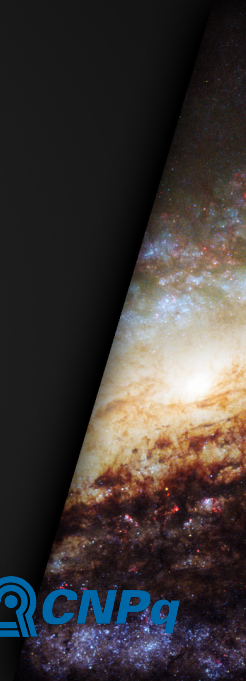
The value-added catalogues for DR2 and iDR3

Classification, photometric redshifts, and stellar masks

Lilianne Nakazono, Erik Lima and Maria Luisa Buzzo

Instituto de Astronomia, Geofísica e Ciências Atmosféricas - USP

June 1, 2021



J-PLUS

CNPq

Classification of stars, quasars, and galaxies

Classification of stars, quasars, and galaxies

- In this talk I want to bring focus on:
  - How to access the data
  - How to understand the data
  - Giving you enough information for your science (such as performance metrics and the main constraints)
- If you are interested in knowing more details about this project, I suggest that you watch this presentation from December 5th 2020: https://youtu.be/endc2yVKpnU (use the updated metrics from today's talk, though!)

Explaining the classification in one slide

- Our classification is based on supervised learning (machine learning): we use spectroscopic data from SDSS to train the model

Explaining the classification in one slide

- Our classification is based on supervised learning (machine learning): we use spectroscopic data from SDSS to train the model
- We trained two models with the Random Forest algorithm [Breiman 2001]: RF18 and RF16

[1]Note: we use a slight different notation in our paper (RF_12S+2W+4M and RF_12S+4M$\oplus$, respectively).

Explaining the classification in one slide

- Our classification is based on supervised learning (machine learning): we use spectroscopic data from SDSS to train the model
- We trained two models with the Random Forest algorithm [Breiman 2001]: RF18 and RF16 [1]

[1] Note: we use a slight different notation in our paper (RF_12S+2W+4M and RF_12S+4M$\oplus$, respectively).

Explaining the classification in one slide

- Our classification is based on supervised learning (machine learning): we use spectroscopic data from SDSS to train the model
- We trained two models with the Random Forest algorithm [Breiman 2001]: RF18 and RF16 [1]
  - RF18: 12 S-PLUS ISO magnitudes (AB) + 2 WISE magnitudes (Vega) + 4 morphological features (FWHM_n, A, B, KRON_RADIUS)

[1] Note: we use a slight different notation in our paper (RF_12S+2W+4M and RF_12S+4M$\oplus$, respectively).

Explaining the classification in one slide

- Our classification is based on supervised learning (machine learning): we use spectroscopic data from SDSS to train the model
- We trained two models with the Random Forest algorithm [Breiman 2001]: RF18 and RF16 [1]
  - RF18: 12 S-PLUS ISO magnitudes (AB) + 2 WISE magnitudes (Vega) + 4 morphological features (FWHM_n, A, B, KRON_RADIUS)
  - RF16: 12 S-PLUS ISO magnitudes (AB) + 4 morphological features (FWHM_n, A, B, KRON_RADIUS)

[1] Note: we use a slight different notation in our paper (RF_12S+2W+4M and RF_12S+4M$\oplus$, respectively).

Explaining the classification in one slide

- Our classification is based on supervised learning (machine learning): we use spectroscopic data from SDSS to train the model
- We trained two models with the Random Forest algorithm [Breiman 2001]: RF18 and RF16 [1]
  - RF18: 12 S-PLUS ISO magnitudes (AB) + 2 WISE magnitudes (Vega) + 4 morphological features (FWHM_n, A, B, KRON_RADIUS)
  - RF16: 12 S-PLUS ISO magnitudes (AB) + 4 morphological features (FWHM_n, A, B, KRON_RADIUS)
- In Nakazono et al. 2021 (submitted) we show that:
  - The 7 S-PLUS narrow bands improve the classification performance at 90% confidence

[1] Note: we use a slight different notation in our paper (RF_12S+2W+4M and RF_12S+4M$\oplus$, respectively).

