ANNz2 - plot reference guide

I. Sadeh[1]

[1]Astrophysics Group, Department of Physics and Astronomy, University College London, Gower Street, London WC1E 6BT, United Kingdom

Abstract: The various plots produced using the example scripts of `ANNz2` are briefly explained. Version 2.0.5 of the code is used.

1 Introduction

`ANNz2` (Sadeh I., Abdalla, F.B. and Lahav O., 2015) is a new implementation of the code of Collister & Lahav (2004), which used artificial neural networks (ANNs) to estimate photometric redshifts. `ANNz2` is free and publicly available[1]. `ANNz2` uses various machine learning methods (MLMs) in addition to ANNs, such as boosted decision trees (BDTs) and k-nearest neighbors (KNNs). The MLMs utilized in `ANNz2` are implemented in the `TMVA` package[2] (Hoecker et al., 2007), which is part of the `ROOT C++` software framework[3] (Brun & Rademakers, 1997).

In the following we give a short description of the graphical output of the code, using version 2.0.5 of the code. The relevant functionality of the code is explained briefly with regards to the different plots. For a more complete explanation, please see Sadeh I., Abdalla, F.B. and Lahav O. (2015). All plots are produced as `.pdf` files, as well as `ROOT` scripts. The latter my be run as e.g.,

```
root -l script.C
```

in order to produce an interactive view of the corresponding `.pdf`.

1.1 Overview of `ANNz2`

`ANNz2` uses both regression and classification techniques for estimation of single-value photo-z solutions and probability density functions (PDFs). The different configuratios are referred to as *single regression*, *randomized regression* and *binned classification*. In addition, it is possible to run `ANNz2` in *single classification* and *randomized classification* modes. These may be used for general classification problems, unrelated to photo-z inference. A short description follows.

[1] `ANNz2` is available at https://github.com/IftachSadeh/ANNZ .
[2] See http://tmva.sourceforge.net .
[3] See http://root.cern.ch .

1.1.1 Single regression (single-value photo-z estimator)

In the simplest configuratio of `ANNz2`, a single regression is performed, using as the output the spectroscopic redshift, denoted hereafter by $z_{\rm spec}$. Consequently, the entire available training sample is used to derive per-galaxy photo-z estimations.

1.1.2 Randomized regression (single-value photo-z and PDF solutions)

Instead of choosing a single MLM, it is possible to automatically generate an ensemble of regression methods. The *randomized MLMs* differ from each other in several ways. This includes setting unique random seed initializations, as well as changing the configuratio parameters of a given algorithm. To give an example, the latter may refer to using various types and numbers of neurons in an ANN, or to arranging neurons in different layouts of hidden layers; for BDTs, the number of trees and the type of boosting algorithm may be changed, etc. Additionally, `TMVA` provides the option to perform transformations on the input-parameters, including normalization or principal component decomposition. The option is also available to only use a subset of the input parameters, or to train with pre-defined functional combinations of parameters. The transformations are done prior to training, and are transparent to the user.

Once randomized MLMs are initialized, the various methods are each trained on the entire available training sample. Subsequently, a distribution of photo-z solutions for each galaxy is generated. A selection procedure is then applied to the ensemble of answers, choosing the subset of methods which achieve optimal performance.

The selected methods are used to derive a single photo-z estimator, based on the method with the best performance. The optimized solutions are also used in concert, producing an additional (averaged) single-value solution. Finally, the ensemble of estimators are used to derive a complete probability density function. This is done by folding a weighted distribution of the solutions with the corresponding uncertainty estimators of the individual MLMs.

1.1.3 Binned classification (PDF solution)

`ANNz2` may also be run in classification-mode, employing an algorithm similar to that used by Gerdes et al. (2010); Kind & Brunner (2013); Bonnett (2013). The first step of the calculation involves dividing the redshift range of the input samples into many small bins. Within the redshift bounds of a given bin, the *signal sample* is defined as the collection of galaxies for which $z_{\rm spec}$ is within the bin. Similarly, the *background sample* includes all galaxies with $z_{\rm spec}$ outside the confines of the bin.

The algorithm proceeds by training a different classification MLM for each redshift bin. The output of a trained method in a given bin, is translated to the probability for a galaxy to have redshift which falls inside that bin. The distribution of probabilities from all of the bins is normalized to unity, accounting for possible varying bin width. It then stands as the photo-z PDF of the galaxy.

1.1.4 Additional applications

`ANNz2` may also be used as a classical classifier, addressing problems such as star/galaxy separation or morphological classification of galaxies. In this case it is possible to use a single MLM, employing the *single classification* setup. Alternatively, it is possible to choose to automatically generate multiple MLMs using *randomized classification*, finally selecting the MLM which performs best.

1.2 Definition of metrics and notation

In order to quantify the performance of the different regression configurations of `ANNz2`, several metrics are used. The metrics serve both as part of the dynamic optimization procedure of `ANNz2`, and as a means of assessing the quality of the results. All calculations take into account per-object weights. Weights may be defined by the user, or derived on the fly based on the type of analysis. For instance, the user may choose to down-weight certain galaxies based on an associated degree of confidence. Such a sub-sample would then have lower relative significance during optimization. Weights are also used in order to correct for incomplete or unrepresentative training samples.

The following metrics are used. The *photometric bias* of a single galaxy is defined as $\delta_{\rm gal} = z_{\rm phot} - z_{\rm spec}$, where $z_{\rm phot}$ and $z_{\rm spec}$ are respectively the photometric and spectroscopic redshifts of the galaxy. The *photometric scatter* represents the standard deviation of $\delta_{\rm gal}$ for a collection of galaxies. Similarly, σ_{68} denotes the half-width of the area enclosing the peak $68^{\rm th}$ percentile of the distribution of $\delta_{\rm gal}$. Another useful qualifier is the *outlier fraction* of the bias distribution, $f(\alpha\sigma)$, defined as the percentage of objects which have a bias larger than some factor, α, of either σ or σ_{68}. The *combined outlier fraction* for 2 and $3\sigma_{68}$, $f(2, 3\sigma_{68}) = \frac{1}{2}\,(\, f(2\sigma_{68}) + f(3\sigma_{68})\,)$, is also used.

