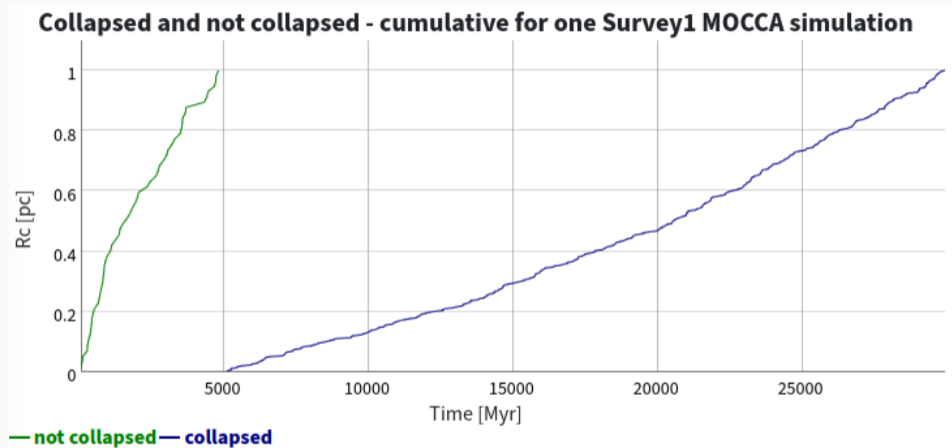


Figure 16: Cumulative plots showing collapsed and not collapsed parts for one MOCCA simulation. The core collapse is when not collapsed closes to 1.0, and not collapsed is still small.

Did it work?

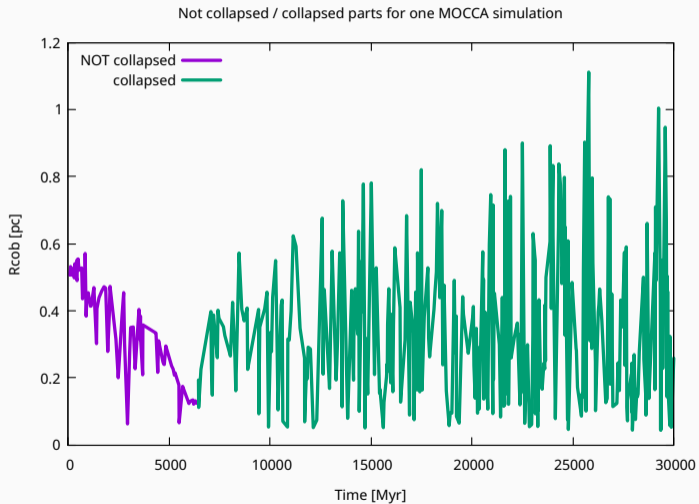


Figure 17: Core collapsed time found by ML

ML accuracy

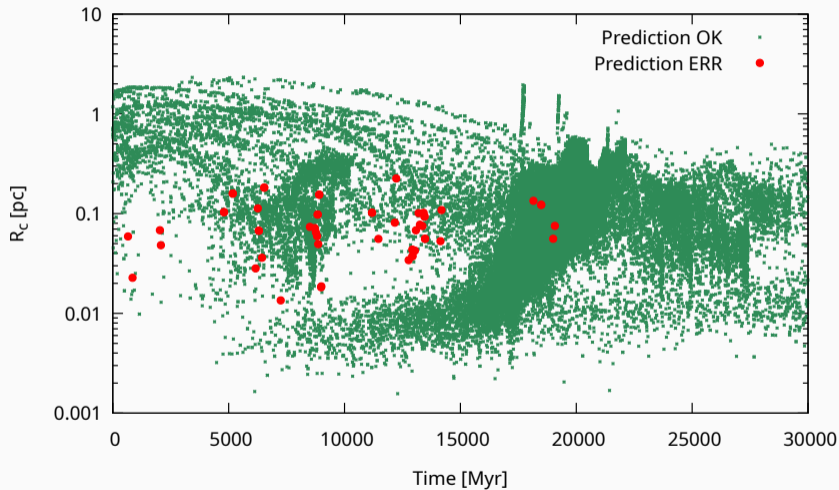


Figure 18: Qualitative ML accuracy for predicting core collapse for a few MOCCA simulations

Current step – testing different classifiers

Nearest Neighbors 'n_neighbors': 3: accuracy=68.59% precision=37.56% recall=49.07% train_time=18.12333s
predict_time=528.67222s

...

Decision Tree 'max_depth': 10: accuracy=67.49% precision=35.26% recall=44.43% train_time=18.15583s
predict_time=0.11338s

...

Random Forest 'max_depth': 10, 'max_features': 'sqrt', 'n_estimators': 10: accuracy=67.97% precision=36.28% recall=46.44%
train_time=102.94294s predict_time=1.12742s

...

**Naive Bayes 'var_smoothing': 1e-07: accuracy=95.29% precision=90.63% recall=89.36% train_time=0.99014s
predict_time=0.31822s**

QDA 'reg_param': 0.0: accuracy=85.84% precision=79.14% recall=54.67% train_time=2.08210s predict_time=0.48526s

...

Gradient Boosting 'learning_rate': 0.01, 'n_estimators': 50: accuracy=67.49% precision=35.26% recall=44.43%
train_time=1294.12376s predict_time=2.26932s

...

Next steps

- check different GCs parameters (or subset of them) to asses whether the predictions would be equally good
- check other ML classifiers:
 - Nearest Neighbors, Decision Tree, Random Forest (different params), Naive Bayes, QDA, Gradient Boosting
- future: use ML to predict CC, nCC, IMBH-GC, BHs-GC clusters



Figure 19: MOCCA, AMU,
NCN

- core collapse in GCs does increases the number of blue stragglers
- BEANS – it is a nice cool toy which allow us to do the full data analysis (+ML) on TBs of data from one place
- machine learning is unbelievable powerful
 - machine learning can automatize many efforts really easily
 - it can be actually easy applied

Arkadiusz Hypki — ahypki@amu.edu.pl — MOCCAcode.net — BEANScode.net