Draft version May 23, 2015
Preprint typeset using LaTeX style emulateapj v. 04/17/13

MONTE CARLO METHOD FOR CALCULATING OXYGEN ABUNDANCES AND THEIR UNCERTAINTIES FROM STRONG-LINE FLUX MEASUREMENTS

Federica B. Bianco[1], Maryam Modjaz[1], Seung Man Oh[1,2], David Fierroz[1], Yuqian Liu[1], Lisa Kewley[3,4], Or Graur[1,5]

Draft version May 23, 2015

ABSTRACT

We present the open-source Python code `pyMCZ` that determines oxygen abundance and its distribution from strong emission lines in the standard metallicity scales, based on the original IDL code of ? with updates from ?, and expanded to include more recently developed scales. The standard strong-line diagnostics have been used to estimate the oxygen abundance in the interstellar medium through various emission line ratios in many areas of astrophysics, including galaxy evolution and supernova host galaxy studies. We introduce a Python implementation of these methods that, through Monte Carlo (MC) sampling, better characterizes the statistical reddening-corrected oxygen abundance confidence region. Given line flux measurements and their uncertainties, our code produces synthetic distributions for the oxygen abundance in up to 13 metallicity scales simultaneously, as well as for *E(B-V)*, and estimates their median values and their 66% confidence regions. In addition, we provide the option of outputting the full MC distributions, and their kernel density estimates. We test our code on emission line measurements from a sample of nearby supernova host galaxies ($z < 0.15$) and compare our metallicity results with those from previous methods. We show that our metallicity estimates are consistent with previous methods but yield smaller uncertainties. We also offer visualization tools to assess the spread of the oxygen abundance in the different scales, as well as the shape of the estimated oxygen abundance distribution in each scale, and develop robust metrics for determining the appropriate MC sample size. The code is open access and open source and can be found at `https://github.com/nyusngroup/pyMCZ`

Subject headings: Galaxy: abundances — ISM: HII regions — supernovae: general

1. INTRODUCTION

Small amounts of carbon, oxygen, nitrogen, sulfur, iron and other elements provide a splash of color to the otherwise dominating grayscape of hydrogen and helium in the stars and gas of galaxies. Nevertheless, even this minute presence of heavy elements is important for many areas of astrophysics. For example, ? suggest that if it was not for the relatively high metallicity level in our Solar System, planet formation may not have been possible. With Z representing the mass fraction of metals, all elements heavier than He (also collectively called metallicity), for our own Sun the value is measured to be $Z = 0.0153$ (?), though there are suggestions of a lower Solar metallicity of $Z = 0.0134$ (??). Furthermore, when properly observed and estimated, metallicity measurements of galaxies can tightly constrain models of galaxy formation and evolution (e.g., ? and references therein), and shed light on the metallicity dependence and production conditions for different types of supernovae (SNe) and long-duration gamma-ray bursts (GRBs) (e.g., ?????????).

Metals are produced in the cores of massive stars during their fusion life cycle but also during the extreme conditions of stellar explosions. For example, the majority of iron is synthesized in thermonuclear explosions (SNe Ia) while nearly all oxygen and other α-elements (element with even atomic number lower than 22, synthesized by α-capture) are released in various kinds of core-collapse SNe (SNe II and stripped-envelope core-collapse SNe). Since new stars are born from the clouds enriched by these explosions, metallicity will increase with each generation of stars.

However, for almost all astronomical objects, metallicity cannot be measured directly. The oxygen abundance in the gas-phase is the canonical choice of metallicity indicator for interstellar medium (ISM) studies, since oxygen is the most abundant metal and only weakly depleted onto dust grains (in contrast to refractory elements such as magnesium, silicon, iron, with iron being depleted by more than a factor of 10 in Orion; see ?). The oxygen abundance[6] is expressed as $12 + \log_{10}(O/H)$, where O and H represent the number of oxygen and hydrogen atoms, respectively. In particular, ? measure a Solar oxygen abundance of $12 + \log_{10}(O/H) = 8.76 \pm 0.07$, while ? suggest $12 + \log_{10}(O/H) = 8.69$.

Importantly, oxygen exhibits very strong nebular lines in the optical wavelength range in spectra of HII regions (e.g., ???), which can be measured. Thus, many different diagnostic techniques relying on different lines of oxygen, hydrogen and other elements, have been developed (e.g.,

[1] Center for Cosmology and Particle Physics, New York University, 4 Washington Place, New York, NY 10003, USA
[2] NYU Abu Dhabi PO Box 129188 Abu Dhabi, UAE
[3] Australian National University, Research School for Astronomy & Astrophysics, Mount Stromlo Observatory, Cotter Road, Weston, ACT 2611, Australia
[4] Institute of Astronomy, University of Hawaii, 2680 Woodlawn Drive, Honolulu, HI 96822, USA
[5] Department of Astrophysics, American Museum of Natural History, Central Park West and 79th Street, New York, NY 10024-5192, USA.

[6] We note that while oxygen abundance is used to gauge metallicity, in many cases in the literature, including here, the terms metallicity and oxygen abundance are used interchangeably.

????), and are discussed in the next section.

The purpose of this paper is to present a public code that computes the metallicity from strong emission line fluxes according to the many different abundance diagnostics as well as the associated statistical uncertainties due to the measured emission line flux uncertainties. Our open-source Python package is named **pyMCZ**. This Python code allows the user to quickly produce metallicity values with sensible confidence regions for several metallicity scales at once, given an input set of spectral line measurements and their errors, and to obtain and visualize the distribution of metallicities. Below we describe the different oxygen abundance diagnostics, and our Python module. While we do not mean to advocate for a particular metallicity scale to be adopted, the comparison of multiple scale outputs, and the shape of each metallicity distribution, can guide the user in understanding the reliability of a metallicity estimate, given a set of line fluxes.

2. OXYGEN ABUNDANCE DIAGNOSTICS

In this section we present a brief overview of the various observational methods for measuring the gas-phase oxygen abundance - however, for a detailed discussion, and to understand the many caveats we encourage the reader to read the reviews by e.g. **?**, **?**, **?**, **?**, **?**, and **?**. The so-called "classical" way to estimate oxygen abundance is the electron temperature (T_e) method, which estimates the electron temperature and density of a nebula using a number of oxygen lines in different ionization states, including the auroral [OIII] λ4363 line, to then directly estimate the OII and OIII abundances and finally obtain the total oxygen abundance, after correcting for the unseen stages of ionization. However, the auroral [OIII] λ4363 line is very weak, except in low-metallicity environments, and saturates at higher metallicity, since at higher metallicities the cooling is dominated by the oxygen fine structure lines in the near-infrared (NIR). Thus other methods have been developed that use stronger lines in the spectra of HII regions. These are called *strong-line methods* and are the subject of this manuscript.

Strong-line methods can be categorized into two types: theoretical methods, which rely on calibrating various observed line ratios using photoionization methods (basically theoretically simulating HII regions, using stellar model atmospheres, stellar population synthesis and photoionization models) and empirical ones, that calibrate various observed strong line ratios using observed T_e-based metallicities.

While historically there have been large systematic offsets between the T_e method and the strong line methods, **?** demonstrated that the T_e method gives the same results as the strong line methods, if the energy distribution of the electrons in the HII regions is assumed to not be a simple Maxwell-Boltzmann distribution (as assumed in prior works), but a more realistic κ distribution (**??**), as observed in Solar plasma. They also find that the effect of changing the specific κ distribution on the strong-line methods is minor (see also **?**).

For theoretical strong-line method, a ratio of oxygen line fluxes to Hβ, referred to as R_{23}, is commonly used to determine the metallicity of galaxies (**?**):

$$R_{23} = \frac{[\mathrm{OII}]\lambda3727\ +\ [\mathrm{OIII}]\lambda4959, \lambda5007}{\mathrm{H}\beta},$$

where [OIII] λ4959,λ5007 stands for the sum of the two [OIII] lines. The drawback of this method is that is double-valued with metallicity: the same R_{23} value may correspond to two metallicity values, and thus other line ratios need to be used to break the degeneracy between the high values ("upper branch") and the low values ("lower branch") of the R_{23} metallicities (e.g., **?**, **?**). Furthermore, **?** has shown the importance of the ionization parameter, q, which can be physically understood as corresponding to the maximum velocity of an ionized front that can be driven by the local radiation field of hot massive stars that are ionizing the ISM gas. This ionization parameter needs to be taken into account in the various strong-line methods, as HII regions at the same metallicity but with different ionization parameters produce different line strengths. Calibrations of R_{23} by **?** (hereafter M91), **?** (hereafter KD02), and **?** (hereafter D13) use different theoretical photoionization models and take the ionization parameter into account, while other calibrations, such as that of **?** (hereafter Z94), do not. Thus, Z94 is valid only for metal-rich galaxies. M91 and KD02 use an iterative process to break the R_{23} degeneracy (KD02 uses the value of [NII]/[OII] and [NII]/Hα), and to constrain the ionization parameter.

