GAME
(GAlaxy Machine learning for Emission lines)

mantained by
Graziano Ucci*

March 22, 2019

1 Introduction

GAME is a self-contained and user-friendly program written in PYTHON, available online at **https://game.sns.it**. It allows the user to infer different ISM physical properties (density n, column density N_H, far-ultraviolet -FUV, 6-13.6 eV- flux G, ionization parameter U, metallicity Z, escape fraction f_{esc}, optical extinction A_V) from the emission lines intensities in galaxy spectra. The code is based on the Supervised Machine Learning (ML) algorithm AdaBoost with Decision Trees as base learner. This ML algorithm is trained with a large library of synthetic spectra (50,000 CLOUDY[1] photoionizations models). The total wavelength coverage of the synthetic spectra ranges from the Lyα (1216 Å) up to the far infrared (1 mm). However, GAME is able to deal with any subset of emission lines within the input spectra. The code is fully described in Ucci et al. (2017) and Ucci et al. (2018). We would appreciate that if you use GAME in preparing a scientific publication, you **cite it using the following BibTex:**

```
@ARTICLE{Ucci2017,
author = {{Ucci}, G. and {Ferrara}, A. and {Gallerani}, S. and {Pallottini}, A.},
title = "{Inferring physical properties of galaxies from their emission-line spectra}",
journal = {\mnras},
archivePrefix = "arXiv",
eprint = {1611.00768},
keywords = {methods: data analysis, ISM: general, ISM: H II regions, ISM: lines and bands,
galaxies: ISM},
year = 2017,
month = feb,
volume = 465,
pages = {1144-1156},
doi = {10.1093/mnras/stw2836},
adsurl = {http://adsabs.harvard.edu/abs/2017MNRAS.465.1144U},
adsnote = {Provided by the SAO/NASA Astrophysics Data System}
}

@ARTICLE{Ucci2018,
author = {{Ucci}, G. and {Ferrara}, A. and {Pallottini}, A. and {Gallerani}, S.},
```

*graziano.ucci@sns.it

[1] https://www.nublado.org/

```
title = "{GAME: GAlaxy Machine learning for Emission lines}",
journal = {\mnras},
archivePrefix = "arXiv",
eprint = {1803.10236},
keywords = {methods: data analysis, galaxies: ISM, ISM: general},
year = 2018,
month = jun,
volume = 477,
pages = {1484-1494},
doi = {10.1093/mnras/sty804},
adsurl = {http://adsabs.harvard.edu/abs/2018MNRAS.477.1484U},
adsnote = {Provided by the SAO/NASA Astrophysics Data System}
}
```

The present README provides a brief description and a tutorial on the usage of GAME.

2 Run the code

GAME can be executed directly online by uploading two input files containing emission line data, as detailed below, and choosing a list of emission line names. In order ro tun GAME, just go to the web page: **https://game.sns.it**, and follow these steps:

1. Upload a file containing the emission line intensities (Sec. 2.1.3);
2. Upload a file containing the errors associated to the emission line intensities (Sec. 2.1.3);
3. Select input line names from the menu. Here the user defines which line transition corresponds to which column of the input files (Sec. 2.1.4);
4. Select the physical properties to compute;
5. Specify if optional files must be provided at the end of the calculations (see Sec. 3.1);
6. Insert a valid email address. After the calculation is performed, an automatic notification will be sent via mail. The mail will contain a link to download the outputs.

2.1 User input files

The input files for GAME must be two **tab-** or **space-separated ascii files**. In the first row of both files, the user must provide the wavelength of each emission line (air or vacuum). Therefore, each column corresponds to a different line transition. Each row (starting from the second one) represent a different input spectrum.

For the units, there is no need to choose a specific one (e.g. erg s^{-1} cm^{-2}), provided they are consistent between all the data and their errors. This is because the input data are re-normalized within the code itself. For the same reason also emission line fluxes relative for example to Hβ line are acceptable.

The input line intensities and their associated errors **must NOT be corrected for reddening** because GAME automatically takes into account extinction.