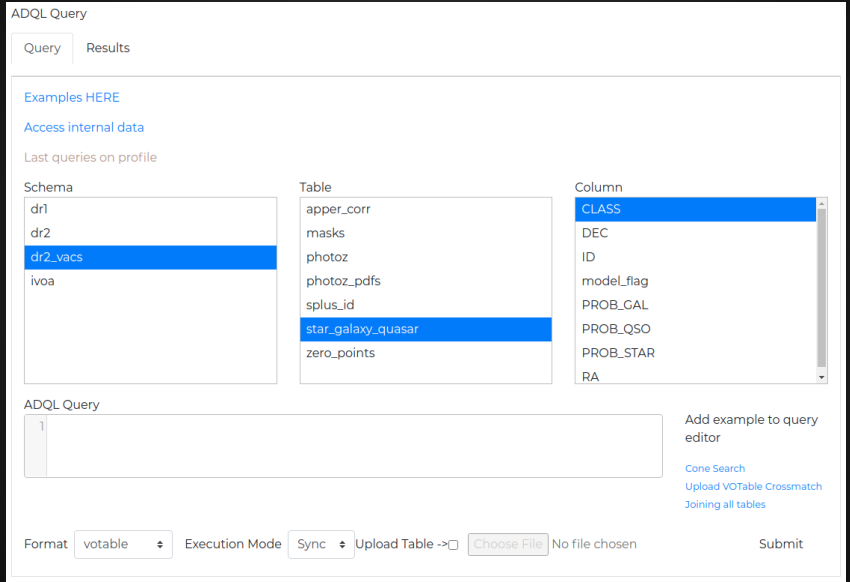
Explaining the classification in one slide

- Our classification is based on supervised learning (machine learning): we use spectroscopic data from SDSS to train the model
- We trained two models with the Random Forest algorithm [Breiman 2001]: RF18 and RF16 [1]
  - RF18: 12 S-PLUS ISO magnitudes (AB) + 2 WISE magnitudes (Vega) + 4 morphological features (FWHM_n, A, B, KRON_RADIUS)
  - RF16: 12 S-PLUS ISO magnitudes (AB) + 4 morphological features (FWHM_n, A, B, KRON_RADIUS)
- In Nakazono et al. 2021 (submitted) we show that:
  - The 7 S-PLUS narrow bands improve the classification performance at 90% confidence
  - Our classifier is robust against S-PLUS missing-band values $\rightarrow$ code runs very fast!

[1] Note: we use a slight different notation in our paper (RF_12S+2W+4M and RF_12S+4M$\oplus$, respectively).

How to access the data

Accessing the classification table via splus.cloud



SQGClass: another way to access the classification via Python

- To install:

```
pip install --upgrade splusdata==3.71
```

- In Python 3+:

```
import pandas as pd
from splusdata.vacs import SQGClass

data = pd.read_table(filename)
clf = SQGClass(model = "RF18")
results = clf.classify(data)
```

Documentation

We are still working on the SQGClass documentation and soon it will be included in splus.cloud. Until then, please contact me if you have any questions (lilianne.nakazono@usp.br).

What you need to know:

- SQGClass receives two parameters:
  - model: "RF18", "RF16" or "both". Option "both" will return classification from RF18 if object has WISE counterpart, otherwise it will return classification from RF16. Option "RF18" can return a dataframe with less rows than the input. Row index are maintained from input to output, using pd.concat([input,output], axis=1) will cross-match both tables.
  - verbose: True or False. Prints status of the process. (default: False)

Documentation

- Method classify() receives 5 parameters:
  - data: pandas dataframe that must contain the following columns with these exact names [FWHM_n, A, B, KRON_RADIUS, u_iso, J0378_iso, J0395_iso, J0410_iso, F0430_iso, g_iso, J0515_iso, r_iso, J0660_iso, i_iso, J0861_iso, z_iso]. Optional columns are: [w1mpro, w2mpro, w1snr, w2snr, w1sigmpro, w2sigmpro]
  - return_prob: True or False. Returns the probability of the source being a quasar, a star, and a galaxy (default: True)
  - match_irsa: True or False. If True, the WISE columns are retrieved from a query via IRSA TAP. It is set to False if model == "RF16" (default: False)
  - columns_wise: dictionary that sets the column names for the WISE features. (default: ["w1mpro": "w1mpro", "w2mpro": "w2mpro", "w1snr": "w1snr", "w2snr":"w2snr", "w1sigmpro":"w1sigmpro", "w2sigmpro":"w2sigmpro"]
  - verbose: True or False. Prints status of the process. (default: False)