As for empirical strong-line methods, the most commonly used ones are those by **?** (hereafter PP04), **?** (hereafter P05) and **?** (hereafter M13). PP04 used HII regions with measured T_e-based metallicities to derive empirical fits to strong-line ratios, and introduced the line ratios $N2$ = [NII]/Hα and the $O3N2$ index, defined as $O3N2 = \frac{[\mathrm{OIII}]}{\mathrm{H}\beta} / \frac{[\mathrm{NII}]}{\mathrm{H}\alpha}$, as metallicity diagnostics. Since $N2$ employs two closely spaced lines (Hα and NII), which are not affected by stellar absorption, nor (uncertain) reddening, and are easily observed in one simple spectroscopic setup, this has become a popular scale to use, at least for low Z SN host galaxy studies (see meta-analyses by, for example, **???**). However, it is important to remember that this scale has a number of shortcomings: it does not take into account the impact of the ionization parameter, it was initially derived based on only 137 extragalactic HII regions, and the nitrogen emission line employed saturates at high metallicity, thus the PP04_N2 method saturates at high-metallicity galaxies (at $12 + \log_{10}(O/H) > 8.8$, **?**). An updated calibration by **?** (whose scale our code also outputs, see below) based on many more T_e-based metallicities (a sample almost three times larger than that of PP04) derives a significantly shallower slope between the $O3N2$ index and the oxygen abundance than the PP04 calibration. In addition, most recently, **?** suggested that the auroral [OIII] λ4363 line, commonly used for T_e measurements, is the most problematic auroral line to use among those of [OII], [OIII] [NII], [SII], [SIII], giving rise to temperature discrepancies.

As can be seen, each scale has different advantages and disadvantages and should be used in different metallicity regimes (see detailed discussion in, for example, **???????**). Thus, `pyMCZ` outputs the oxygen abundance in the main metallicity scales (13 as of version v1.0, Spring 2015, with the KD02 diagnostic having four outputs and the PP04, P10, M13 , M08, and D13 diagnostic having multiple outputs as well).

While there is a long-standing debate about which diagnostic to use, as there are systematic metallicity offsets between different methods (recombination lines vs. strong-line method vs. the "direct" T_e method, see the above sources), **the relative metallicity trends can be considered robust, if the analysis is performed self-consistently in the same scale, and trends are seen across different scales (??)**. Thus, it is necessary to obtain error bars for the relative comparisons to be meaningful. Note that while conversion values between different scales (?) have been estimated, they are derived as ensemble values from large data sets (tens of thousands of SDSS galaxies), and they should be used with caution, if at all, on individual metallicity measurements. In addition, one should note that on account of the ongoing debate about the value of the Solar oxygen abundance (??), that the absolute oxygen calibration is still uncertain.

3. PYMCZ: A PYTHON MODULE FOR STRONG-LINE METALLICITY DIAGNOSTICS

We developed a Python package, `pyMCZ` for the simultaneous calculation of $12+\log_{10}(O/H)$ in various scales, and its confidence region. We started with the iterative IDL code by ?, hereafter referred to as IDLKD02, which has been updated in ? We translated the code into Python and added new, more recent scales (see Section 3.1) and new features, of which the most important is the capability of obtaining uncertainties on the metallicity outputs via MC sampling.

3.1. *Input and Output of code*

The input of the code is a set of spectral emission line fluxes. We assume that the observed emission lines originate in HII regions and are not due to non-thermal excitation by, for example, AGN, interstellar shocks from SNe, or stellar winds. Tests to exclude data contaminated by such non-thermal sources should be executed *prior to running this code* using the recommended line ratios by, e.g., ?, ?, and ?. Furthermore, these lines should have all the correct flux calibration (at least correct relative calibration) and *should have a signal-to-noise ratio (S/N) of at least 3.* The latter requirement is important for the success of the sampling technique. In our code, synthetic line flux measurements are drawn within a Gaussian distribution with standard deviation equal to the measurement error, and centered on the measured flux value (Section 3.2). Thus, a S/N >= 3 assures that fewer than ~ 1% of the sampled fluxes fall below zero (and are thus invalid). The code checks if for any line the S/N >= 3 condition is not satisfied, and issues a warning message.

Emission line flux values are fed into our Python implementation as in the original IDLKD02 code. The inputs are emission line flux values, and their uncertainties, for the following lines: Hα, Hβ, [OI] λ 6300, [OII] λ 3727, [OIII] λ 4959, [OIII] λ 5007, [NII] λ 6584, [SII] λ 6717, [SII] λ 6731. [SIII] λ 9096 and [SIII] λ 9532 can be used to calculate the sulfur ratio: S_{23} = ([SII] λ6717 + [SIII] λ9069)/Hβ, but are not often observed since they are in the NIR. Only one metallicity scale based on S_{23} is implemented in the current version of the code (DP00 from ?). The line fluxes are to be stored in an ASCII file, and the measurement errors in a separate ASCII file. Consult the README.md[7] in the GitHub repository for details about the input format, and find example files in the repository. If the fluxes for the specified lines are not available, the entry should be set to *NaN*. The oxygen abundance will be calculated only for metallicity scales that use valid, non-*NaN*, line fluxes. If the line fluxes necessary for specific scales are not provided, the output metallicities will default to *NaN*. In absence of measurement uncertainties, the error should be set to 0.0 in the input ASCII file. If the errors in the measurements are not provided, the code will specify that it cannot create a measurement distribution or determine a confidence interval, but it will calculate and output the nominal metallicity obtained from the input flux measurement.

The inputted line fluxes are corrected for reddening by using the observed Balmer decrement, for which Hα and Hβ flux values need to be provided. We assume case B recombination, and thus the standard value of 2.86 as the intrinsic Hα/Hβ ratio (?), and apply the standard Galactic reddening law with $R_V = 3.1$ (?). However, the user can choose other extinction laws and R_V values, if desired, given the code's open-source nature. If the input measurements are already de-reddened, the user can disable the reddening correction. If either Hα or Hβ are not provided, the reddening correction cannot be implemented. The user is notified with a warning message and has the option to proceed with the calculations with uncorrected line fluxes.

We obtain $12 + \log_{10}(O/H)$ values and their uncertainties in the metallicity scales listed below, and summarized in Table 1, and the user can choose which of the following scales to calculate, implemented as prescribed in ?, except where noted.

- **M91** (?)
- **Z94** (?) which is valid for the upper branch of R_{23} only, and we conservatively constrain it to $\log(R_{23}) < 0.9$, i.e., the range that is covered by the photoionization model grids.
- **D02** (?) for which we include, in addition to the uncertainties in the measurements, the uncertainty in the fit parameters as they appeared in the original paper.
- **PP04**: (?) 2 scales: PP04_N2, based on the [NII]/Hα ratio, and PP04_O3N2, based on $(\frac{\text{[OIII]}}{\text{H}\beta}/\frac{\text{[NII]}}{\text{H}\alpha})$.
- **P05** (?)
- **KD02 & KK04**: 4 scales: KD02_N2O2, which uses the [NII]/[OII] ratio (?), KK04_N2Hα which uses the [NII]/Hα ratio (?), KK04_R_{23} (?, appendix A2.2), which is based on the value of R_{23}, and a combined method, KD02_comb that chooses the optimal method given the input line fluxes and is implemented as described in Appendix 2.3 of ?.

