The deep-learning algorithm π-DOC

Predicting dynamical mass, age and distance of globular clusters

Chardin & Bianchini 2021, MNRAS, 504, 5656
Version 0.1

Paolo Bianchini, December 2022

1. Overview of π-DOC

The main task of π-DOC is to predict the mass distribution of a globular cluster from its observed photometry. Using the same input, π-DOC additionally estimates the age and the distance of the globular cluster. The current version of the algorithm (version 0.1) is described in Chardin & Bianchini 2021 (MNRAS, 504, 5656; hereafter C&B21).

π-DOC is composed of two neural networks: 1) a convolutional encoder-decoder (CED) to predict the mass distribution of a cluster, and 2) a convolutional neural network (CNN) to predict the distance and the age of the cluster. The input is a flux map of a cluster in the V-band in a field-of-view of 40 arcsec x 40 arcsec (160 pixel x 160 pixel) and a pixel scale of 0.25 arcsec.

π-DOC is currently trained on a set of mock globular clusters observations derived from direct N-body simulations (see C&B21, Section 2). These simulations reproduce the realistic effects of dynamical and stellar evolution of typical globular clusters and the typical observational effects (e.g. seeing effects) treated in a forward modelings way. However, the simulations have a number of stars of $N=5\times10^4$, lower than the one of typical globular clusters. For this reason, at this stage, the applicability of π-DOC to observations is limited to low-luminosity clusters, as carefully discussed in Section 5 and 6 of C&B21. Future versions of the algorithm will allow for a much more general applicability of the algorithm to observed clusters.

The example provided here consists of predicting the mass, age and distance of 100 mock observations of simulated globular clusters. This is a subsample of the test set used to verify the performances of π-DOC, described in Section 2.3 of C&B21. With this code one can reproduce the equivalent of Figures 4, 5, 6, 8 and 9.

The algorithm was built and trained using the Keras API, the high-level API of TensorFlow. The code is written in Python3 language.

2. Structure of the code

2.1 Architecture of the directory

PiDOC_CB21/
- **trained_piDOC/**
- **input_files/**
  - **standardization_files/**
- **predictions/**

The main code **piDOC_predictions.py** is contained in the directory **PiDOC_CB21/**, together with the file **functions.py** containing the necessary functions. The files containing the trained algorithm π-DOC (CED and CNN) are in the directory **trained_piDOC/.** The input files and the output files are in the directories **input_files/** and **predictions/**, respectively.

2.2 piDOC_predictons.py

The main code is structured in two parts. The first part uses the routine *make_predictions()* to predict mass, age and distance from the input, and the routine *display_mosaic_prediction()* to visualize the results. The second part uses a series of routines to plot the comparison between the predicted properties and the expected ones for all the maps in the input and it calculates the statistics of the performances of the algorithm. All these routines are in the file **functions.py** and are described below:

FIRST PART:

make_predictions(pathtrained,pathinput,pathprediction,L_testset)

Make the mass predictions using the CED and distance and age prediction using the CNN, starting from the input flux maps.

—> Inputs

pathtrained: path of the trained models
pathinput: path of the input maps
pathprediction: path of the directory in which to save the predictions
L_testset: N flux maps in V-band (each map is 160 pixel x 160 pixel; pixel scale 0.25 arcsec)

—> Outputs

predictionM.npy: predicted mass maps (same format as input flux maps)
predictiondistance_age.npy: predicted distance and age for the N maps

display_mosaic_prediction(pathimage,L_testset,M_testset,distanceage_testset,prediction_M,prediction_distanceage)

Plot a mosaic of 4 cluster maps selected randomly, comparing the predictions to the expected results. Equivalent to Figure 4 in C&B21.

—> Inputs

pathimage: location of the output figure
L_testset: N flux maps in V-band (each map is 160 pixel x 160 pixel; pixel scale 0.25 arcsec)
M_testset: true mass maps corresponding to the L_testset
distanceage_testset: true distances and ages corresponding to the maps in L_testset
prediction_M: mass maps predicted by π-DOC *make_predictions()*
prediction_distance_age: distances and age predicted by π-DOC *make_predictions()*

—> Outputs

figure: pathimage/mosaic_predicitons_maps_a-b-c-d.png

SECODN PART:

display_predicted_vs_true_mass(pathimage,M_testset,prediction_M)

Plot the pixel-by-pixel predicted mass vs. true mass of the entire sample. Equivalent to Figure 5 in C&B21.