The various metrics are calculated for galaxies in bins of either $z_{\rm phot}$ or $z_{\rm spec}$. They can also be defined as the average values of the metrics over all redshift bins, denoted by $<\delta>$, $<\sigma>$, $<\sigma_{68}>$ and $<f>$; these then serve as single-value qualifiers of the entire sample of galaxies.

The purpose of the bias, scatter and outlier fraction is to qualify the galaxy-by-galaxy photo-z estimation. Additionally, the overall fit of the photometric redshift distribution, $N(z_{\rm phot})$, to the true redshift distribution, $N(z_{\rm spec})$, is assessed using two metrics. The first is denoted by $N_{\rm pois}$, and stands for the sum of the bin-wise difference between the two distributions, normalized by the Poisson fluctuations. The second measure is the value of the *Kolmogorov-Smirnov* (KS) test of $N(z_{\rm phot})$ and $N(z_{\rm spec})$, which stands for the maximal distance between the cumulative distribution functions of the two distributions.

2 Randomized regression and binned classification

The first stage is to process the input dataset from `.csv` into a format recognized by `ANNz2`:

```
python scripts/annz_rndReg_advanced.py --randomRegression --genInputTrees
```

This produces some plots in

```
output/test_randReg_advanced/rootIn/plots/
```

including the distributions of the input parameters.

If the option

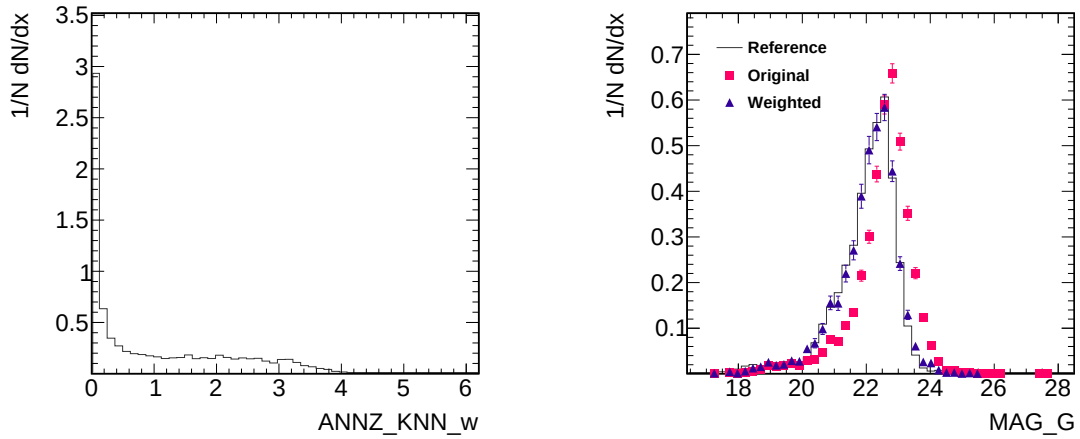
```
glob.annz["useWgtKNN"] = True
```

is used, then KNN weighting factors are also calculated. An example of the corresponding plots is shown in Fig. 1.

The next step is training,

```
python scripts/annz_rndReg_advanced.py --randomRegression --train
```

which does not produce plots. Following training, optimization takes place,



(a) (b)

Figure 1: (a) : Differential distribution of the weights which are applied to the training dataset in order for it to "become more compatible" with the reference sample. (b) : Differential distribution of one of the inputs to the weight derivation, the g-band magnitude, `MAG_G`, for three samples, as indicated: the *reference* sample, which is the photometric dataset for which the photo-zs will be calculated; the *original* training dataset, which is the training dataset before any weights are applied; the *weighted* training dataset, which is the training dataset after the derived weights are applied.

```
python scripts/annz_rndReg_advanced.py --randomRegression --optimize
```

in which the MLMs are ranked by performance and the PDFs are constructed. The relevant set of plots for the optimization are created in

```
output/test_randReg_advanced/regres/optim/plots/
```

The optimization phase in which the PDFs are derived includes producing sets of relative weights for the MLMs. An MLM with a high relative weight has a corresponding high significance in the PDF calculation. Figure 2(a) shows the relative weights for a setup of 50 MLMs. MLMs which have zero-weights are effectively ignored in the PDF calculation.

The different weighting options are compared against PDF quality criteria. The PDF with the best performance determines which set of weights is finally used. There are two quality criteria, for the two types of PDFs produced by `ANNz2`.

The first PDF is indicated by `PDF_0` in the output. The optimization of `PDF_0` is based on comparing a set of templates of PDF shapes. The PDF shapes are generated according to the intrinsic metrics of the training dataset: the distribution of the target variable, which is the redshift in case of photo-z derivation; the bias, defined as the difference between the true redshift and the derived photo-z; the scatter, defined as the variance of the distribution of bias values. The intrinsic metrics of the training dataset used for the PDF templates are shown in Fig. 2(b). The metric for optimizing the PDF is based on the cumulative distribution of the PDF. This metric for the selected PDF, along with the combination of fitted templates, is shown in Fig. 2(c).