In the following sections there are reported two simple examples of possible input files[2]. For the sake of clarity, these examples include also missing values and upper limits. Note that we will consider in the following as a missing value an emission line in the input spectrum for which there are **neither measurements nor upper limits**. Specific flags must be provided in the input files for missing values and upper limits.

2.1.1 Flag for missing values

If the user wants to exclude a specific emission line intensity, i.e. there are neither measurements nor upper limits, it is necessary to put the intensity of this line and the associated error as **zero**.

2.1.2 Flag for upper limits

If the user wants to indicate that a given emission line intensity value is an upper limit, it is necessary to put the intensity of this line equal to **zero** and the associated error as **-99**.

2.1.3 Example of input files

Let us have 6 spectra (rows in the input files) of the following lines (columns in the input file): [O II] 3726Å, [O II] 3729Å, [Ne III] 3869Å, H-delta 4102Å, H-gamma 4340Å, [O III] 4363Å, H-beta 4861Å, [O III] 4959Å, [O III] 5007Å, He I 5876Å, [O I] 6300Å, [N II] 6548Å, H-alpha 6563Å, [N II] 6584Å, [S II] 6716Å, [S II] 6731Å, [Ar III] 7135Å. This is the input file structure:

Emission line intensities

3726	3729	3869	4102	4340	4363	4861	4959	5007	5876	6300	6548	6563	6584	6716	6731	7135
6.71	6.23	9.35	1.22	2.20	**0.00**	4.64	2.51	7.55	5.62	3.74	9.01	1.33	2.66	5.99	6.90	3.68
6.61	6.22	9.30	1.10	1.98	**3.00**	4.00	3.51	7.65	5.50	3.74	9.00	1.20	2.65	6.01	6.95	4.00
6.51	6.32	9.20	1.00	2.10	**0.00**	4.30	3.61	7.65	5.50	**0.00**	9.10	1.25	3.01	6.25	6.97	4.01
6.40	6.30	9.00	1.01	2.01	**0.00**	5.10	4.01	8.01	**0.00**	**0.00**	8.65	2.15	3.18	6.37	6.64	4.10
6.41	6.42	9.40	1.27	2.15	8.00	4.34	3.62	7.00	5.14	4.02	9.00	1.20	2.99	5.99	6.98	3.94

Errors on emission line intensities

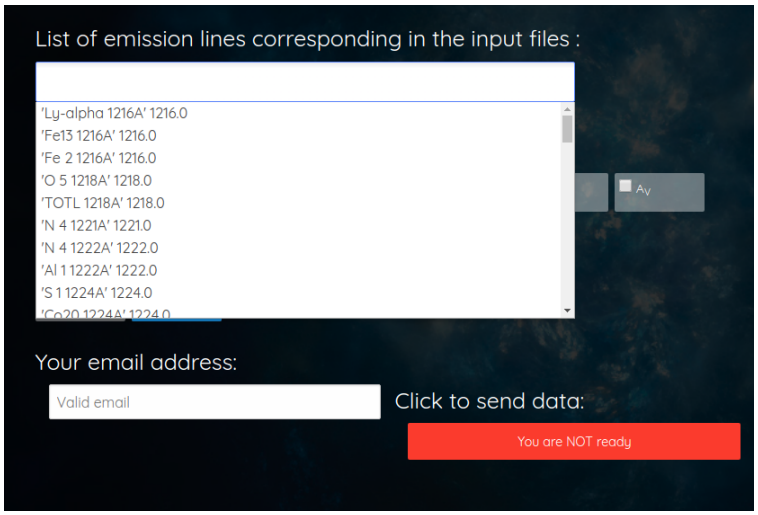
3726	3729	3869	4102	4340	4363	4861	4959	5007	5876	6300	6548	6563	6584	6716	6731	7135
0.67	0.62	0.93	0.12	0.22	**0.00**	0.46	0.25	0.75	0.56	0.37	0.90	0.13	0.26	0.59	0.69	0.36
0.66	0.62	0.93	0.11	0.19	**-99**	0.40	0.35	0.76	0.55	0.37	0.90	0.12	0.26	0.60	0.69	0.40
0.65	0.63	0.92	0.10	0.21	**0.00**	0.43	0.36	0.76	0.55	**0.00**	0.91	0.12	0.30	0.62	0.69	0.40
0.64	0.63	0.90	0.10	0.20	**0.00**	0.51	0.40	0.80	**0.00**	**0.00**	0.86	0.21	0.31	0.63	0.66	0.41
0.64	0.64	0.94	0.12	0.21	0.80	0.43	0.36	0.70	0.51	0.40	0.90	0.12	0.29	0.59	0.69	0.39