—> Inputs

pathimage: location of the output figure

M_testset: true mass maps (each map is 160 pixel x 160 pixel; pixel scale 0.25 arcsec)

prediction_M: mass maps predicted by π-DOC *make_predictions()*

—> Outputs

figure: pathimage/Mass_map_true_vs_predicted.png

table: statistics of the predictions as defined in Sect. 4 of C&B21

pathimage/predicted_vs_true_mass.txt

display_predicted_vs_true_total_mass(pathimage,M_testset,prediction_M)

Plot the predicted total masses vs. true total masses of the entire sample. Equivalent to Figure 6 in C&B21

—> Inputs

pathimage: location of the output figure

M_testset: true mass maps (each map is 160 pixel x 160 pixel; pixel scale 0.25 arcsec)

prediction_M: mass maps predicted by π-DOC *make_predictions()*

—> Outputs

figure: pathimage/Total_mass_map_true_vs_predicted.png

table: statistics of the predictions as defined in Sect. 4 of C&B21

pathimage/predicted_vs_true_total_mass.txt

display_predicted_vs_true_age(pathimage,distanceage_testset,distanceage_prediction)

Plot the predicted ages vs. true ages of the entire sample. Equivalent to Figure 8 in C&B21

—> Inputs

pathimage: location of the output figure

distanceage_testset: true distances and ages corresponding to the maps in L_testset

distanceage_prediction: predicted distances and ages by π-DOC *make_predictions()*

—> Outputs

figure: pathimage/Age_true_vs_predicted.png

table: statistics of the predictions as defined in Sect. 4 of C&B21

pathimage/predicted_vs_true_age.txt

pdf_distances(pathimage,distanceage_testset,distanceage_prediction)

Plot the predicted ages vs. true ages of the entire sample. Equivalent to Figure 8 in C&B21

—> Inputs

pathimage: location of the output figure

distanceage_testset: true distances and ages corresponding to the maps in L_testset

distanceage_prediction: predicted distances and ages by π-DOC *make_predictions()*

—> Outputs

figure: pathimage/Pdf_distances.png

table: statistics of the predictions as defined in Sect. 4 of C&B21

pathimage/distance_predictions_stats.txt

2.3 Inputs

The input files in the input_files/ directory are a subsample of 100 maps extracted from the test set used in C&B21. They are issued from simulated globular clusters from which we created mock V-band observations with their associated dynamical mass maps, distances and ages (see Section 2 C&B21). Each map is 160 pixel x160 pixel (pixel scale 0.25 arcsec), the distances are 15, 30, 60 ,80 kpc, and the ages range from 0 to 14 Gyr. The flux maps, mass maps, distance and age information are in the following files:

Luminosity_V_testset.npy
mass_testset.npy
distance_age_testset.npy

The directory input_files/standardization_files/ contains the values needed to "standardize" the input values in the same way done during the training process of π-DOC(see Section 2 C&B21).

2.4 Outputs

The directory predictions/ contains all the outputs of π-DOC from the *make_predictions()* routine:

predictionM.npy: 100 predicted mass maps (same format as input flux maps)
predictiondistance_age.npy: predicted distances and ages for the 100 maps

Moreover, this directory contains all the figures and tables produced by the other routines.

3. How to install and run the code

3.1 Requirements

The code is written in Python3 and requires the following modules:
`numpy, matplotlib, tensorflow, tabulate`

3.2 Installation

Download the piDOC_v0.1.zip and unzip it:

```
unzip piDOC_v0.1.zip
```

Install the needed modules:

```
pip install matplotlib
pip install tensorflow
pip install tabulate
```

3.3 Running the code

Run the code with the command

```
python3 piDOC_predicitons.py
```

The mass, distance, and age predictions take a few seconds to complete, for the entire sample of 100 maps provided.

4. Examples of outputs

Below we show the figures that are produced by the code, specifying the name of the routine that produces them, and their corresponding figure in C&B21.