Documentation

- Attention: the implemented WISE query is only optimized for the case for which the input data are limited for a single S-PLUS field! If your input data are spread in RA and DEC, consider splitting it in multiple dataframes by field. Then run .classify(data, match_irsa==True) in a for loop for each of the dataframes.
- BE CAREFUL: If you want to provide the WISE columns, the magnitudes must be in Vega system. Note that no Python warning is raised if the WISE magnitudes are in other system. **Not following this can lead to wrong results.**
- BE CAREFUL: SQGClass assumes that you have already corrected your data from interstellar extinction! Note that no Python warning is raised if the data were not corrected from extinction beforehand. **Not following this can lead to wrong results.**

Understanding the output

Output example - SQGClass(model="both")

HYDRA-0014

	CLASS	model_flag	PROB_QSO	PROB_STAR	PROB_GAL
0	0.0	1	0.450	0.14	0.410
1	2.0	0	0.200	0.11	0.690
2	2.0	1	0.185	0.05	0.765
3	1.0	0	0.000	1.00	0.000
4	0.0	1	0.540	0.08	0.380
...	...	...	...	...	...
78946	2.0	0	0.360	0.01	0.630
78947	0.0	1	0.520	0.26	0.220
78948	2.0	1	0.340	0.23	0.430
78949	1.0	1	0.070	0.92	0.010
78950	1.0	1	0.360	0.60	0.040

- **CLASS**: indicates the class of the source
  0 = QSO
  1 = STAR
  2 = GALAXY
- **model_flag**: indicates by which model the classification was obtained
  0 = RF18
  1 = RF16
- **PROB_QSO, PROB_STAR or PROB_GAL** $\in [0,1]$
  Note: this must return the exact same output from splus.cloud

Output example

HYDRA-0014 (model=="RF16")

	CLASS	PROB_QSO	PROB_STAR	PROB_GAL
0	0.0	0.450	0.14	0.410
1	2.0	0.180	0.24	0.580
2	2.0	0.185	0.05	0.765
3	1.0	0.000	1.00	0.000
4	0.0	0.540	0.08	0.380
...	...	...	...	...
78946	0.0	0.470	0.26	0.270
78947	0.0	0.520	0.26	0.220
78948	2.0	0.340	0.23	0.430
78949	1.0	0.070	0.92	0.010
78950	1.0	0.360	0.60	0.040

78951 rows × 4 columns

HYDRA-0014 (model=="RF18")

	CLASS	PROB_QSO	PROB_STAR	PROB_GAL
1	2	0.20	0.11	0.69
3	1	0.00	1.00	0.00
7	2	0.00	0.00	1.00
11	0	0.59	0.01	0.40
19	1	0.03	0.93	0.04
...	...	...	...	...
78939	1	0.00	1.00	0.00
78940	1	0.00	0.97	0.03
78941	1	0.00	1.00	0.00
78942	1	0.00	0.98	0.02
78946	2	0.36	0.01	0.63

24217 rows × 4 columns

Useful information for your science

Metrics

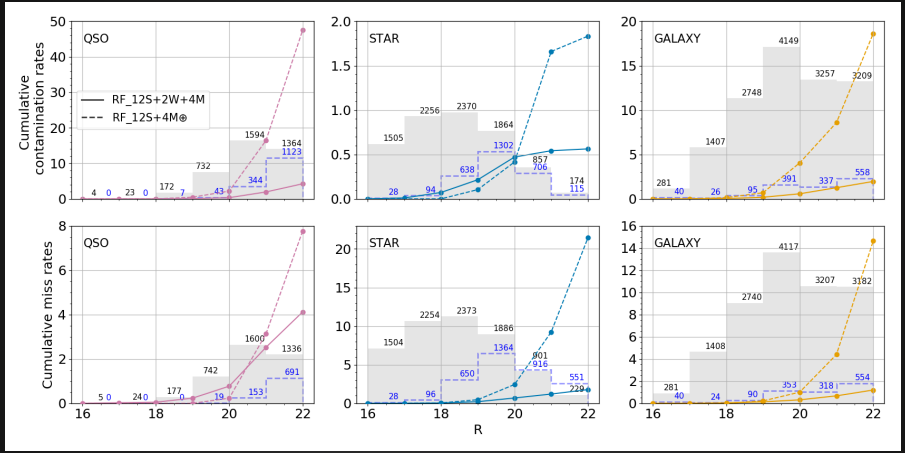


Figure 1: Cumulative contamination and miss rates per magnitude in r. Model RF18 is shown in solid lines and model RF16 is shown in dashed lines. They are calculated on the set of objects with and without WISE counterpart shown in gray and blue histograms, respectively. (Figure 11 from Nakazono et al. 2021, submitted)