Note that in this example:
- 1st spectrum has a missing value for [OIII] λ4363;
- 2nd spectrum has an upper limit for [OIII] λ4363;
- 3rd spectrum has a missing value for [OIII] λ4363 and [OI] λ6300;
- 4th spectrum has a missing value for [OIII] λ4363, HeI λ5876 and [OI] λ6300;
- 5th spectrum has all lines available.

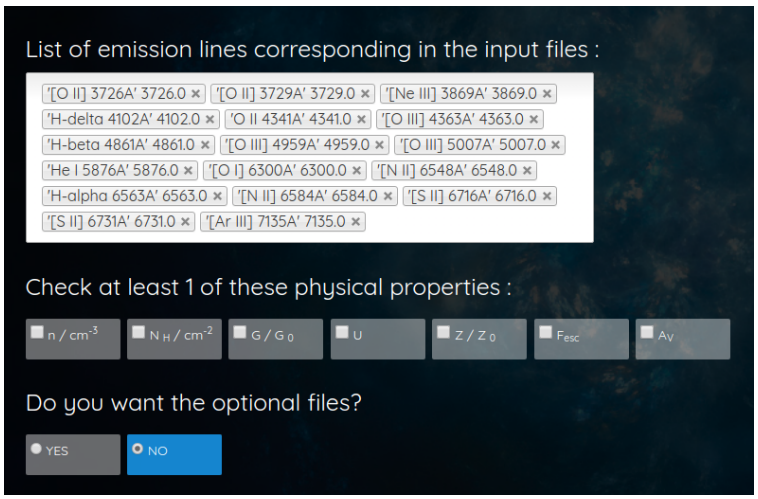
[2]The numbers reported in this README are purely illustrative to give a full description of the GAME features.

2.1.4 Emission lines labels

Once the user has uploaded the files, it is necessary to choose from a specific list what is the transition corresponding to each column of the input files. This can be done choosing from the drop-down menu:



For the example reported in Sec 2.1.3, this should be the final result:



2.1.5 Unique models

The code computes the number of unique models (i.e. unique combinations of input emission lines) from the input file. In the example reported in Sec. 2.1.3, there are four different unique models:

1. includes all the lines except [OIII] λ4363 (1st row in Sec. 2.1.3);
2. includes all the lines (2nd and 5th row in Sec. 2.1.3);
3. includes all the lines except [OIII] λ4363 and [OI] λ6300 (3rd row in Sec. 2.1.3);

4. includes all the lines except [OIII] λ4363, HeI λ5876 and [OI] λ6300 (4th row in Sec. 2.1.3);

3 Output files

At the end of the calculation, GAME produces the following output files:

- a *log file* named "model_ids.dat" with the details of each unique model. For each of them, (see Sec. 2.1.5), this file reports the following information:

```
################################
Id model: 2
Standard deviation of log(G0): 0.600
Standard deviation of log(n):  0.642
Standard deviation of log(NH): 0.446
Standard deviation of log(U):  0.819
Standard deviation of log(Z):  0.177
Standard deviation of log(Av):  0.893
Standard deviation of log(fesc):  0.778
Cross-validation score for G0: 0.897 +- 0.002
Cross-validation score for n:  0.879 +- 0.006
Cross-validation score for NH: 0.925 +- 0.004
Cross-validation score for U:  0.799 +- 0.003
Cross-validation score for Z:  0.974 +- 0.003
Cross-validation score for Av:   0.837 +- 0.013
Cross-validation score for fesc: 0.897 +- 0.018
List of input lines:
['[O II] 3726A' '[O II] 3729A' '[Ne III] 3869A'
'H-delta 4102A' 'H-gamma 4340A' '[O III] 4363A'
'H-beta 4861A' '[O III] 4959A' '[O III] 5007A'
'He I 5876A' '[O I] 6300A' '[N II] 6548A'
'H-alpha 6563A' '[N II] 6584A' '[S II] 6716A'
'[S II] 6731A' '[Ar III] 7135A']
################################
```