[7] `https://github.com/nyusngroup/pyMCZ/blob/master/README.md`

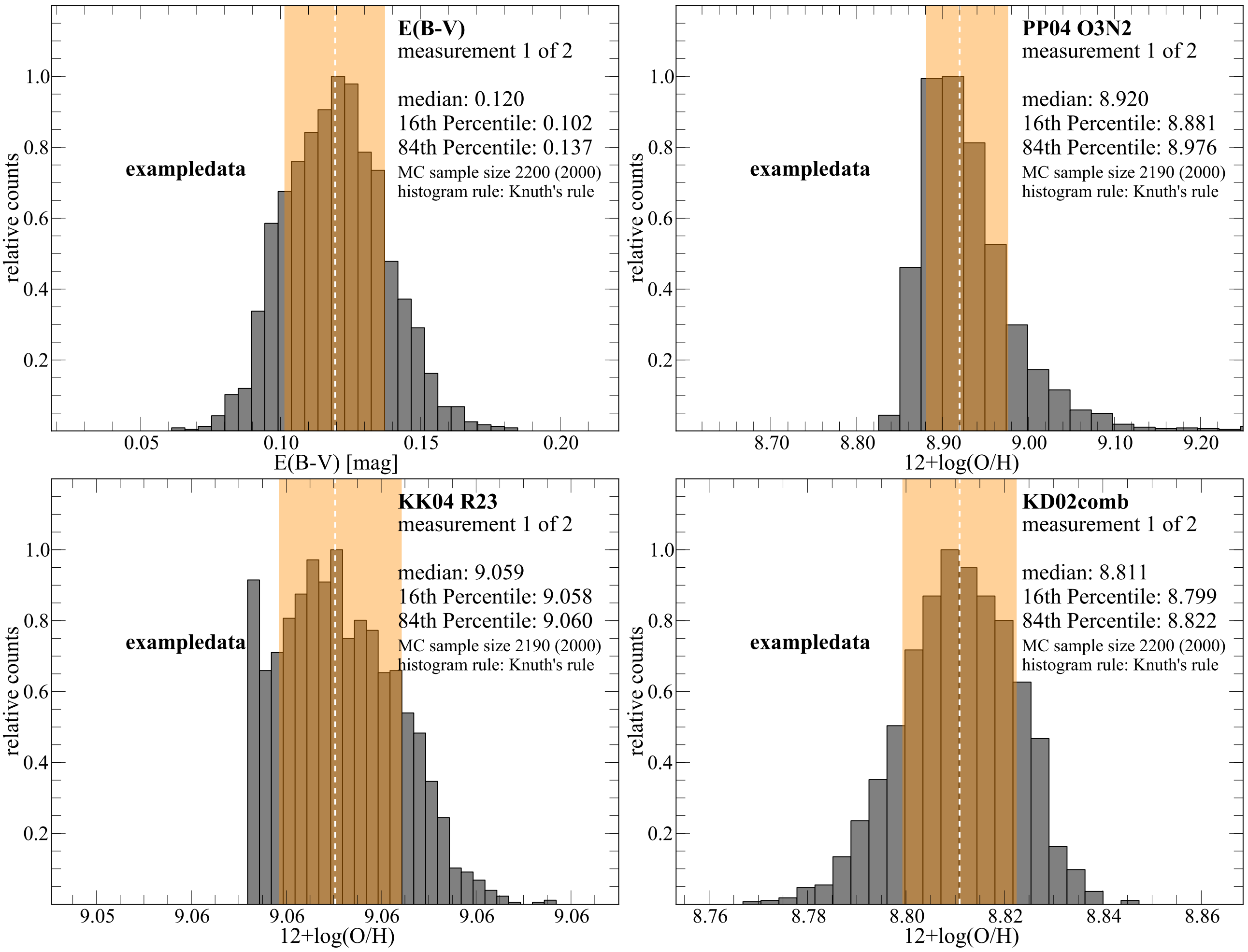

FIG. 1.— Metallicity and reddening *E(B-V)*parameter distributions based on the example data shown in Table 2: emission line data of the HII regions at the position of SN 2008D, published in **?**. The distributions are generated from N=2,000 samples. The median values are shown with dashed white lines, while the confidence region, between the 16^{th} and the 84^{th} percentile is shaded in orange. We show the metallicity scales from **?**, using [OIII] and [NII] (PP04_O3N2), the R_{23}-based scale described in **?** (KK04_R_{23}), and the combined scale of **?** (KD02_comb), updated as described in **?**. Similar plots are outputted by the code for all computed metallicity scales. Each plot indicates the scale, the sequential number of measurement in the input file (here "measurement 1 of 2", which corresponds to the first line of the input ASCII files), the median, 16^{th} and 84^{th} percentile values, the sample size (which is by default initially set to 10% larger than the requested N value, but can be smaller if some simulations lead to invalid metallicities), and, finally, the method used to choose the bin size for the histogram (Kuth's rule in this case, see Section 3.2.1).

- **M08** (**?**): This scale is a combination of the KD02 photoionization models at high metallicities and T_e based metallicities at low metallicities. Our default outputs their strong line diagnostic that is based on R_{23}, and the diagnostics based on $O3O2$ and [NII]/Hα, since the metallicity estimates from [NII]/Hα or $O3O2$ are necessary to resolve the degeneracy in the double-valued R_{23} metallicity. The other scales (based on [OII]/Hβ, [OIII]/Hβ, and [OIII]/[NII]) can be outputted upon explicit user request, via the command line options.

- **P10** (**?**): This is the so-called ONS diagnostic (involving the [OII], [OIII], [NII] and [SII] lines) and is calibrated with HII regions that have T_e based metallicities.

- **M13** (**?**): Two scales: M13_N2, which is a linear fit to the [NII] λ6584/Hα, and M13_O3N2. This is an updated calibration of the PP04 $O3N2$ method, based on a large number of T_e-based metallicity measurements, including those from the Calar Alto Legacy Integral Field spectroscopy Area survey (CALIFA, **?**, almost three time larger than the sample used in PP04). This method derives a significantly shallower slope between the $O3N2$ index and the oxygen abundance than the PP04 calibration did.

- **D13**: (**?**). The photoionization models used in KD02 and in KK04 are updated by including new atomic data within a modified photoionization code and by assuming a κ distribution for the energy of the electrons in the HII regions, rather than the simple Maxwell-Boltzmann distribution assumed in prior works. This distribution is more realistic, as observed in Solar plasma (**?**). If the user has installed the publicly available `pyqz`[8] Python module, [NII], [SII], [OIII], Hα, and Hβ lines are fed to

[8] `https://datacommons.anu.edu.au:8443/DataCommons/item/anudc:5037`

TABLE 1
SCALES AND LINE RATIOS/PARAMETERS FROM WHICH THE SCALES ARE DERIVED

Scale [a]	Lines used for metallicity calculation [b]	Support lines and scales [c]	Reference
M91	$R_{23}, \frac{\mathrm{O[III]}\lambda4959+[\mathrm{OIII}]\lambda5007}{\mathrm{O[II]}\lambda3727}$	$\frac{[\mathrm{NII}]\lambda6584}{\mathrm{H}\alpha}, \frac{[\mathrm{NII}]\lambda6584}{[\mathrm{OII}]\lambda3727}$	?
Z94	R_{23}	-	?
(DP00)	S_{23}	-	?
(P01)	$R_{23}, \frac{\mathrm{O[III]}\lambda4959+[\mathrm{OIII}]\lambda5007}{\mathrm{H}\beta}, \frac{[\mathrm{OII}]\lambda3727}{\mathrm{H}\beta}$	$\frac{[\mathrm{NII}]\lambda6584}{\mathrm{H}\alpha}, \frac{[\mathrm{NII}]\lambda6584}{[\mathrm{OII}]\lambda3727}$	?
(C01_R23)	$R_{23}, \frac{[\mathrm{OIII}]\lambda5007}{\mathrm{H}\beta}, \frac{\mathrm{O[II]}\lambda3727}{[\mathrm{OIII}]\lambda5007}$	-	?
(C01_N2S2)	$R_{23}, \frac{\mathrm{O[II]}\lambda3727}{[\mathrm{OIII}]\lambda5007}, \frac{[\mathrm{NII}]\lambda6584}{[\mathrm{SII}]\lambda6717}$	-	?
D02	$\frac{[\mathrm{NII}]\lambda6584}{\mathrm{H}\alpha}$	-	?
PP04_N2Ha	$\frac{[\mathrm{NII}]\lambda6584}{\mathrm{H}\alpha}$	-	?
PP04_O3N2	$\frac{[\mathrm{NII}]\lambda6584}{\mathrm{H}\alpha}, \frac{\mathrm{O[III]}\lambda4959+[\mathrm{OIII}]\lambda5007}{\mathrm{H}\beta}$	-	?
P05	$R_{23}, \frac{\mathrm{O[III]}\lambda4959+[\mathrm{OIII}]\lambda5007}{\mathrm{H}\beta}, \frac{[\mathrm{OII}]\lambda3727}{\mathrm{H}\beta}$	$\frac{[\mathrm{NII}]\lambda6584}{\mathrm{H}\alpha}, \frac{[\mathrm{NII}]\lambda6584}{[\mathrm{OII}]\lambda3727}$	?
KD02_N2O2	$\frac{[\mathrm{NII}]\lambda6584}{[\mathrm{OII}]\lambda3727}$	-	?
KK04_N2Ha	$\frac{[\mathrm{NII}]\lambda6584}{\mathrm{H}\alpha}, q$	$\frac{[\mathrm{NII}]\lambda6584}{[\mathrm{OII}]\lambda3727}$	?
KK04_R_{23}	$R_{23}, q,$	$\frac{[\mathrm{NII}]\lambda6584}{\mathrm{H}\alpha}, \frac{[\mathrm{NII}]\lambda6584}{[\mathrm{OII}]\lambda3727}$	?
KD02_comb [d]	M91, KD02_N2O2, KD02_N2Ha, KD04_R23	$\frac{[\mathrm{NII}]\lambda6584}{\mathrm{H}\alpha}, \frac{[\mathrm{NII}]\lambda6584}{[\mathrm{OII}]\lambda3727}$	?
M08_O3O2	$\frac{\mathrm{O[III]}\lambda4959+[\mathrm{OIII}]\lambda5007}{\mathrm{O[II]}\lambda3727}$	-	?
M08_N2Ha	$\frac{[\mathrm{NII}]\lambda6584}{\mathrm{H}\alpha}$	-	?
M08_R23	R_{23}	M08_O3O2, M08_N2Ha	?
(M08_O3Hb)	$\frac{[\mathrm{OIII}]\lambda5007}{\mathrm{H}\beta}$	M08_O3O2, M08_N2Ha	?
(M08_O2Hb)	$\frac{[\mathrm{OII}]\lambda3727}{\mathrm{H}\beta}$	M08_O3O2, M08_N2Ha	?
(M08_O3N2)	$\frac{[\mathrm{OIII}]\lambda5007}{[\mathrm{NII}]\lambda6584}$	M08_O3O2, M08_N2Ha	?
P10_ON	$\frac{[\mathrm{NII}]\lambda6584}{\mathrm{H}\alpha}, \frac{\mathrm{O[III]}\lambda4959+[\mathrm{OIII}]\lambda5007}{\mathrm{H}\beta}, \frac{[\mathrm{OII}]\lambda3727}{\mathrm{H}\beta}$	-	?
P10_ONS	$\frac{[\mathrm{NII}]\lambda6584}{\mathrm{H}\beta}, \frac{\mathrm{O[III]}\lambda4959+[\mathrm{OIII}]\lambda5007}{\mathrm{H}\beta}$ $\frac{[\mathrm{OII}]\lambda3727}{\mathrm{H}\beta}, \frac{[\mathrm{SII}]\lambda6717+[\mathrm{SII}]\lambda6731}{\mathrm{H}\beta}$	-	?
M13_N2	$\frac{[\mathrm{NII}]\lambda6584}{\mathrm{H}\beta},$	-	?
M13_O3N2	$\frac{[\mathrm{NII}]\lambda6584}{\mathrm{H}\beta}, \frac{\mathrm{O[III]}\lambda4959+[\mathrm{OIII}]\lambda5007}{\mathrm{H}\beta}$	-	?
D13_N2S2_O3S2	$\frac{[\mathrm{NII}]\lambda6584}{[\mathrm{SII}]\lambda6717+[\mathrm{SII}]\lambda6731}, \frac{[\mathrm{OIII}]\lambda5007}{[\mathrm{SII}]\lambda6717+[\mathrm{SII}]\lambda6731}$	-	?
D13_N2S2_O3Hb	$\frac{[\mathrm{NII}]\lambda6584}{[\mathrm{SII}]\lambda6717+[\mathrm{SII}]\lambda6731}, \frac{\mathrm{O[III]}\lambda4959+[\mathrm{OIII}]\lambda5007}{\mathrm{H}\beta}$	-	?
D13_N2S2_O3O2	$\frac{[\mathrm{NII}]\lambda6584}{[\mathrm{SII}]\lambda6717+[\mathrm{SII}]\lambda6731}, \frac{\mathrm{O[III]}\lambda4959+[\mathrm{OIII}]\lambda5007}{\mathrm{O[II]}\lambda3727}$	-	?
D13_N2O2_O3S2	$\frac{[\mathrm{NII}]\lambda6584}{[\mathrm{OII}]\lambda3727}, \frac{[\mathrm{OIII}]\lambda5007}{[\mathrm{SII}]\lambda6717+[\mathrm{SII}]\lambda6731}$	-	?
D13_N2O2_O3Hb	$\frac{[\mathrm{NII}]\lambda6584}{[\mathrm{OII}]\lambda3727}, \frac{\mathrm{O[III]}\lambda4959+[\mathrm{OIII}]\lambda5007}{\mathrm{H}\beta}$	-	?
D13_N2O2_O3O2	$\frac{[\mathrm{NII}]\lambda6584}{[\mathrm{OII}]\lambda3727}, \frac{\mathrm{O[III]}\lambda4959+[\mathrm{OIII}]\lambda5007}{\mathrm{O[II]}\lambda3727}$	-	?
D13_N2Ha_O3Hb	$\frac{[\mathrm{NII}]\lambda6584}{\mathrm{H}\alpha}, \frac{\mathrm{O[III]}\lambda4959+[\mathrm{OIII}]\lambda5007}{\mathrm{H}\beta}$	-	?
D13_N2Ha_O3O2	$\frac{[\mathrm{NII}]\lambda6584}{\mathrm{H}\alpha}, \frac{\mathrm{O[III]}\lambda4959+[\mathrm{OIII}]\lambda5007}{\mathrm{O[II]}\lambda3727}$	-	?