Metrics

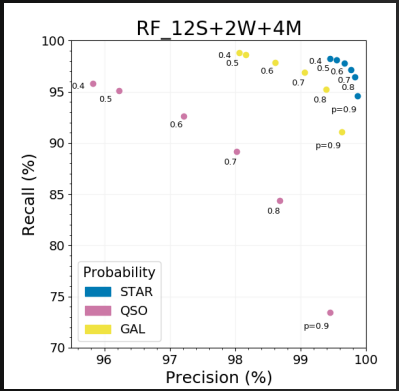


Figure 2: Precision (1 - contamination rate) and recall (1 - miss rate) relative to the chosen probability threshold for model RF18. (Figure 13 from Nakazono et al. 2021, submitted)

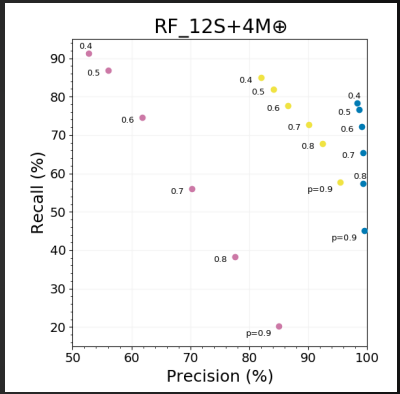


Figure 3: Precision (1 - contamination rate) and recall (1 - miss rate) relative to the chosen probability threshold for model RF16. (Figure 13 from Nakazono et al. 2021, submitted)

Final considerations

- model == "both" (or query from splus.cloud) is best recommended
- Figure 1, 2 and 3 are your best friends for deciding magnitude and/or probability thresholds!
- We expect more confusion of quasars/galaxies with stars in the Galactic Plane region ($20h < RA < 21h$) - see Figure 18 from Nakazono et al. 2021, submitted
- Consider using the same criteria that we used for training the classifiers:
  - PhotoFlag_r == 0
  - $13 < r\_iso \leqslant 22$ (we recommend stopping at 21 magnitudes for objects without WISE counterpart)

Machine learning photometric redshifts for galaxies

Introduction to the Photo-Z VACs

- The ML PDFs are generated from a combination of 20 Gaussian functions.
- dr*_vacs.photoz:
  - zml: Best single-point estimate, corresponding to the peak of the PDF;
  - zml_err: Width of the Gaussian with highest contribution to the PDF.
- dr*_vacs.photoz_pdfs:
  - PDF_Means: The mean of each component;
  - PDF_STDs: The standard deviation of each component;
  - PDF_Weights: The weight of each component.

How to download the catalogues

- Using the website:
  - Go to Tools > Catalog Tool > Query for the `photo-zs` with a query similar to this:

```
SELECT pz.ID, pz.RA, pz.DEC, pz.zml, pz.zml_err
FROM dr*_vacs.photoz as pz
(...)
```

- Using the `splusdata` package:

```
My_Query = f"""SELECT pz.ID, pz.RA, pz.DEC, pz.zml, pz.zml_err
               FROM dr*_vacs.photoz as pz
               (...)"""
Result = conn.query(My_Query)
Result.write('Query_Result.fits') # To save the resulting table
```

How to download the catalogues

- Using the website:
  - Go to Tools > Catalog Tool > Query for the PDFs with a query similar to this:

```
SELECT pz_p.ID, pz_p.PDF_Weights, pz_p.PDF_Means, pz_p.PDF_STDs
FROM dr*_vacs.photoz_pdfs as pz_p
(...)
```

- Using the `splusdata` package:

```
My_Query = f"""SELECT pz_p.ID, pz_p.PDF_Weights, pz_p.PDF_Means,
↪  pz_p.PDF_STDs
              FROM dr*_vacs.photoz_pdfs as pz_p
              (...)"""
(...)
```