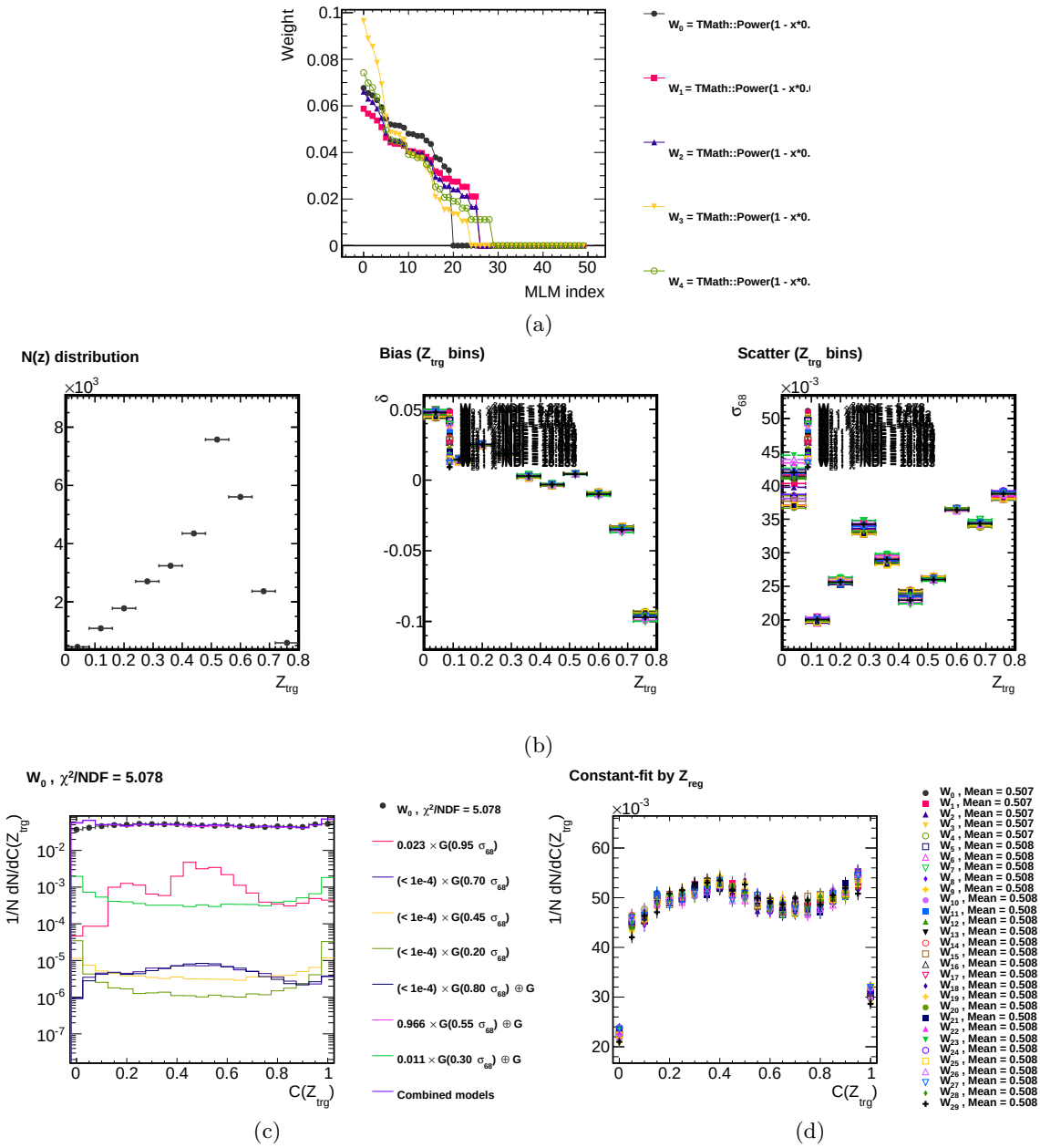


Figure 2: (a) : The relative weights for a collection of 50 MLMs, which are used as part of the PDF derivation. The various datasets correspond to different weighting schemes. In each case, the MLMs are ordered by their performance (the better-performing MLM has a higher relative weight). Candidate PDFs are computed for each one of these weighting schemes, and finally, only one is chosen. The PDF selection for `PDF_0` is done by creating templates, using the average redshift distribution, photo-z bias and photo-z scatter, which are shown in (b). The optimization metric is denoted by $C(Z_{trg})$. The selected PDF is shown in (c), as a composition of various templates. For `PDF_1`, the reference for the metric $C(Z_{trg})$ is the true redshift, Z_{trg}, as shown in (d). The various datasets in (d) represent different weighting schemes, where the one which produces the most flat distribution of $C(Z_{trg})$ is selected as the optimal `PDF_1`.

The second PDF, denoted by `PDF_1`, is optimized based on the metric of the cumulative distribution of the PDF as well. However, in this case the reference for the metric is the true redshift instead of the generated collection of templates. The cumulative distribution metric for the various weighting schemes is shown in Fig. 2(d).

After optimization is done, some performance plots are generated for the final solutions. The relevant set of plots are created in

```
output/test_randReg_advanced/regres/optim/eval/plots/
```

The correlation between the true redshift and the various photo-z solutions in shown in Figure 3. Figure 4(a) shows the bias, scatter and outlier fractions of one of the photo-z solutions as a function of either the true redshift or the photo-z. Figure 4(b) shows the corresponding bias of the ratio between the per-object bias and the associated scatter of the photo-z solution (left). The variance of this variable is also shown (right). The latter qualifies how well the photo-z uncertainty is representative of the true uncertainty, where values close to unity are the preferred result. A similar representation for the bias, scatter and outlier fraction metrics is shown in Figure 5, where here the various photo-z solutions are compared. Figure 6 shows the average metrics (over all redshift bins), comparing the results for the different photo-z solutions, in addition to the global metrics, the $N_{\rm pois}$ and KS test. Finally, Fig. 7 shows a comparison of the stacked redshift distribution of the true redshift with the photo-z solutions.

For binned classification, the optimization process is different, as it takes place during the training phase. The same performance plots as in Figs. 4 - 7 are created here as well. This is done as part of the verification phase,

```
python scripts/annz_binCls_quick.py --binnedClassification --verify
```

with the plots stored in

```
output/test_binCls_quick/binCls/verif/eval/plots/
```

`ANNz2` also has the option of computing a quality-flag for objects, which indicates whether a given object is "compatible" with the training sample; objects may be deemed incompatible if they have no corresponding training examples in the training dataset, therefore the result of the training would not be reliable.