[a] Scales in parentheses are optional outputs of our pyMCZ.
[b] The flux ratios used to calculate the metallicity.
[c] Additional flux ratios used to estimate the metallicity regime, select the upper or lower branch in double valued metallicity solutions, initiate minimization, etc.
[d] This combined scale uses the output of other metallicity scales in different regimes and outputs a "recommended" value appropriate for that abundance regime.

the `pyqz` module, which produces up to eight emission line ratio diagnostics for $12+\log_{10}(O/H)$, each using two of the line ratios [NII]/[SII], [NII]/Hα, [OIII]/[SII], and [OIII]/Hβ. Our code sets the κ parameter to 20, which is the value that **?** found best resolves the inconsistencies between oxygen abundance values derived from the strong-line methods and from the "direct" T_e method[9].

- **DP00, P01 & C01 (only upon request)**: (?, ?, ?). DP00 is based on S_{23}, and is the only S_{23} scale implemented in the current version of the code. P01 is superseded by P05. C01 produces a diagnostic based on R_{23}, C01_R_{23}, and one based on [NII]/[SII], C01_N2S2; C01_R_{23} was used to calculate the combined scale described in KK04, and included in the IDLKD02 code, but it is no longer used in the new combined scale KD02_comb, which supersedes the old one. Thus, P01 and C01 are only included for historical reasons. These three scales are *not* part of the default output of our code, however they are still available upon explicit user request via command line input.

The following diagnostics were calculated in the original IDLKD02 code, and are discussed in detail in **?** and **?**: C01, P01, M91, Z94, D02, PP04, P05, KD02, KK04,

[9] The user can modify the value of κ by editing the code, if they wish.

and KD02_comb. We refer the reader to those papers for further details.

DP00 is the only diagnostic that relies on sulfur ratios S_{23}. The shortcomings of S_{23} as a tool to measure abundances are discussed in **?**. The DP00 diagnostic we implement is corrected with the addition of a term $\propto (c + S_{23}^3)^{-1}$ as suggested by **?**, which corrects the tendency of the scale to systematically underestimates the abundance, with a discrepancy growing larger at higher metallicity. However, we point out that the scatter in metallicity derived from this diagnostic compared to others remains high.

The distributions of *E(B-V)* and log(R_{23}) values are also part of the default output. Certain parameters, such as the ionization parameter q and the electron density (using the [SII] lines) are computed as long as the necessary lines are provided, they are not outputted in the current version of our code − however, since the code is open-source, the reader can easily modify the code to suite their needs.

3.2. *Computing Statistical Uncertainties*

The novel aspect of our work is that for every set of input line measurements we introduce a sampling method to obtain iterations via random sampling within the measurement uncertainties, and thus we obtain a robust result for error estimation (e.g., **???**).

Given a data set with error bars from which certain parameters are estimated, MC sampling generates synthetic data samples from a given distribution. We draw synthetic data from a Gaussian distribution centered on each measured line flux value, with standard deviation corresponding to the measurement uncertainty. The implicit assumption is made, of course, that the line flux error is Gaussian distributed in nature, and that the errors are independent.[10] For each metallicity scale, and for each of the N values chosen randomly within the relevant emission line distributions, we run the calculations that compute the metallicity. By generating synthetic data, this method effectively simulates conducting multiple experiments when repeating observations is impractical or impossible, as is the case of the emission line flux data.

The sample size N is set by the user, and one should expect an appropriate value of N to be a few 1000s, depending on the metallicity scale chosen and measurement errors (for example: $N = 2,000$ is determined to be sufficient for our example data, as shown in Section 3.3).

A distribution of parameter estimates for the oxygen abundance is generated, from which the median metallicity and its confidence region are calculated, and the results are binned and visualized in an outputted histogram (see Section 3.2.1). This is done for each scale the user chooses to calculate. The fiftieth (50%) percentile, i.e. the median, is reported as the estimated metallicity value, and the 16^{th} and 84^{th} percentiles of the metallicity estimate distribution, as its confidence region. The user can also choose to output the full metallicity distribution as binary[11] or ASCII files

This MC sampling approach takes into account the impact of the uncertain reddening (due to the uncertainties in the measurement of the Hα and Hβ fluxes), when the option for de-reddened metallicities is chosen. For each synthetic set of measurements a new reddening value is calculated based on the sampled Hα and Hβ fluxes, and used to compute the de-reddened metallicity value, so that the derived distribution of metallicity values naturally takes into account the uncertainty in reddening. The median value, and confidence intervals for *E(B-V)*, as well as a distribution histogram, are outputted together with the metallicity scales (first panel in Figure 1). If either Hα or Hβ are not provided, however, no reddening correction can be applied. The computed metallicity will not be reddening-corrected and the *E(B-V)* output will be set to zero.

Figure 1 shows metallicity estimate distributions for three representative scales (PP04_O3N2, KK04_R_{23}, and KD02_comb), and for the reddening parameter *E(B-V)*. Similar plots are produced for all scales calculated and for log(R_{23}) as a default output. Although the input line flux distributions are Gaussian, the metallicity distributions are not, for two reasons: first, since the metallicities are computed based on log values of line flux ratios, symmetric error bars in linear space will translate into asymmetric error bars in log space; and second, some metallicity scale computations are non-linear, and sometimes bimodal, especially those that use R_{23} and S_{23}, as shown in Figure 2, since the upper or lower branch metallicity value has to be chosen to break the degeneracy for each synthetic measurement.