How to download the catalogues

	ID	PDF_Weights	PDF_Means	PDF_STDs
0	iDR3.SPLUS-n01s38.000912	(0.00000000000011691008,0.00071095064,0.9321711…	(0.6956844,0.77322596,0.74428886,0.9110105,0.6…	(0.3336049,0.09159473,0.04273409,0.71822184,0.…
1	iDR3.SPLUS-n01s38.000913	(0.000000034468897,0.11656065,0.33243123,0.0000…	(0.3480742,0.6858107,0.6150918,0.5938524,0.643…	(0.17727922,0.2511519,0.06366815,0.14300312,0.…
2	iDR3.SPLUS-n01s38.000914	(0.000000000060101681,0.20211542,0.62547183,0.…	(0.46014214,0.83396125,0.5931754,0.7172453,0.6…	(0.21711008,0.42345712,0.07902342,0.2195485,0.…
3	iDR3.SPLUS-n01s38.000915	(0.000000000003536996,0.003834483,0.91712326,…	(0.65963846,0.94798356,0.70551306,0.9250077,0.…	(0.14546907,0.13704625,0.06037244,0.48117927,0.…
4	iDR3.SPLUS-n01s38.000916	(0.000000456352041,0.0523828454,0.0672648251,0…	(0.2441794,0.7525811,0.5235204,0.41225418,0.24…	(0.03093212,0.21951512,0.22058882,0.17748667,0.…
...	...	...	...	...
995	iDR3.SPLUS-n01s38.001909	(0.000034747951,0.016483108,0.0064003281,0.001…	(0.3377546,0.80199885,0.63551575,0.32971585,0.…	(0.10977823,0.21321587,0.1032693,0.11919042,0.…
996	iDR3.SPLUS-n01s38.001910	(0.0000000727338119,0.0000601644497,0.57573336…	(0.67881477,0.69872445,0.7397674,0.5351653,0.5…	(0.04874988,0.07330245,0.06713889,0.10451961,0.…
997	iDR3.SPLUS-n01s38.001911	(0.000000000000000014191232,0.000017362341,0.83…	(0.7669618,0.9846852,0.7491668,1.2696162,0.600…	(0.15206116,0.06889208,0.04018091,1.0556588,0.…
998	iDR3.SPLUS-n01s38.001912	(0.000000033635214,0.73089552,0.086668693,0.00…	(0.5111286,0.8350811,0.6877251,0.5025446,0.693…	(0.24841408,0.31570786,0.08579049,0.15831935,0.…
999	iDR3.SPLUS-n01s38.001913	(0.03136253,0.0000056511922,0.00000074118822,0…	(0.15942731,0.48294953,0.38357127,0.14896888,0.…	(0.01734987,0.04633716,0.20841356,0.01596459,0.…

How to obtain PDFs

- Use the `Calculate_PDFs` function in the `splusdata` package:

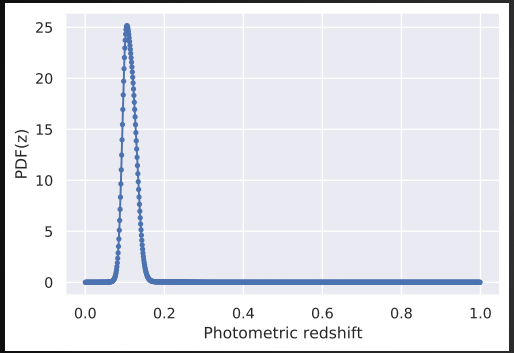
```
import pandas as pd
import splusdata

PDF_Data = pd.read_csv('PDF_File.csv')
x, PDF_List = splusdata.vacs.Calculate_PDFs(PDF_Data)
```

or

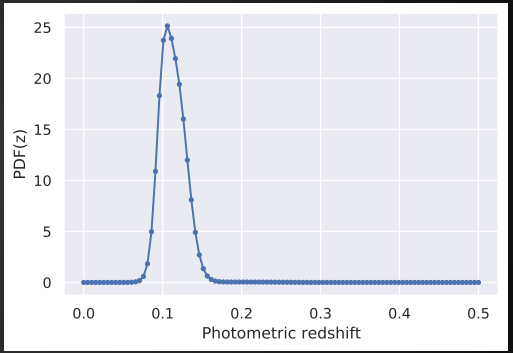
```
PDF_Data = pd.read_csv('PDF_File.csv')
new_x = np.linspace(0, 0.5, 100)
x, PDF_List = splusdata.vacs.Calculate_PDFs(PDF_Data, new_x)
```