  "Standard deviation" and "Cross-validation score" are two indicators of the overall accuracy of the code (for details we refer to Appendix A2 of Ucci et al. 2018); the lower (higher) is "Standard deviation" ("Cross-validation score"), the better the code is evaluating the physical property of interest;

- the *main output* file named "output_ml.dat" with the determinations of the physical properties. For each input spectrum (i.e. each row in the input files, except the first one reporting wavelengths), this file contains the mean, median and standard deviation of 10,000 "bootstrap" realizations constructed using the errors uploaded by the user (for details we refer to Appendix A4 of Ucci et al. 2018). This file also contains the "id_model" necessary to link each input spectrum to a given unique model. The first 4 columns of the file (in this case reporting the density n) look like the following:

```
# id_model  mean[Log(n)]  median[Log(n)]  sigma[Log(n)]
1.00000       1.68200        1.39300        0.49995
2.00000       2.27013        1.19000        0.86704
```

3.00000	2.39010	1.65500	0.72328
4.00000	2.24428	1.19000	0.92534
2.00000	2.74181	1.19000	0.90500

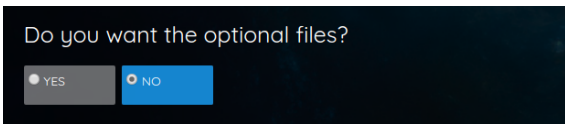
The values reported as "sigma", could be used as a first estimate of the associated uncertainty on the physical properties. A more refined analysis can be done if the user chooses to compute also the "optional files" (see Sec. 3.1). The full set of outputs and their units are: $\log_{10}(n/\mathrm{cm}^{-3})$, $\log_{10}(N_H/\mathrm{cm}^{-2})$, $\log_{10}(G/G_0)$[3], $\log_{10}(U)$[4], $\log_{10}(Z/Z_\odot)$, $\log_{10}(f_{esc})$, $\log_{10}(A_V/\mathrm{mag})$;

- a set of f files named "output_feature_importances_*.dat", where f is the number of physical properties (going from 1 to 7) that the user wants to calculate and *, depending on the user choice, can be "Av", "fesc", "G0", "n", "NH", "U", "Z". These files contain the Machine Learning "feature importances" (see Appendix B of Ucci et al. 2018) of each line for each unique model. If a line has no available measurement nor an upper limit, the feature importance is set to zero. Each row refers to each unique "id_model". For example, this is the output file for the input reported in Sec. 2.1.3:

3726	3729	3869	4102	4340	4363	4861	4959	5007	5876	6300	6548	6563	6584	6716	6731	7135
0.06	0.08	0.03	0.04	0.04	**0.00**	0.03	0.02	0.01	0.14	0.04	0.02	0.01	0.02	0.10	0.03	0.04
0.07	0.08	0.03	0.04	0.04	0.22	0.03	0.02	0.02	0.13	0.04	0.02	0.01	0.02	0.10	0.03	0.05
0.06	0.08	0.03	0.04	0.04	**0.00**	0.03	0.02	0.01	0.14	**0.00**	0.02	0.01	0.02	0.10	0.03	0.04
0.06	0.08	0.03	0.04	0.04	**0.00**	0.03	0.02	0.01	**0.00**	**0.00**	0.02	0.01	0.03	0.12	0.05	0.04

3.1 Optional output files

The user is asked to choose for the creation of "optional files":



If the user chooses "yes", these are the files produced in addition to the ones previously described:

