The Astronomical Journal, 165:25 (15pp), 2023 January https://doi.org/10.3847/1538-3881/ac9b1f

© 2022. The Author(s). Published by the American Astronomical Society.

OPEN ACCESS

# The PhotoDissociation Region Toolbox: Software and Models for Astrophysical Analysis

Marc W. Pound and Mark G. Wolfire

Astronomy Department University of Maryland College Park, MD 20742, USA; mpound@umd.edu, mwolfire@umd.edu

*Received 2022 September 2; revised 2022 October 14; accepted 2022 October 14; published 2022 December 20*

## Abstract

The PhotoDissociation Region Toolbox provides comprehensive, easy-to-use, public software tools and models that enable an understanding of the interaction of the light of young, luminous, massive stars with the gas and dust in the Milky Way and in other galaxies. It consists of an open-source Python toolkit and photodissociation region (PDR) models for analysis of infrared and millimeter/submillimeter line and continuum observations obtained by ground-based and suborbital telescopes, and astrophysics space missions. PDRs include all of the neutral gas in the interstellar medium where far-ultraviolet photons dominate the chemistry and/or heating. In regions of massive star formation, PDRs are created at the boundaries between the H II regions and neutral molecular cloud, as photons with energies $6\,\mathrm{eV} < h\nu < 13.6\,\mathrm{eV}$ photodissociate molecules and photoionize metals. The gas is heated by photoelectrons from small grains and large molecules and cools mostly through far-infrared (FIR) fine-structure lines like [O I] and [C II]. The models are created from state-of-the art PDR codes that include molecular freeze-out; recent collision, chemical, and photorates; new chemical pathways, such as oxygen chemistry; and allow for both clumpy and uniform media. The models predict the emergent intensities of many spectral lines and FIR continuum. The tools find the best-fit models to the observations and provide insight into the physical conditions and chemical makeup of the gas and dust. The PDR Toolbox enables novel analysis of data from telescopes such as the Infrared Space Observatory, Spitzer, Herschel, the Stratospheric Terahertz Observatory, the Stratospheric Observatory for Infrared Astronomy, the Submillimeter Wave Astronomy Satellite, the Atacama Pathfinder Experiment, the Atacama Large Millimeter/submillimeter Array, and the JWST.

*Unified Astronomy Thesaurus concepts:* Photodissociation regions (1223); Astronomy software (1855); Molecular gas (1073); Interstellar atomic gas (833)

## 1. Introduction

Over 20 years ago, we created the PhotoDissociation Region Toolbox (PDRT), a web-based interface that allowed users to analyze the line and continuum emission from photodissociation regions (PDRs; Pound & Wolfire 2008). Back then, web programming meant Common Gateway Interface (CGI) and Perl was the workhorse scripting language. Single-pixel detectors were cutting-edge technology, and the submillimeter window had just begun to be explored. We put together PDRT with Perl, HTML, Apache 1.3, FITS files, Concurrent Versioning System, shell scripts, and a sense of humor. PDRT became a leading on-line site for analyzing PDRs and developed an international user base with users in over 35 countries. It garnered many refereed citations, and the output plots have been used directly in published papers, and in posters and presentations, As new telescopes arrived, we added spectral lines and low-metallicity models. Browser-free scripting interfaces were created by users. Although funding for the project ran out, we continued to add spectral lines when users requested and kept the service running. Single-pixel detectors gave way to cameras, and the submillimeter science matured. Now, thanks to renewed funding, we have rebuilt PDRT as an open-source Python 3 package called `pdrtpy` with far more capabilities than the original CGI scripts. The PDR Toolbox website[1] remains the central clearing house to keep users apprised of our work, with the code now moved to GitHub. The `pdrtpy` version at the time of writing is 2.2.9.

In this paper, we describe the scientific motivation to develop `pdrtpy`, its architecture, and primary capabilities. In Section 2, we discuss the astrophysics of PDRs. Section 3 describes our development paradigm. In Section 4, we describe the modeling physics and code. In Section 5 we review `pdrtpy`'s core capabilities, and in Section 6 we outline the development we would like to pursue in the near future. To Improve readability, example code listings are given in the Appendix rather than in the main text. The code listings are downloadable[2] and can be used to reproduce Figures 2–9 in this paper.

## 2. The Importance of PDRs

The interstellar medium (ISM) plays a central role in the lifecycle of stars and galaxies. The coldest phases of the ISM, the molecular clouds, give rise to star formation. Stars return energy to their surroundings in the form of photons and kinetic energy from winds and supernovae explosions. In addition, stars enrich the ISM with metals that affect the gas cooling. Thus, understanding the production, chemistry, thermal balance, and evolution of the ISM is essential to understanding star formation and the evolution of galaxies.