How to obtain PDFs

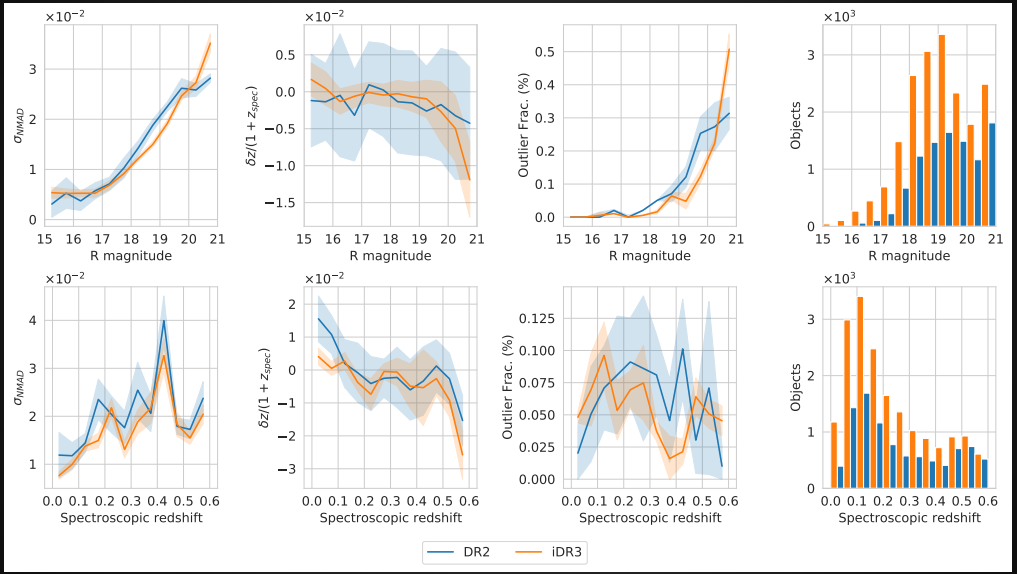
Default settings:



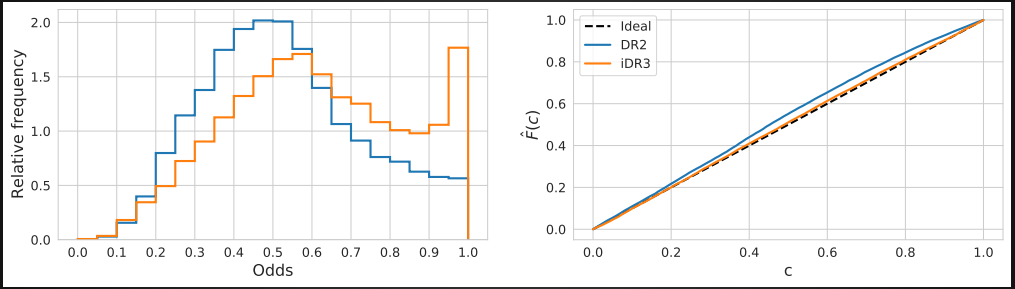
With `new_x`:



Performance of the DR2 and iDR3 models - Single point estimates



Performance of the DR2 and iDR3 models - Probability Distribution Functions



Caveats

- The result of ML techniques are most reliable inside the parameter range of the training sample,
  - ML is not good at extrapolating.
- The most reliable photo-zs are for objects with:
  - r_aper_6 between 14 and 21;
  - PhotoFlagDet 0 or 2;
  - Galaxies only (CLASS $= 2$);
  - photo-z $\lesssim 0.7$. Above this value you should check the Odds;
  - s2n_Det_aper_6 $\geqslant 3$.
- The new iDR3 Photo-Z VACs are not available yet. You will be notified when it is ready.
- Contact: erik.vini@usp.br;
- Cite as: Vinicius-Lima et al. (submitted).

Stellar masks

Stellar Masks

- The presence of bright stars can affect images/catalogs in a number of ways:
  - Saturation;
  - Increased background and noise;
  - Spikes, ghosts and artifacts;
  - Lost area, etc.

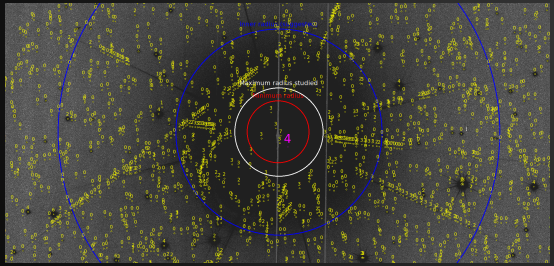


Figure 4: Inner blue circle: minimum mask suggested, Outer blue circle: suggested mask for high-accuracy studies, Red circle: PhotoFlag=0 objects start to appear.