The option of computing the quality-flag is set by

```
glob.annz["addInTrainFlag"] = True
```

For instance, using the example,

```
python scripts/annz_rndReg_weights.py --genInputTrees
python scripts/annz_rndReg_weights.py --inTrainFlag
```

plots are generated in

```
output/test_randReg_weights/inTrainFlag/plots/
```

There is an option to derive a binary estimator of the quality flag, which takes only the values 0 or 1, or instead to produce a floating point estimator between 0 and 1. The latter option is activated by setting

```
glob.annz["maxRelRatioInRef_inTrain"] = -1
```

It is recommended to first generate a floating-point estimate of the quality flag, and to study the distribution. For production, once a proper cut value for `maxRelRatioInRef_inTrain` is determined, the quality flag should be set to produce a binary estimate.

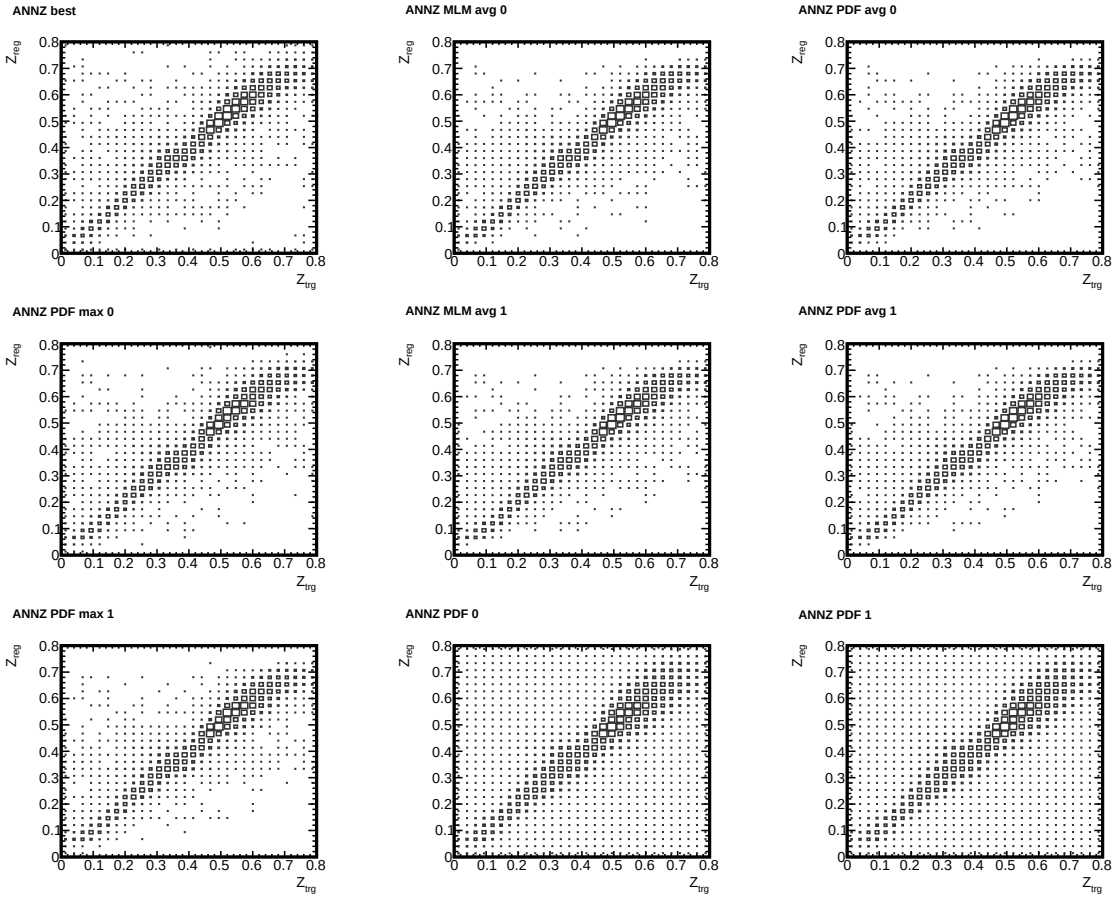
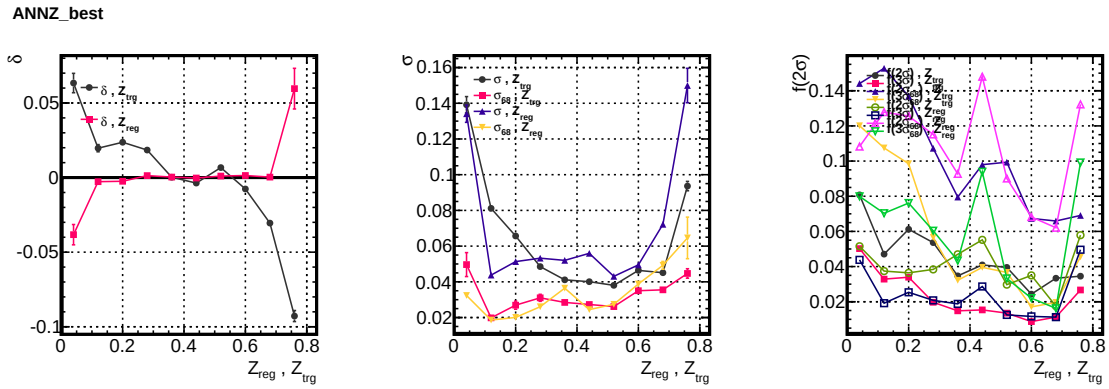
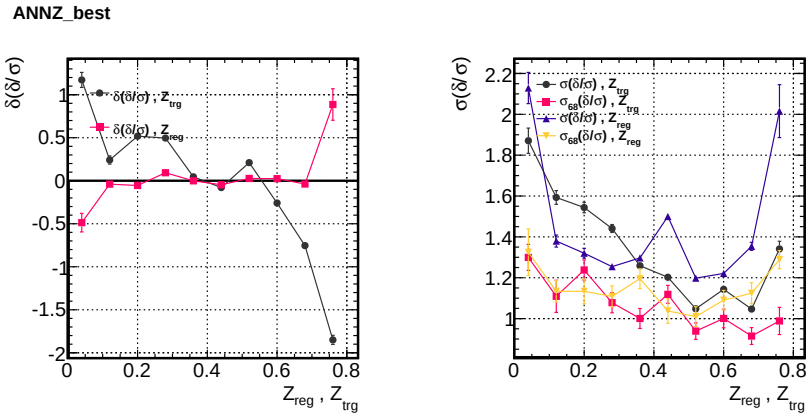


Figure 3: Correlation between the true redshift, Z_{trg}, and the photometric redshift, Z_{reg} for the different photo-z solutions (single-value solutions and PDFs).