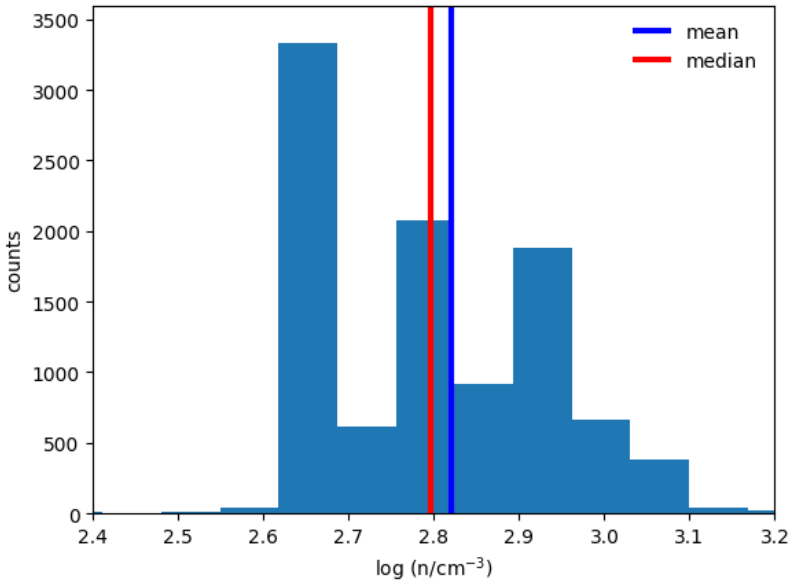
- a set of f files named "output_pdf_*.dat", where f is the number of physical properties (going from 1 to 7) that the user wants to calculate and *, depending on the user choice, can be "Av", "fesc", "G0", "n", "NH", "U", "Z". As described in Appendix A4 of Ucci et al. (2018), the code generates N = 10,000 individual new observations of each input spectrum, and outputs f optional files (one for each physical property) containing the full collection of N inferred physical properties that could subsequently be combined into a PDF. Therefore the number of rows in these files is the total number of input spectra, the number of columns is instead N (i.e. the total number of bootstrap determinations of the physical properties for each input model). The first 6 columns of the file "output_pdf_n.dat" look like the following:

[3] $G_0 = 1.6 \times 10^{-3}$ erg s^{-1} cm^{-2} (Habing 1968)

[4] The following definition for the ionization parameter is adopted: $U = Q(H)/(4\pi R_S^2 nc)$, where $Q(H)$ is the ionizing photon flux, c is the speed of light and R_S is the Strömgren radius.

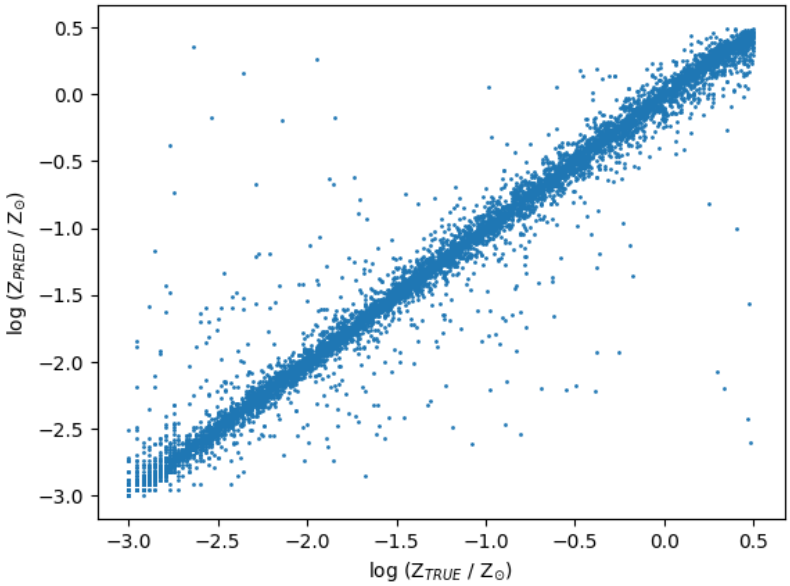
1.74700	1.67000	1.38200	1.33800	1.95700	1.29400
2.66400	2.99500	2.94000	2.80700	2.79600	2.90700
2.79400	2.90700	2.81000	2.81800	2.81000	2.69400
2.79400	3.08600	2.94100	2.94400	2.90700	2.99500
2.61100	2.67900	2.61100	2.61100	2.61900	2.59800

If the user plots the distribution of the inferred values (i.e. the distribution of values of one of the rows contained in the file), it should look like the following:



Note that the mean, median, and standard deviation described for the main output file named "output_ml.dat" are computed using precisely these distributions.