Stellar Masks

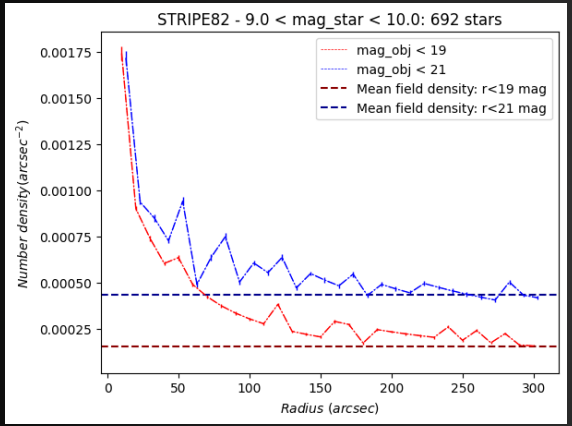
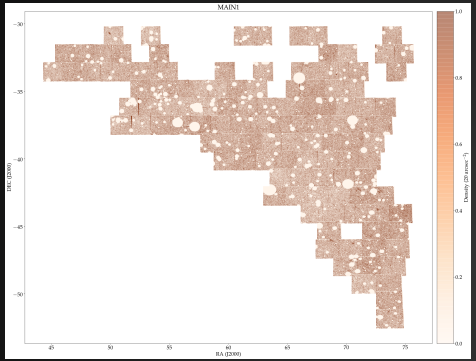


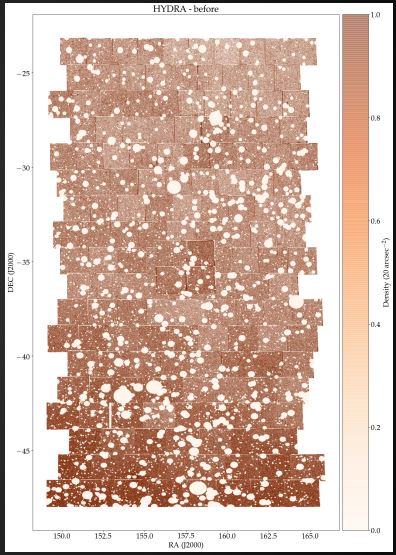
Figure 5: Number density variation in different annuli around bright stars.

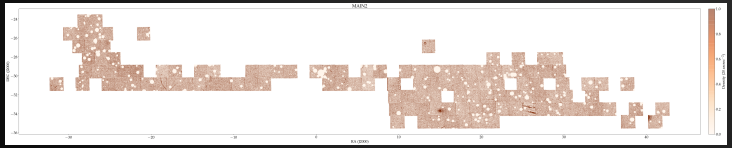
- Map of stars using Guide Star Catalogue.
- Annuli around stars to see at which radius the density reaches the average tile density.

GSC magnitude (Bmag)	**Minimum radius** (arcsec)	**Suggested radius** (arcsec)
$4.0 < mag_\star \leqslant 5.0$	700	1500
$5.0 < mag_\star \leqslant 6.0$	370	1000
$6.0 < mag_\star \leqslant 7.0$	210	800
$7.0 < mag_\star \leqslant 8.0$	160	700
$8.0 < mag_\star \leqslant 9.0$	90	450
$9.0 < mag_\star \leqslant 10.0$	50	250
$10.0 < mag_\star \leqslant 11.0$	40	200

Stellar Masks







How to download the masks

- Flag 0 = Object is not affected by any star;
- Flag 1 = Object may be affected by diffraction spikes or higher background level;
- Flag 2 = Object is inside the main body of the star.

```
SELECT mask.ID, mask.RA, mask.DEC, mask.BrightstarFlag
FROM dr*_vacs.masks as mask
(...)
```

Information

- Classification of stars, quasars, and galaxies:
  - Cite as Nakazono et al. 2021 (submitted)
  - Contact lilianne.nakazono@usp.br
- Machine learning photometric redshifts for galaxies:
  - Cite as Vinicius-Lima et al. (submitted)
  - Contact erik.vini@usp.br
- Stellar masks:
  - Cite as Buzzo et al. in preparation
  - Contact maria.buzzo@usp.br