(a)



(b)

Figure 4: (a) : The bias, δ, scatter, σ, 68$^{\rm th}$ percentile scatter, σ_{68}, and outlier fractions for 2 and $3\cdot\sigma$ or σ_{68} of one of the photo-z solutions, as a function of either the true redshift, $\rm Z_{trg}$, or the photometric redshift, $\rm Z_{reg}$. (b) : **Left:** the bias of the ratio between the per-object bias (the average value of δ from (a)) and the associated scatter of the photo-z solution, $\delta(\delta/\sigma)$; **right:** the variance of the latter variable, $\sigma(\delta/\sigma)$.

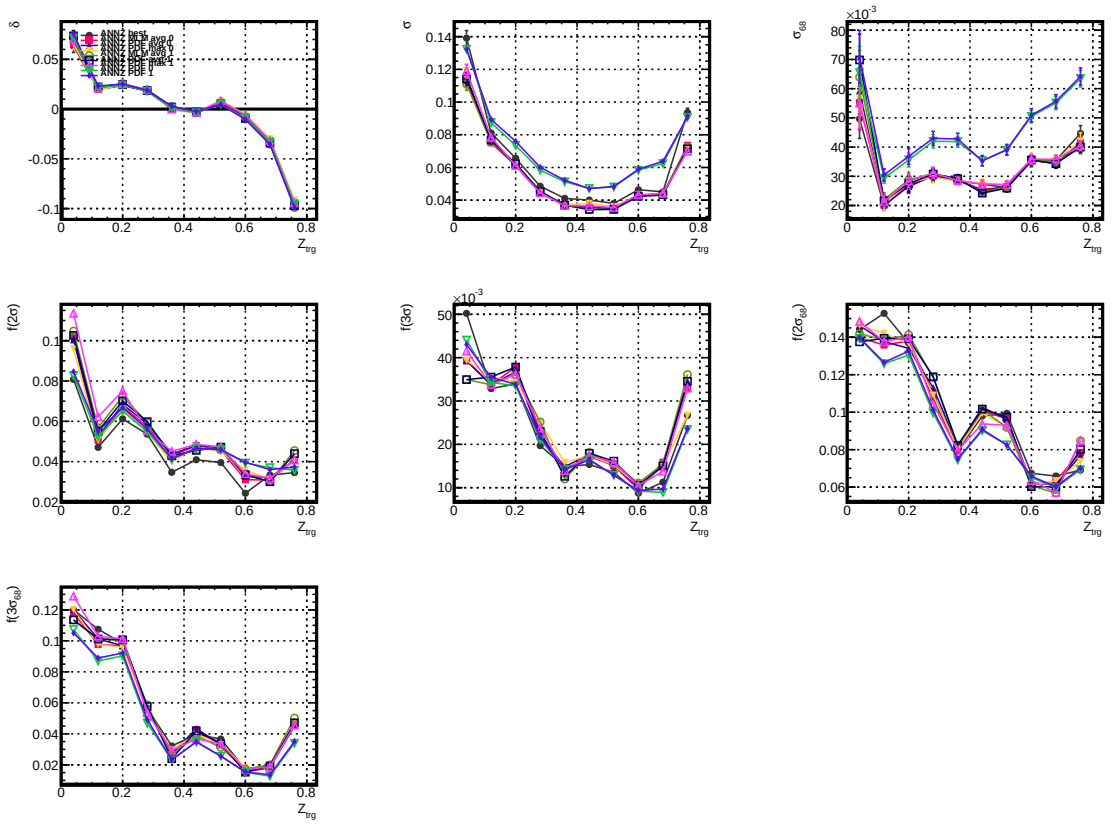


Figure 5: The bias, δ, scatter, σ, 68$^{\text{th}}$ percentile scatter, σ_{68}, and outlier fractions for 2 and $3\cdot\sigma$ or σ_{68} of all of the photo-z solutions, as a function of the true redshift, $\mathrm{Z_{trg}}$.