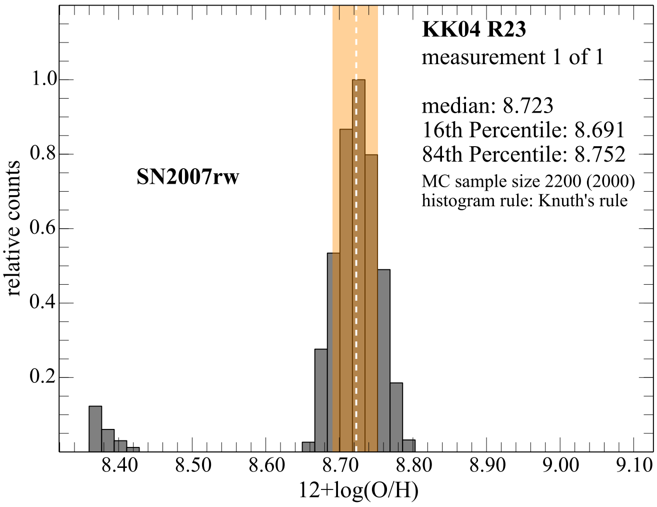

FIG. 2.— Metallicity distribution according to the KK04_R_{23} scale for the site of SN 2007rw, as measured in **?**. The R_{23} based scales are double valued: for the same R_{23} value there are two metallicity solutions. The KK04_R_{23} scale uses the ionization parameter q in an iterative fashion to determine the metallicity by selecting either the upper or lower branch value. In some cases, when the errors in the measurements are large, or for particular flux ratios, the solution may oscillate between the upper and lower branch in different realizations, giving rise to a bimodal metallicity distribution. These cases are easily identifiable by looking at the visual tools the code generates, such as this histogram.

Since the metallicity distributions are not Gaussian, the percentiles we report cannot be expressed in terms of σ values. In determining the confidence region for asymmetric and multi-modal distributions, there are broadly

[10] Users may wish to provides their own probability distribution for the emission line uncertainties, and modify the code to suite their needs.

[11] using the *pickle* Python module

three approaches (e.g., ?): choosing a symmetric interval, the shortest interval, or a *central* interval. With the central method we determine the confidence interval by choosing the left and right boundaries such that the region outside the confidence interval on each side contains 16% of the total distribution − in analogy to the one-sigma-interval of a Gaussian distribution. This ensures that the algorithm finds the proper boundaries even for asymmetric, non-Gaussian distributions, and in the case of multiple peaks (Figure 2). In summary, the output for the measured value corresponds to the fiftieth (50%) percentile (median), while the lower error bar corresponds to the 50^{th}-16^{th} percentile and the upper error bar corresponds to 84^{th}-50^{th} of the metallicity estimate distribution.

The distributions for the D02 scale include the uncertainty in the fit parameters: the oxygen abundance in this scale is generated as $12 + \log_{10}(O/H) = 9.12 \pm 0.05 + (0.73 \pm 0.10) \times$ [NII]/Hα (?). The parameters of the fit are generated as the sum of the nominal parameters (9.12 and 0.73) and a Gaussian distributed random value centered on zero, and within a standard deviation of 0.05 and 0.10, respectively, in the above units. Similarly, the distributions for the M13 scales include the uncertainty in the fit parameters as stated in ?: the oxygen abundance as a function of [NII]/Hα is parametrized as $12 + \log_{10}(O/H) = 8.743 \pm 0.027 + (0.462 \pm 0.024) \times \mathrm{NII/H}\alpha$, and as a function of $\frac{\mathrm{[OIII]/H}\beta}{\mathrm{[NII]/H}\alpha}$ as $12 + \log_{10}(O/H) = 8.533 \pm 0.012 + (0.214 \pm 0.012) \times \frac{\mathrm{[OIII]/H}\beta}{\mathrm{[NII]/H}\alpha}$.

We note that our code does not output the *systematic* uncertainty of each scale, which, for example, is ∼0.15 dex for KD02. However, if all metallicity measurements are in the *same* scale and only *relative* comparisons are made, as recommended by a number of authors, then the systematic error has no impact. Nevertheless, it is then necessary to obtain statistical error bars, as we do, for meaningful relative comparisons.

3.2.1. *Visual diagnostics to interpret the metallicity outputs*

In order for the user to check the validity of a measurement, and to better understand the distribution of metallicity values, we provide two visualizations: for each set of input line fluxes, we generate a histogram of the output distribution in all metallicity scales calculated (Figures 1, 2, and 3), and for each set of input line fluxes we generate a *box-and-whiskers* plot (hereafter *boxplot*, for short) summarizing the result of all scales calculated (Figure 4). All the plots generated are created with Python `matplotlib`[12] (?).

Choosing the binning size for a histogram is not a trivial task. ?, and ?, among others, describes various data analysis recipes for selecting a histogram bin size. Too many bins will result in an "over-fit" histogram, while too few bins may miss features of the distribution. By default, we use *Knuth's method* to choose the number of bins $N_{\rm bins}$ for each histogram. Knuth's method optimizes a Bayesian fitness function across fixed-width bins (?). Additionally, we enable a number of binning options from which the user can choose, including: the square root of the number of points, $N_{\rm bins} = \sqrt{N}$, *Rice rule* ($N_{\rm bins} = 2\sqrt[3]{N}$, e.g., ?), *Doane's formula* ($N_{\rm bins} = 1 + \log_2 N + \log_2\left(1 + \mathrm{Kurt}\sqrt{(N/6)}\right)$, where Kurt is the third moment of the distribution, ?[13]), and the full Bayesian solution, known as Bayesian Blocks, which optimizes a fitness function across an arbitrary configuration of bins, such that the bins are of variable size (?). The implementation of the latter method requires the `astroML`[14] Python package to be installed on the user's system (?). If the `astroML` package is not found, the code will default to Knuth's method. As mentioned, Knuth's method implies an optimization. In cases in which the convergence takes too long, or if the number of bins after the minimization is $N_{\rm bins}/\sqrt{N} > 5$ or $N_{\rm bins}/\sqrt{N} < 1/3$, the code will revert to Rice rule. Some methods may be computationally prohibitive with a very large sample size, or very little computational power, such as the Bayesian methods that rely on optimization (Knuth's method and the Bayesian Block method) in which case it may be convenient for a user to choose Doane's formula, Rice rule, or even the square root of the number of samples.

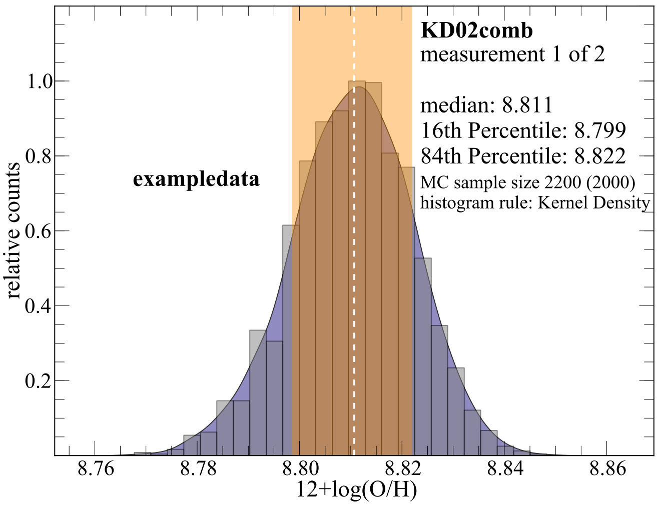

FIG. 3.— The Kernel Density for the distribution of values for the KD02comb scale for the first measurement of our example data (Table 2). The Kernel density is displayed as a blue shaded region, and is calculated via *KD Tree* with a Gaussian kernel with bandwidth given by Silverman's rule , and normalized, as described in Section3.2.1. The histogram of the distribution with bin size chosen according to Knuth's method is also plotted (gray bins) and the median and confidence intervals are shown as described in Figure 1.

Lastly, the user can generate and visualize the metallicity distribution *Kernel Density* if the `sklearn`[15] package is installed. Kernel Density Estimation (KDE) alleviates the problem of choosing the bin size, at the cost of specifying a convolution kernel. The Kernel Density of the distribution is here calculated via *KD Tree* with a gaussian kernel, as explained in the `sklearn` pack-

[12] `http://matplotlib.org/`

[13] ? attempted to address the issue of finding the proper number of bins for the histogram of a skewed distribution. Several versions of the so-called Doane's formula exist in the literature. Our formula can be found in ?