Figure: `mosaic_predicitons_maps_a-b-c-d.png`
Routine: *`display_mosaic_prediction()`*
Corresponding to Figure 4 in C&B21

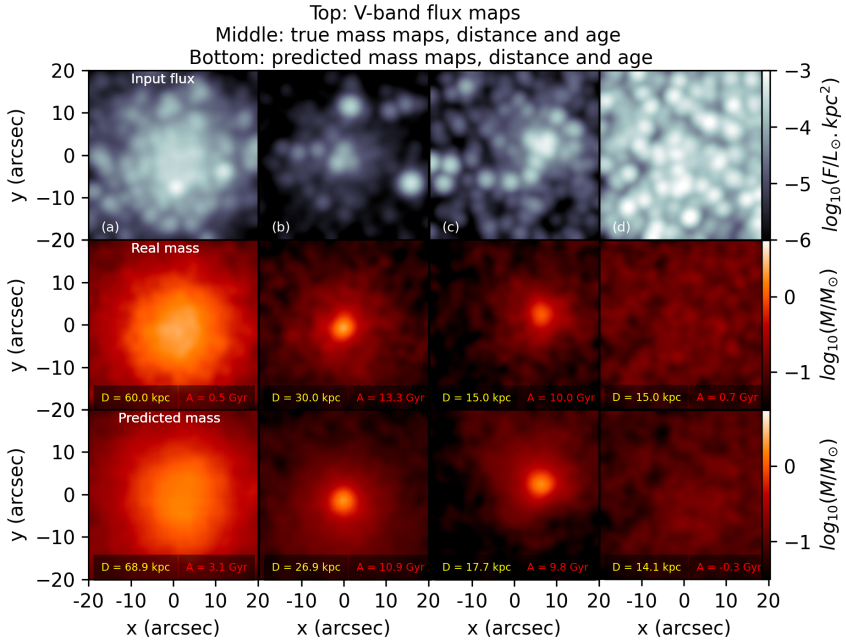


Figure: `Mass_map_true_vs_predicted.png`
Routine: *`display_predicted_vs_true_mass()`*
Corresponding to Figure 5 in C&B21

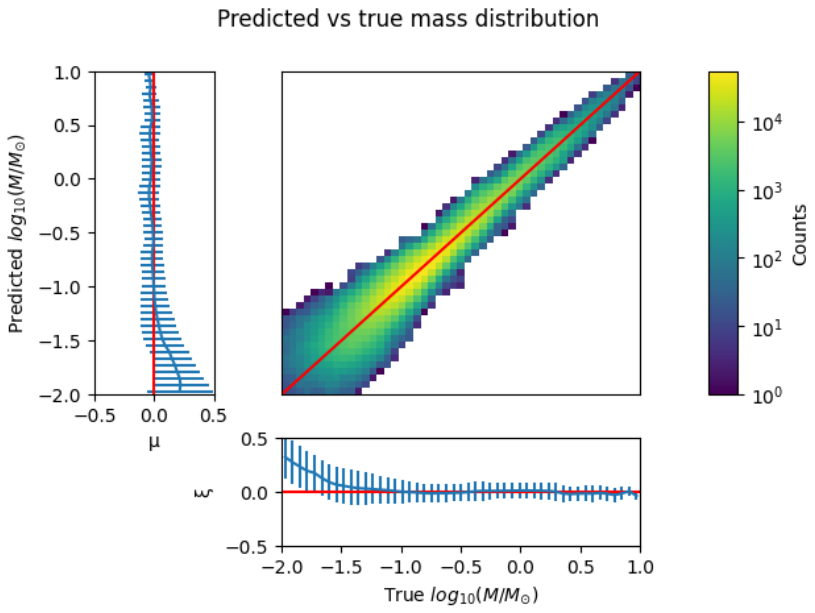


Figure: `Total_mass_map_true_vs_predicted.png`
Routine: *`display_predicted_vs_true_total_mass()`*
Corresponding to Figure 6 in C&B21