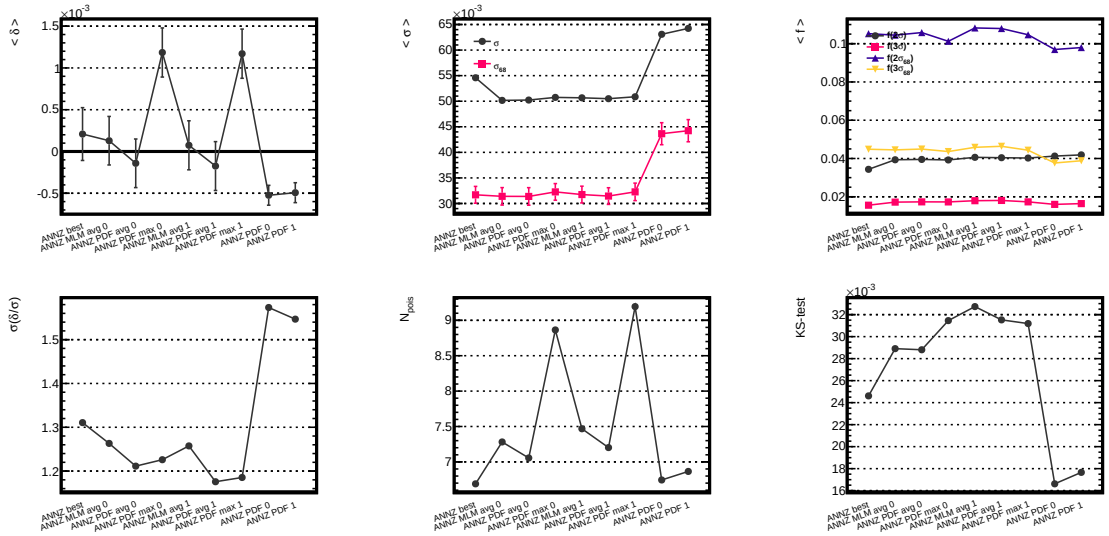


Figure 6: The average metrics for all of the photo-z solutions, including the bias, δ, scatter, σ, 68$^{\text{th}}$ percentile scatter, σ_{68}, and outlier fractions for 2 and $3 \cdot \sigma$ or σ_{68}, as well as the variance of the bias of the ratio between the per-object bias and the associated scatter of the photo-z solution, $\sigma(\delta/\sigma)$, the N_{pois} and the Kolmogorov-Smirnov (KS) metrics, as indicated.

An example of the relevant output plots is given in Fig. 8. The distribution of the quality flag is shown in Fig. 8(a), and the affect of apllying the flag on one of the input variables is shown in Fig. 8(b).

3 Randomized classification

Randomized classification may be run using

```
python examples/scripts/annz_rndCls_quick.py --randomClassification --genInputTrees
python examples/scripts/annz_rndCls_quick.py --randomClassification --train
python examples/scripts/annz_rndCls_quick.py --randomClassification --evaluate
```

The fist step generates some plots in

```
output/test_binCls_quick_paper/rootIn/plots
```

which just includes distributions of the input variables. The latter step creates performance plots in

```
output/test_binCls_quick_paper/clasif/optim/plots
```

The performance plots are shown in Figs. 9 - 10.

Figure 9 includes the distribution of $S_{\text{s/b}}$ for all generated MLMs. The latter quantity is called the *separation metric*, defined for a pair of normalized distributions, $\psi_{\text{s}}(\phi)$ and $\psi_{\text{b}}(\phi)$, as

$$S_{\text{s/b}}(\psi_{\text{s}}, \psi_{\text{b}}) = \frac{1}{2} \int_{-\infty}^{\infty} \frac{(\psi_{\text{s}}(\phi) - \psi_{\text{b}}(\phi))^2}{\psi_{\text{s}}(\phi) + \psi_{\text{b}}(\phi)} \, \text{d}\phi \, ; \tag{1}$$

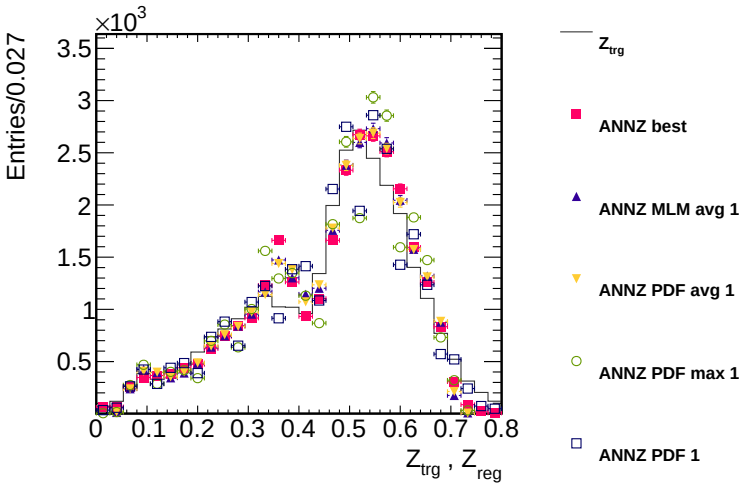


Figure 7: Redshift distribution of the true redshift, $\mathrm{Z_{trg}}$, and the various photo-z solutions, commonly denoted by $\mathrm{Z_{reg}}$, as indicated.

$S_{\mathrm{s/b}}$ qualifies the separation between the signal and background distributions. The separation may be computed using either the nominal response of the classifier (as in Figure 9(a)) or the probability estimate of the classifier (as in Figure 9(b)). We also define the *classification purity* as

$$\wp_{\mathrm{s}}(p_{\mathrm{ref}}) = \frac{\int_{p_{\mathrm{ref}}}^{1} p^{\mathrm{s}}_{\mathrm{cls}}(\phi)\,\mathrm{d}\phi}{\int_{p_{\mathrm{ref}}}^{1} \left(p^{\mathrm{s}}_{\mathrm{cls}}(\phi) + p^{\mathrm{b}}_{\mathrm{cls}}(\phi)\right)\mathrm{d}\phi}\ , \tag{2}$$

and the *classification completeness* as

$$\epsilon_{\mathrm{s}}(p_{\mathrm{ref}}) = \int_{p_{\mathrm{ref}}}^{1} p^{\mathrm{s}}_{\mathrm{cls}}(\phi)\,\mathrm{d}\phi\ . \tag{3}$$

Here $p^{\mathrm{s}}_{\mathrm{cls}}(\phi)$ and $p^{\mathrm{b}}_{\mathrm{cls}}(\phi)$ respectively stand for the value of the classification probability functions for signal and background objects at some value, $p_{\mathrm{cls}} = \phi$, and it is assumed that the probability functions are properly normalized,

$$\int_{0}^{1} p^{\mathrm{s}}_{\mathrm{cls}}(\phi)\,\mathrm{d}\phi = \int_{0}^{1} p^{\mathrm{b}}_{\mathrm{cls}}(\phi)\,\mathrm{d}\phi = 1\ .$$

Distributions of the response and of the probability estimate for several of the MLMs are also shown in Figure 9. The MLM with the highest value of $S_{\mathrm{s/b}}$ is considered as the best solution. Figure 10(a) shows the best MLMs, derived by maximizing $S_{\mathrm{s/b}}$ for either the response or the probability of the classifiers. The corresponding values of the classification purity and of the classification completeness of these MLMs are shown in Figure 10(b).