[14] `https://github.com/astroML/astroML`

[15] `http://scikit-learn.org/stable/`

TABLE 2
Example Data and their Uncertainties based on Data in ?

site[a]	[OII] λ3727	Hβ	[OIII] λ4959	[OIII] λ5007	[OI] λ6300	Hα	[NII] λ6584	[SII] λ6717	[SII] λ6731
08D	1.842 (.053)	0.958 (.032)	NaN	0.302 (.029)	0.127 (.021)	4.746 (.026)	1.642 (.022)	0.941 (.021)	0.543 (.019)
06fo	2.875 (.101)	1.251 (.044)	0.168 (.028)	0.064 (.025)	NaN	4.026 (.069)	0.781 (.033)	0.821 (.035)	0.573 (.032)

[a] These spectra were collected at the site of SN explosions: SN 2008D (line 1) and SN 2006fo (line 2)

age documentation.[16] The bandwidth of the kernel is chosen according to *Silverman's rule* (?).[17] The results will then show both a histogram, with $N_{\rm bins}$ chosen via Knuth's method, as well as the distribution Kernel Density, as shown in Figure 3. The KDE is saved as a binary python object (an `sklearn KernelDensity` object), so that it can be recovered outside the code and used as a probability distribution for further inference.

The histograms are always normalized so that the highest bin value is 1, and the Kernel Density is normalize to contain the same area as the overplotted histogram.

At each run the code also generates a boxplot (Figure 4) that summarizes the result from each scale the user chooses to calculate. For each scale the median of the $12 + \log_{10}(O/H)$ distribution is plotted as a black horizontal line. The height of the corresponding box represents the 25^{th} percentile of the $12 + \log_{10}(O/H)$ distribution, known as the *interquartile range* (IQR). The bars represent the maximum and minimum value of the distribution, excluding outliers. The outliers are plotted as circles, and are defined as all data points farther than $1.5 \times$ IQR from the 25^{th} percentile (i.e. from either end of the box).

The Solar oxygen abundance is indicated in this plot for comparison: a gray box shows a range of estimated values for Solar oxygen abundances, from $12 + \log_{10}(O/H)$=8.69 (?) to $12+\log_{10}(O/H)$=8.76 (?). The diagnostics requested by the user have a slot in the plot (in the example in Figure 3 the computed scales are the PP04, the M13, and the KD02 scales), as do diagnostics that are computed in support of other requested diagnostics (M91 in this case, which is needed to calculate KD02_comb). However a slot exists in the plot whether the diagnostic can be produced or not, i.e. if the set of input lines does not allow a requested scale to be calculated an empty column will be generated in this plot.

With this plot the user can immediately check for consistency or scatter in the metallicities derived by the requested scales (e.g., there are well-known systematic offsets between different methods, where the T_e-based, empirical methods usually give lower values than the photoionization based ones), and coarsely assess the shape of the distribution in each scale (e.g. strong asymmetry or bimodality would show up as an asymmetric box, or a very asymmetric distribution of outliers). Although, as we stated earlier, relative comparison *within* a metallicity scale are valid despite the typical spread in metallicity derived by different strong line methods, a particularly large spread may indicate difficulties in assessing metallicity for this spectrum. The user is responsible for understanding the strengths and caveats of the various diagnostics and in which ranges and conditions they may be used. The visualization tools we provide are meant to

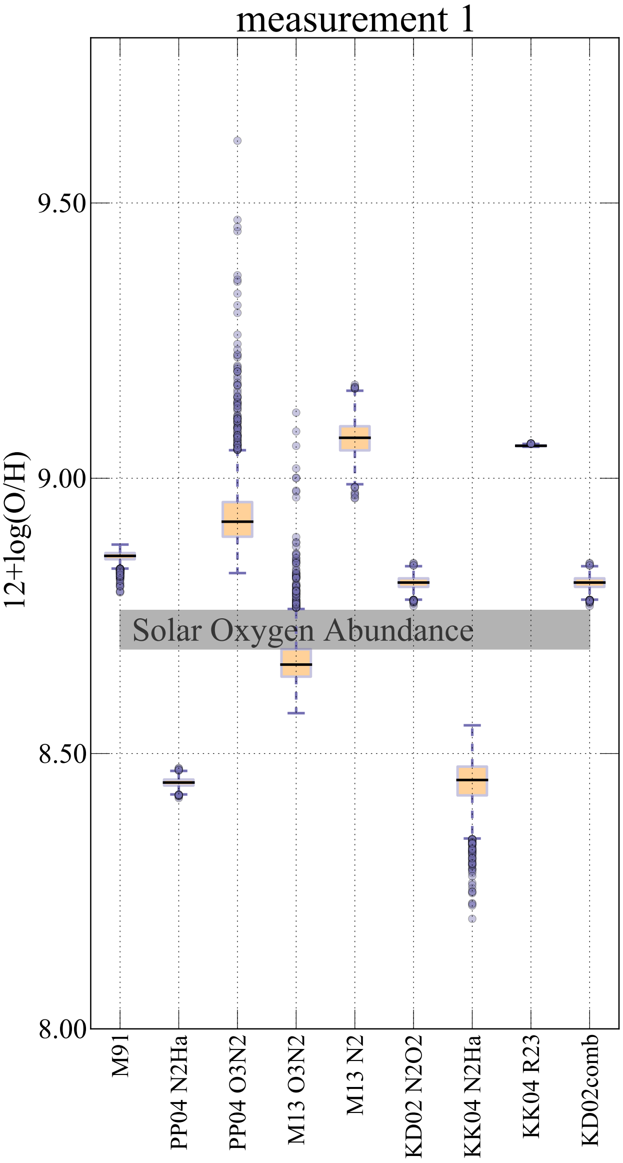

Fig. 4.— A box-and-whiskers plot showing the comparison of the results of 10 metallicity calculations, corresponding to 5 scales as listed above, calculated from the same set of measured lines (Table 2, host galaxy of SN 2008D). For each scale, the median of the resulting distribution is plotted as a horizontal line, the inter quartile range (IQR) is represented as an orange box, and the bars, joined to each end of the box by a dashed line, represent the minimum and maximum of the distribution *excluding outliers*, where outliers are defined as any point farther than 1.5× IQR from the edges for the IQR.

[16] `http://scikit-learn.org/stable/modules/density.html`

[17] By Silverman's rule the bandwidth for the KDE kernel is set to $w = 1.06\sigma\ N^{-0.2}$, where σ is the sample standard deviation. Although the bandwidth chosen accordingly to Silverman's rule is only optimal in the case of a Gaussian basis and a Gaussian distribution, and our distributions are explicitly *not* Gaussian, as explained earlier, this kernel parametrization generally provides good results for our metallicity distributions.

help the user understand the reliability of the strong-line metallicity derived for a particular HII region.

3.3. *Visual diagnostic to assure completeness of the MC simulation*

The user chooses N, the number of samples to be generated. The sample size is automatically increased by 10% at the beginning of the run, in order to assure that even if during the calculations some of the output metallicities were to result in non-valid values (*NaN*'s or infinities, for example, if division by very small numbers is required) the actual sample size is at least as large as the user intended. The code rarely produces non-valid values, and if the size of the valid output distribution were in the end smaller than the *requested* value N, i.e. if the number of non-valid outputs is larger than 10% of N, the user should in fact worry about the set of input line fluxes used.

The reliability of the metallicity estimates depends crucially on the sample size being sufficiently large to properly characterize the distribution of metallicity values. It is, however, not trivial to decide when N is sufficiently large. As soon as N is large enough, and the distribution is well characterized, adding more synthetic data will not change its shape. Consider a cumulative distribution for a metallicity scale, D02 for example, which has noise from the measurement errors, as well as from the error in the fit parameters, or KD02, which uses a non-linear combination of the input line flux values. We plot the cumulative distribution for four randomly selected subsamples of the data points in the distribution, of size 10%, 25%, 50%, 75% of N, as well as for the entire distribution. If $N/10$ constituted a sufficiently large MC sample, the cumulative distributions would typically appear smooth and, most importantly, they would overlap. In Figure 5 we show these cumulative plots for a distribution generated with $N = 200$, $N = 2,000$, and $N = 20,000$ for D02 and KD02comb.

The cumulative distributions for the subsets of the $N = 200$ samples simulation (left panel in Figure 5) are noisy and different from each other, indicating that 200 data points is not a sufficiently large dataset. Conversely, at $N = 20,000$ (right panel) the distributions are indistinguishable, indicating that $N = 2,000$ samples (the smaller subset in this panel) is large enough. The $N = 2,000$ cumulative distributions rapidly converge as the subsample size increases, indicating that a value of N between 200 and 2,000 samples will characterize the distribution appropriately for our example data set. In light of this we choose, conservatively, $N = 2,000$ to run our simulations for this dataset.

The appropriate number N of samples will depend on the scale calculated and on the input data. The errors on the measurements and the calculations performed to derive the metallicity from line ratios, which for many scales are non-linear, will determine the shape of the distribution, and thus the number of data points that are needed to fully characterize it. Consult the repository README[18] for instructions on how to generate these diagnostic plots.

3.4. *Performance and Benchmarks*

We ran a benchmark calculation on a MacBook Air with a dual-core Intel Core i7 (1.7 GHz) and 8GB of 1600 MHz DDR3 memory. The dataset we used for the calculation includes two sets of line measurements (measurement 1, the host galaxy of SN 2008D, and measurement 2, the host galaxy of SN 2006fo). The flux values and their associated errors are shown in Table 2. The code performs simple algebraic operations on large data arrays. It is vectorized along the sample-size dimension, so that the code loops over the smaller dimension corresponding to the number of sets of line measurements in input (two in our example data), while operations are generally performed on the N-dimensional vectors storing the synthetic measurements, and the N-dimensional variables derived from them (certain scales, such as KD02 and KK04, derive the ionization parameter in an iterative fashion, and require further loops).