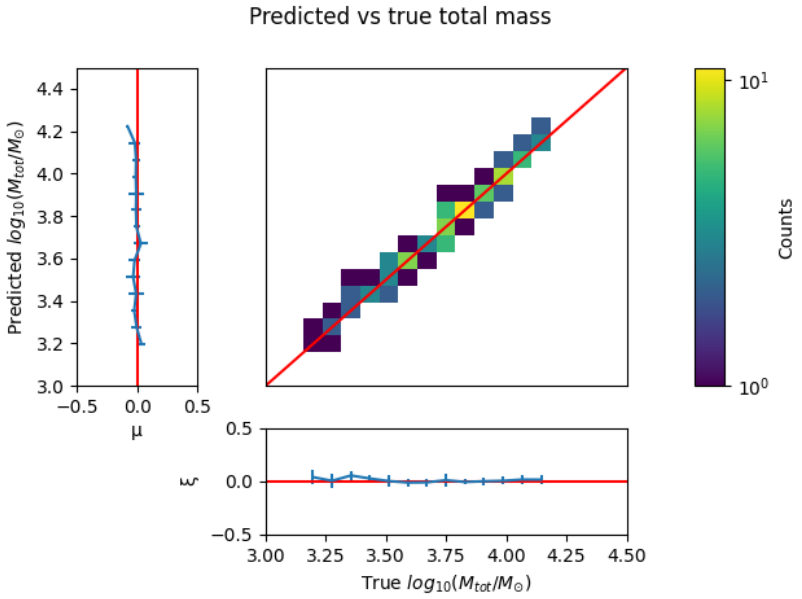


Figure: `Age_true_vs_predicted.png`
Routine: *`pdf_distances()`*
Corresponding to Figure 8 in C&B21

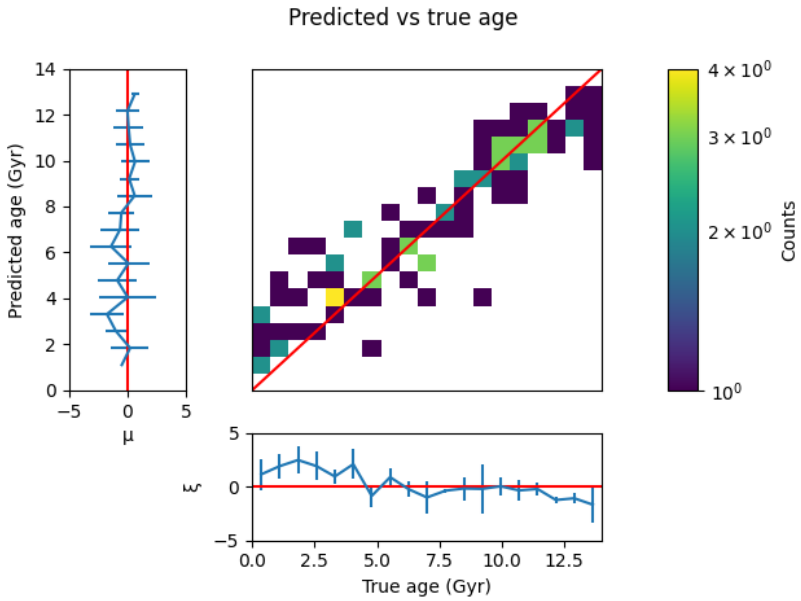
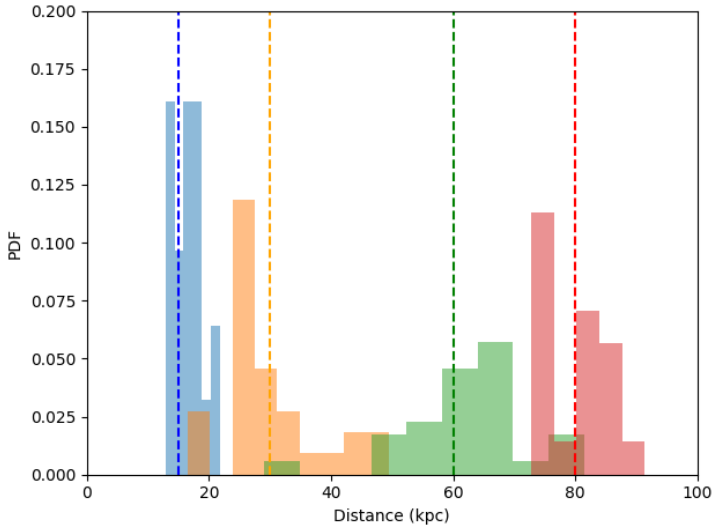


Figure: `Pdf_distances.png`
Routine: *`display_predicted_vs_true_age()`*
Corresponding to Figure 8 in C&B21



Note that to exactly reproduce the figures from C&B21 the entire test set with 8172 maps is needed. Here the example is based on a subsample of 100 maps only.