- a set of $2f$ files named "output_true_*.dat" and "output_pred_*.dat", where f is the number of physical properties (going from 1 to 7) that the user wants to calculate and *, depending on the user choice, can be "Av", "fesc", "G0", "n", "NH", "U", "Z". In the files named "output_true_*.dat" and "output_pred_*.dat", each row refers to each unique "id_model", and contains a collection of 10,000 (i.e. 10% of the training library) values of the true and inferred physical properties respectively, of specific models in the library. These files could be useful to produce for each of the physical properties a plot like the one of Figure 4 in Ucci et al. (2017):

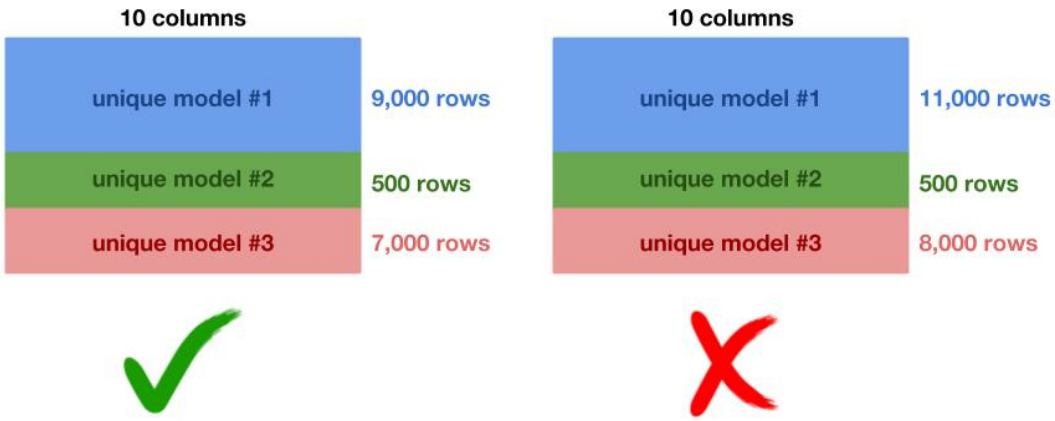


to assess the overall accuracy of the Machine Learning algorithm, or for example in the case when the user wants to compute another type of accuracy score.

4 Caveat! Size of the input files

GAME internally realizes 10,000 different realization for each input spectrum (i.e. each row of the input files described in Sec. 2.1.3). Because of memory limits, **if the user chooses "yes" for the creation of optional files, the maximum number of entries in the input file is 100,000.** This means that $i \times j < 100{,}000$, where i and j are the number of rows and columns in the input files, respectively. If, provided this limit, the user submit a larger file and chooses "yes", the run will be correctly performed, but **no optional files will be provided.**

If the user chooses "no" for the creation of optional files, the maximum number of entries in the input file FOR EACH UNIQUE MODEL is 100,000. This means that the user, before uploading the files must check that $i_{un} \times j_{un} < 100{,}000$, where i_{un} and j_{un} are the number of rows and columns in the input files for each unique model. In this example we report two different input files:



the input file on the left is accepted and does not yield memory issues because the largest value of $i_{un} \times j_{un}$ is $90,000 < 100,000$. The input file on the right is not accepted because in this case the largest value of $i_{un} \times j_{un}$ is $110,000 > 100,000$. If the user do not provide a input file with this requirement, GAME will stop and an email suggesting to divide the files into n chunks and submit n different runs will be sent.

References

Habing, H. J. 1968, Bull. Astron. Inst. Netherlands, 19, 421

Ucci, G., Ferrara, A., Gallerani, S., & Pallottini, A. 2017, MNRAS, 465, 1144

Ucci, G., Ferrara, A., Pallottini, A., & Gallerani, S. 2018, MNRAS, 477, 1484