With full graphic output (all histograms are plotted) and performing all default metallicity calculations, except those of D13 (*pyqz*), for our example data sets and a sample size $N = 2,000$ the time required by the code is ~ 9 seconds (wall clock time, and less then half a second longer in total CPU time, as the machine we tested on is dual-core), and less then 0.3 seconds of CPU time spent in the kernel within the process. Including the scales of D13 *pyqz* for the same datasets, the run time becomes ~ 19 seconds of clock time (and actual CPU time). The code time was tested on sets of 1, 2, 5, 10, 25, 50, and 100 identical measurements (copies of the emission line fluxes of the SN 2008D host galaxy in our example dataset) and the clock time is found to scale roughly quadratically with the number m of measurement in input, but with a small quadratic coefficient: $t \sim 7m + 0.07m^2$. This means effectively that the 100 measurements sample will take 200 times longer than the 1 measurement sample (of course with dependence on which line values are available in each measurement, enabling different metallicity scales to be calculated). For the CPU time spent in the kernel we find a roughly linear relation with the number of input measurements, with a very shallow dependency: $t \sim 0.05m$. This is the actual computational time in calculations performed in the code: most clock time is in fact spent in root finding, input-output, and plotting activities.

The time spent on plotting functions, which includes the calculation of the appropriate number of bins for each scale, is substantial: 1.67 seconds per distribution on average with 14 calls for this data set (one for each metallicity scale, *E(B-V)*, and R_{23}). We summarize the run time spent on each metallicity scale and memory usage for the host galaxy of SN 2008D, the first measurement of the sample dataset, in Figure 6. While the CPU usage is modest, the memory usage can be large, depending on the size of the sample. The memory usage ramps up quickly, as soon as the N line samples are created, and remains fairly constant throughout the run. Figure 6 shows the memory used by each function in the code as a function of time for a single set of input lines of our example data (the host galaxy of SN 2008D). The scales that take longer are those that require root finding (e.g. M08), or optimizations (which are done iteratively, e.g. KD02_N2O2).

[18] https://github.com/nyusngroup/pyMCZ/blob/master/README.md

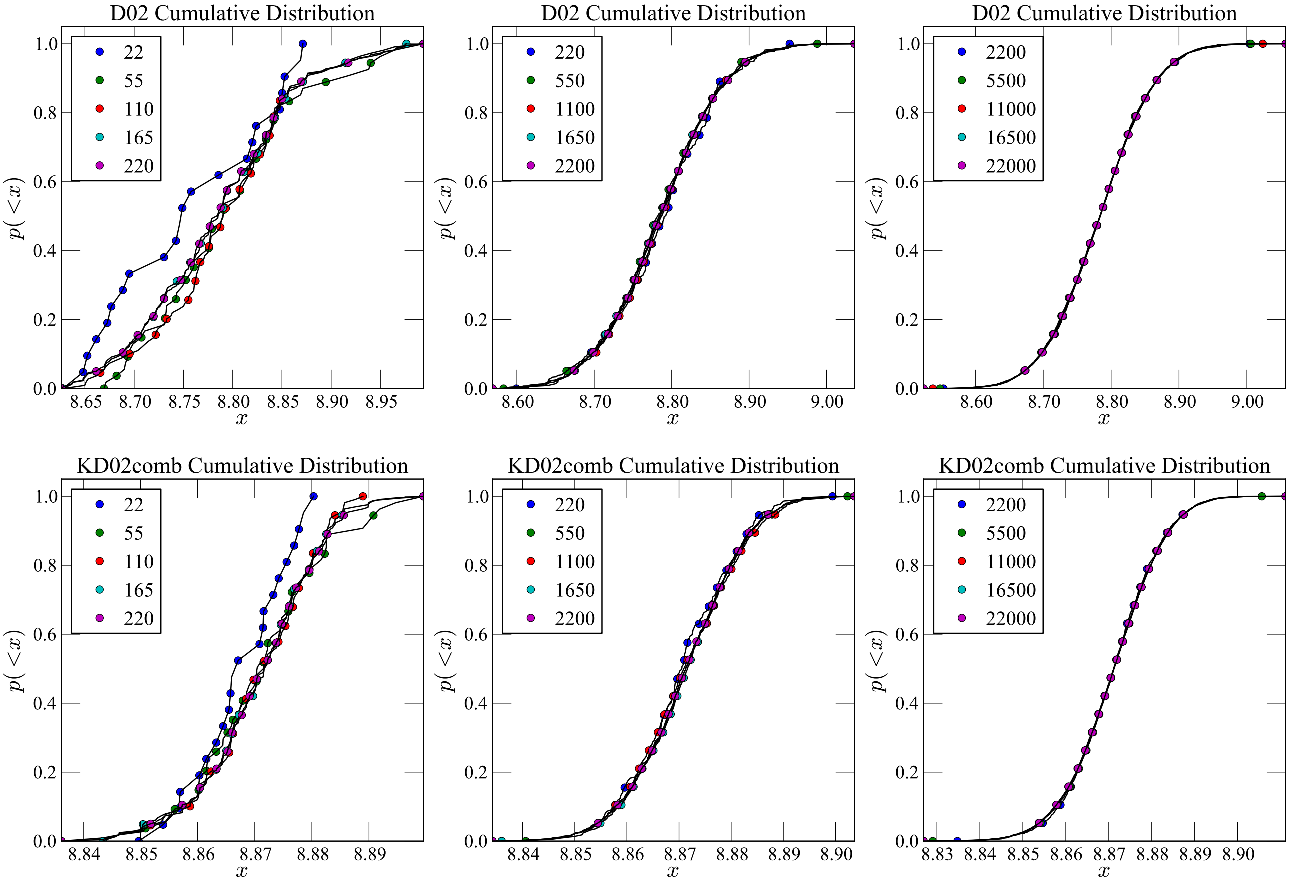

FIG. 5.— Cumulative plots of the distribution of metallicity values for the D02 (?) and KD02 scale (?, updated by ?), chosen here just as examples, where x refers to $12 + \log_{10}(O/H)$. The input used is example data 1, the emission line values for the host galaxy of SN 2008D from ?. In each plot the cumulative distribution is shown for a randomly chosen subsample of 10% of the data in the sample, as well as a subsample of 25%, 50%, and 75%, of the data, and for all data in the distribution. In the left plots the distributions are generated from $N = 200$ samples, in the center top and bottom plots from $N = 2,000$, and in the right-most column from $N = 20,000$ samples. The increasing overlap of the distributions reflects increasing completeness. In the left plots the distributions do not fully overlap, indicating that completeness is not achieved with the $N = 200$ sample. On the other end, since all subsamples are indistinguishable in the top and bottom plots on the right, which are generated with $N = 20,000$ samples, we conclude that completeness is already achieved at 10% of the $N = 20,000$ sample (the smallest sample in the plots on the right) for our example data, i.e. , at $N = 2,000$ which we use for the rest of this work.

The code is trivially parallelizable, as each measurement can be computed on a different core. Multiprocessing is enabled via the `multiprocessing` Python module, and, when enabled, it uses up to $n - 1$ cores, where n is the total number of available cores to the user (or less, if a smaller maximum number of cores is set by the user via the global variable MAXPROCESSES).

3.5. *Availability*

The source code is published under MIT-license[19] on GitHub.[20] At this time the code is released under DOI NUMBER HERE as version 1.0: NAME HERE. Project details, step-wise tutorials, and further information can be found in the module README.[21] Development is done in Linux and OS X. The package requires standard python packages, such as `numpy`, `scipy`, `pylab`, and additional features are enabled if the packages `astroml`, `skitlearn`, and `pyqz` are installed, but these packages are *not* required. Contact the authors to be included in a mailing list and notified critical changes.

4. COMPARISON TO PRIOR UNCERTAINTY COMPUTATION AND OTHER WORKS

A previous method for determining the uncertainty in the oxygen abundance (as used in ????) was an *analytic* approach of propagating the emission-line flux uncertainties: it found the minimum and abundances by minimizing (maximizing) the line ratios by adding (subtracting) to the measured line flux values their uncertainties. For comparison, we computed the metallicities and their errors in both ways (analytically and using our current MC sampling method) for three representative scales. We plot our results and the residuals in Figure 7. The errorbars in the residuals are generated by adding asymmetric errors in quadrature: $\text{residual}_{\text{min}} = \sqrt{x^2_{\text{max}} + y^2_{\text{min}}}$ and $\text{residual}_{\text{max}} = \sqrt{x^2_{\text{min}} + y^2_{\text{max}}}$.