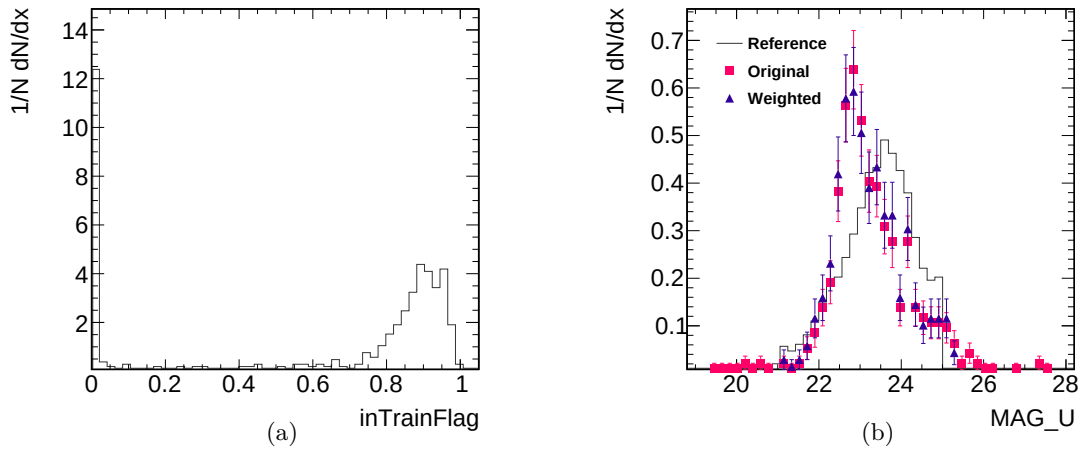
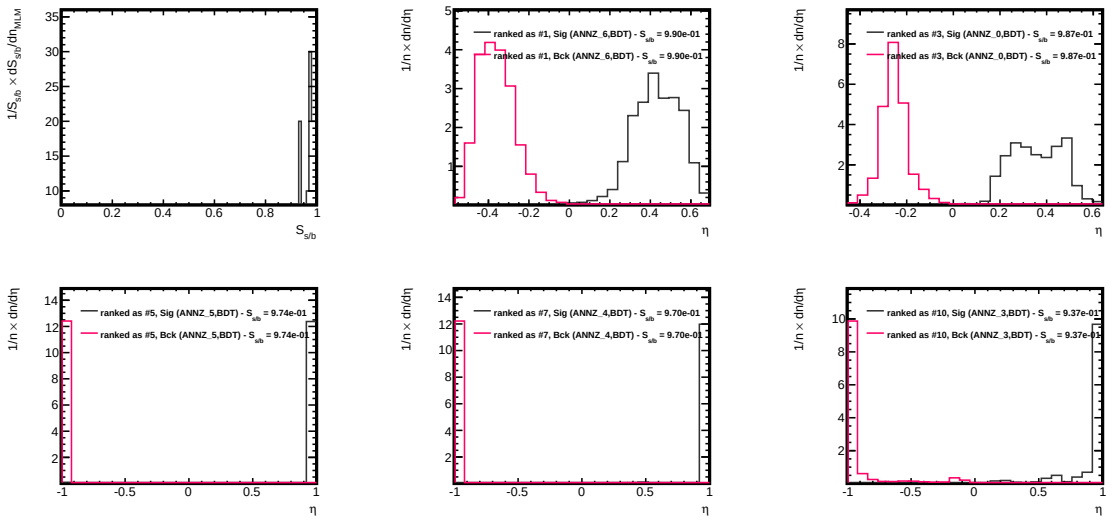
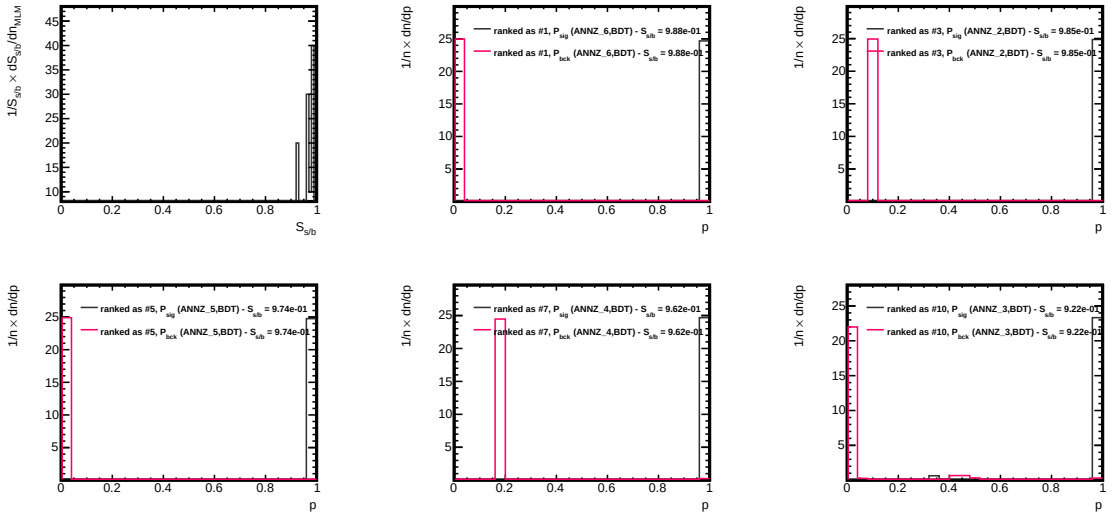


Figure 8: (a) : Differential distribution of the quality flag, in the case where it is not set as a binary estimator. (b) : Differential distribution of one of the inputs to the weight derivation, the u-band magnitude, `MAG_U`, for three samples, as indicated: the *reference* sample, which is the training dataset from which the MLMs are derived; the *original* photometric dataset for which photo-zs are being calculated (before the quality flag is applied); the *weighted* photometric dataset for which photo-zs are being calculated, where only objects which passed the quality-cut, are accepted. In this case the quality cut may be expressed as `inTrainFlag` $>$ `maxRelRatioInRef_inTrain`, with `maxRelRatioInRef_inTrain` $= 0.1$.

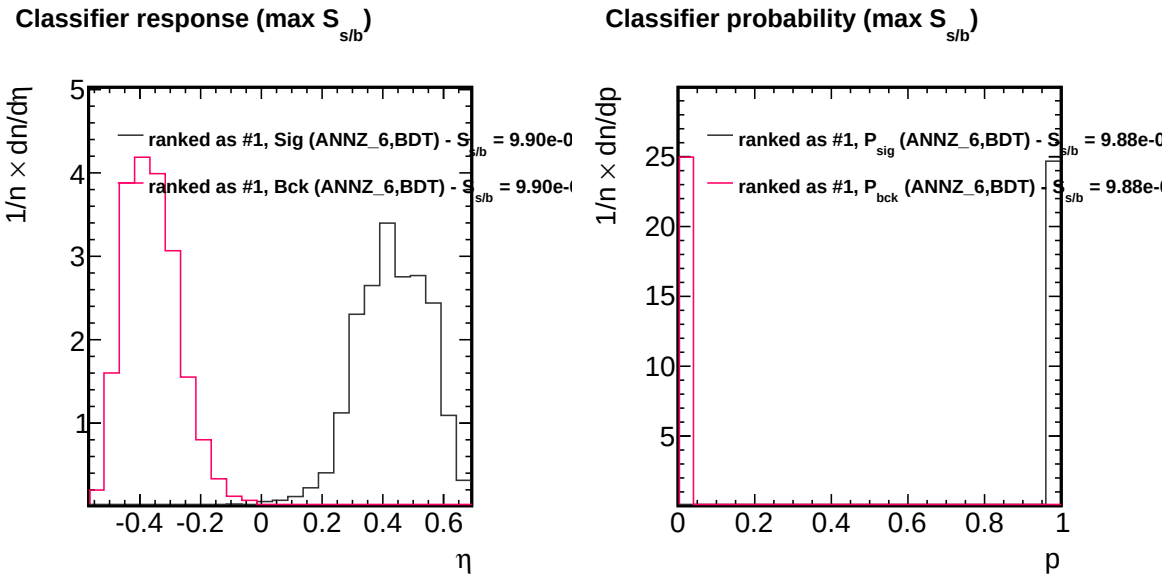


(a)

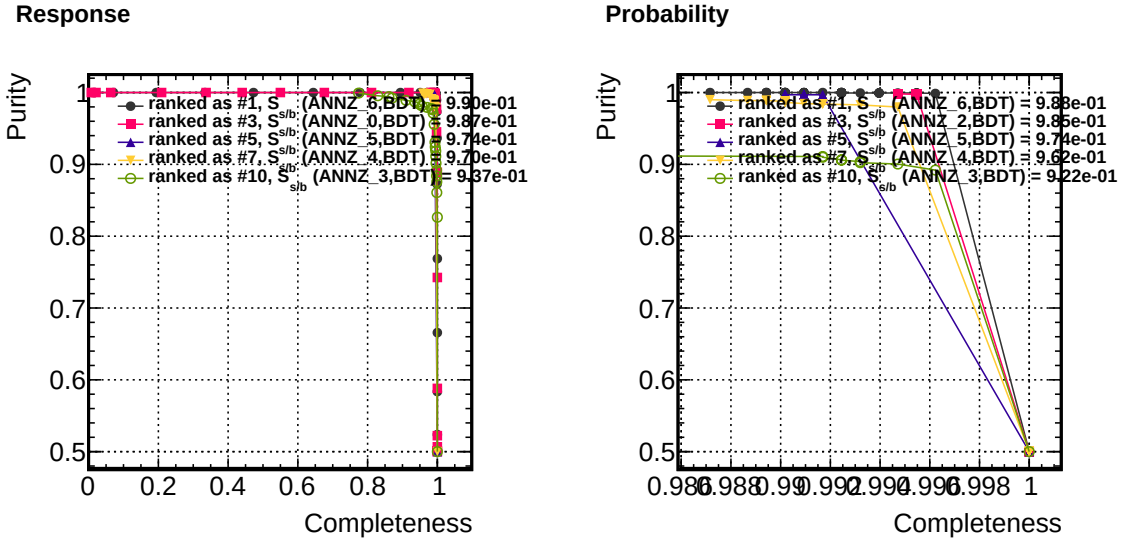


(b)

Figure 9: Differential distribution of the separation metric, $S_{\mathrm{s/b}}$, for all randomly generated MLMs, where $S_{\mathrm{s/b}}$ is calculated using distributions of either the classifier response, η, (a) or the classification probability, p, (b), as well as the corresponding distributions of η and of p for several MLMs, as indicated.



(a)



(b)

Figure 10: (a) : The classifier response, η, and the classification probability, p, for the MLMs with the best (highest) corresponding values of $S_{\mathrm{s/b}}$, as indicated. (b) : Relation between the classification purity and completeness for different choices of the classification response or the classification probability, for several of the randomized MLMs, as indicated.

References

Bonnett, C. 2013, arXiv:1312.1287

Brun, R. & Rademakers, F. 1997, Nucl.Instrum.Meth., A389, 81

Collister, A. A. & Lahav, O. 2004, Publ.Astron.Soc.Pac., 116, 345, arXiv:astro-ph/0311058

Gerdes, D. W., Sypniewski, A. J., McKay, T. A., et al. 2010, Astrophys.J., 715, 823, arXiv:0908.4085

Hoecker, A., Speckmayer, P., Stelzer, J., et al. 2007, PoS, ACAT, 040, arXiv:physics/0703039

Kind, M. C. & Brunner, R. 2013, arXiv:1303.7269

Sadeh I., Abdalla, F.B. and Lahav O. 2015, arXiv:(In preparation)