This figure highlights a number of important points:

- The metallicity reported as the 50^{th} percentile of the metallicity parameter distribution from the MC sampling method is consistent with the analytically derived metallicity. The mean of the residuals is

[19] `https://github.com/nyusngroup/pyMCZ/blob/master/LICENSE.txt`

[20] `https://github.com/nyusngroup/pyMCZ`

[21] `https://github.com/nyusngroup/pyMCZ/blob/master/README.md`

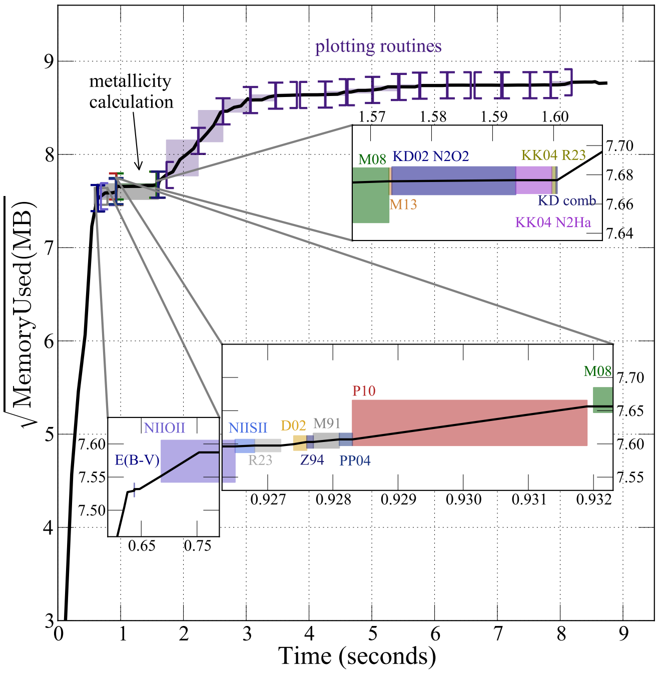

FIG. 6.— Memory usage: we plot the square root of the memory usage in Megabytes as a function of time for running our code (using N=2,000 and all default metallicity scales except the D13 *pyqz* ones) on a single set of measured emission lines (Table 2, host galaxy of SN 2008D). The square root is plotted, instead of the natural value, to enhance visibility. Three inserts show the regions where most of the metallicity scales are calculated, zoomed in, since the run time of the code is dominated by plotting routines, including the calculation of the bin size with Knuth's rule. Each function call is represented by an opening and closing bracket in the main plot, and by a shaded rectangle in the zoomed-in insets. The calculation of $N2O2$, which requires 0.25 seconds, is split between two insets, as well as the calculation of the M08 scales (three scales) which require 0.65 seconds. Altogether, the calculation of metallicity in all scales, except those in the D13 *pyqz* scales, takes ∼ 1 seconds. For technical details about the D13 *pyqz* scales performance we refer the reader to the *pyqz* package.

$\sim -2 \times 10^{-3}$, $\sim 5 \times 10^{-3}$, and $\sim 5 \times 10^{-2}$, for M91, PP04_O3N2, and KD02_comb respectively, for 20, 32, and 35 valid measurementd from the **?** data; the standard deviation of the residuals is ∼ 0.018, 0.005, 0.004 respectively - well within the respective error bars - and thus, the published results still stand (unsurprisingly, since our code, aside for the calculation of the confidence interval, uses the same algorithms developed for IDLKD02).

- The MC sampling method has smaller error bars than the analytic method. This is easily understandable, since the analytic method assumes the worst-case-scenarios, as it basically yields two metallicity parameter draws (the "minimum" and "maximum") which are in the tail of the full metallicity probability distribution. The MC sampling method is a more correct method as it empirically characterizes the full parameter estimation distribution.

The (minimal and statistically not significant) differences between our and the IDLKD02 implementation of the KD02_comb scale arise from numerical differences in the solutions of the root finding algorithms, and in the optimizations, which we carry out until convergence, while IDLKD02 estimated three iterations would be sufficient for minimization. The differences in the M91 and PP04_O3O2 scales are only due to the shape of the distributions.

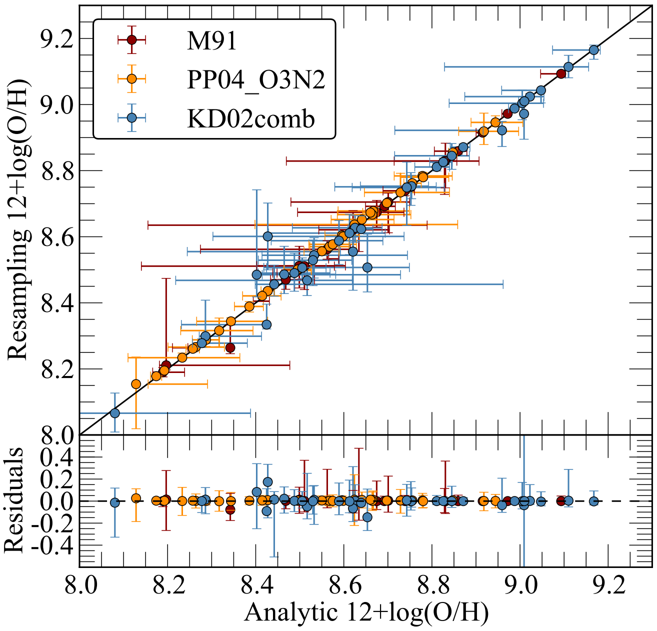

FIG. 7.— Comparison of metallicity estimation between the analytic method and our MC sampling method (top) and their residuals (bottom) for three different metallicity scales. Flux measurements come from 19 galaxies previously measured in **?**. Note that for KD02_comb and M91, an din most cases for PP04_O3N2 as well, the MC sampling error bars are smaller than those of the analytic propagation, which assumes worst-case scenarios. The metallicity value derived by our code, which is the 50th percentile of the metallicity distribution from MC sampling, is consistent with the analytically derived metallicity in all scales, for all measurements.

4.1. *Comparison with other works*

The field of SN host metallicity studies is rapidly developing, and these studies are crucial avenues for constraining the progenitor systems of different kinds of explosions - however, a few of the works in this field do not compute errors and others do not show how they compute their statistical errors (e.g., **????**).

In contrast, the general metallicity field has considered in detail how to estimate the uncertainties in measured metallicities. However, those codes are not open-source and many of them are for specific scales. For example: **?** also use MC sampling to estimate the metallicity uncertainties (in their case using N=500 trials and assuming a Gaussian distribution for metallicity parameter distribution) but only do this for two scales, KK04 and P05.

Some strong line methods fit a set of stellar population synthesis and photoionization models directly to the spectra of HII regions by changing the model parameters. For computing the metallicities of the SDSS star forming galaxies, **?** fit a combination of stellar population synthesis models and photoionization models to the observed strong emission lines [OII], $H\beta$, [OIII], $H\alpha$, NII and SII and report the median of the metallicity likelihood distribution as the metallicity estimate, with the width of the distribution giving the 1-σ (Gaussian) error. **?** use Bayesian inference to derive the joint and marginalized posterior probability density functions for metallicity Z and ionization parameter q given a set of observed line fluxes and an input photoionization model. They provide a publicly available IDL implementation of their method named *IZI* (inferring metallicities – Z – and ionization parameters).

5. CONCLUSIONS

We have presented the open-source Python code `pyMCZ` that determines oxygen abundance and its error distribution from strong emission lines in a total of up to 13 scales (10 default and 3 additional upon request), for a total of up to 32 different metallicity measurement outputs, since a number of scales produce different metallicity estimates based on different input line ratios. These estimates are based on the original KD02IDL code in **?** and expanded to include more recently developed scales, and to generate confidence intervals for each scale via MC sampling. We supply visualization tools that, as part of the default output of `pyMCZ`, enable the user to assess the validity of each derived metallicity distribution and understand when line flux measurements may lead to misleading metallicity estimates, for example in proximity of the demarcation between the upper and lower branch for R_{23}-based methods. `pyMCZ` outputs the full estimated metallicity distribution (on demand), and its Kernel Density, and it offers visualization tools to assess the spread of the oxygen abundance in the different scales.

The validity of our metallicity measurements and of their confidence regions hinges upon generating probability distributions that properly sample the metallicity distribution, given the input parameters and the specific metallicity calculation algorithm. Thus, we have developed metrics that allow the user to ascertain that the sample drawn in the simulation is sufficiently large. However, we note that our code does not take into account systematic errors, but only computes statistical ones.

This code is open access and we welcome input and further development from the community. Since the code is open-source, the users can include any new metallicity scales and modify any parts of the algorithms, or any assumption (e.g. that the line flux errors are Gaussian distributed).

We hope that this open-access code will be helpful for the many different fields where gas-phase metallicities are important, including in the emerging field of SN and GRB host galaxies.

We thank John Moustakas for insightful discussions. The Modjaz SNYU group at NYU is supported in parts by the NSF CAREER award AST-1352405 and by NSF award AST-1413260. F. B. Bianco is supported by a *James Arthur Fellowship* at the NYU-Center for Cosmology and Particle Physics and Y. Liu by a *James Arthur Graduate Award*. This code made use of several Python Modules, including `Matplotlib` . Some plots were produced with public code DOI:10.5281/zenodo.15419 available at `https://github.com/fedhere/residuals_pylab`. This research made use of NASA Astrophysics Data System; the NASA/IPAC Extragalactic Database (NED), which is operated by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.