The infrared line and continuum emission from atoms, molecules, and dust provide the observational diagnostics of

Original content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

[1] https://dustem.astro.umd.edu

[2] Digital Repository at the University of Maryland identifier http://hdl.handle.net/1903/29105.

the stellar feedback. The line and continuum emission arise primarily from PDRs. Classical PDRs are largely neutral regions that are photodissociated and heated by far-ultraviolet (FUV; $6\,\mathrm{eV} < h\nu < 13.6$ eV) photons from nearby massive stars (Tielens & Hollenbach 1985; Wolfire et al. 2022). The same physical and chemical processes that dissociate, partially ionize, and heat the gas in these classical PDRs are also Important for the ISM as a whole, and PDRs are now generally taken to include all regions where FUV photons play an important role in the chemistry and/or heating of the gas. These regions include diffuse and translucent H I clouds and the warm neutral medium in the ISM (Wolfire et al. 2003), the surfaces of molecular clouds illuminated by the ambient radiation field and by nearby stars (Kaufman et al. 2006), the warm dust envelopes surrounding newly formed stars (e.g., Visser et al. 2012), and the neutral ISM in the disk and nuclei of normal and starburst galaxies (Roussel et al. 2007; Kennicutt et al. 2011; Herrera-Camus et al. 2015). Most of the nonstellar baryons within galaxies are in PDRs.

While gaining a conceptual understanding of PDRs is relatively easy, extracting the physical conditions and underlying physics from the observations is difficult. For example, observers often use a simple large velocity gradient (LVG) model to analyze line ratios. Such a model is appropriate for emission from large clouds where there is an overall velocity gradient but not appropriate for individual star-forming regions where PDR lines arise. In addition, LVG models typically do not account for the temperature distribution in thermal equilibrium and abundance profiles in chemical balance through the emitting layer. See Wolfire et al. (2022) for a review citing additional examples of PDRs, PDR models, and their applications.

## 3. Development Philosophy and Framework

Our approach is open-source development, user-friendliness, sensible and consistent interfaces, good documentation and examples, and responsive user support. We use a GitHub open-source public repository[3] that includes the Python code, text documentation files, and FITS files of the models. The model FITS files can also be browsed and downloaded from the PDR Toolbox website. Using GitHub Actions at code check-in, the code is checked against the PEP8 coding style,[4] regression and integration tests are run, and the code coverage of tests is calculated. A separate repository contains Jupyter[5] notebooks[6] that demonstrate how to do analysis with `pdrtpy`. (Notebooks are convenient but not required). The repositories are governed by a GPL3 license. We make use of major Python libraries such as `astropy` (Astropy Collaboration et al. 2013, 2018), `lmfit-py` (Newville et al. 2021), `matplotlib` (Hunter 2007), `numpy` (van der Walt et al. 2011), `scipy` (Virtanen et al. 2020), and `emcee` (Foreman-Mackey et al. 2013). The full list of dependencies is given in the repository. `pdrtpy` is installed via `pip` or by cloning the Git repository. We use `sphinx` [7] to generate documentation from code comments and text files, which is then hosted on Read The Docs.[8] Where applicable, we make an effort to "promote" keywords to our APIs from, e.g., `matplotlib` and `astropy`, which many users will already be familiar with. This can be especially helpful in creating plots where users may want more fine-grained control than our default plots, which we already strive to make publication quality.

The fitting tools in the Toolbox inherit from a common parent class `ToolBase`, which defines a few common attributes and properties, and the `run()` interface, which all child classes must implement. The workflow for the user is to instantiate a fitting tool, optionally set some attributes, and invoke `run()` to perform the fit. Subsequently, users instantiate the companion plotting tool, which is similarly derived from a parent `PlotBase`, to explore the fit results.

As astronomers who research PDRs, we have a reasonable idea of the kinds of tools that users need but welcome suggestions for desired functionality. For example, model phase-space plotting with data overlay was requested by members of the Stratospheric Observatory for Infrared Astronomy (SOFIA) FEEDBACK (Schneider et al. 2020) team, so we prioritized its development and worked with them to beta test and to refine its functionality. It found immediate use in publications and talks (Tiwari 2022; Tiwari et al. 2022).

## 4. The PDR Models and Model Codes

In the `pdrtpy` distribution, all models are precomputed from PDR model codes, currently either the "Wolfire–Kaufman" code, which we have developed, or the KOSMA-tau code (Rollig et al. 2013; Röllig & Ossenkopf-Okada 2022). The models are computed using a given set of parameters (Table 1) and presented as grids of intensity or intensity ratio as a function of the hydrogen nucleus density $n$ and radiation field strength $F_{\mathrm{FUV}}$. The results, collectively called `ModelSet`, are stored as FITS images in subdirectories organized by the modeling code origin and major parameters such as metallicity. A list of the available `ModelSets` is given in Table 2.

The current set of distributed models, both Wolfire–Kaufman and KOSMA-tau, are most appropriate for the "classical" PDRs described in Wolfire et al. (2022), where stars illuminate nearby molecular clouds. The maximum depths from the cloud surface are larger than those found in diffuse or translucent molecular clouds, where $A_V \sim 1$–2, and the illumination is only on the front side where for diffuse clouds the illumination would be on both the front and back sides. Although a soft ($E < 100$ eV) X-ray spectrum is included in the Wolfire–Kaufman PDR code, neither set of PDR models is appropriate for gas illuminated by hard X-ray radiation as would be emitted by an active galactic nucleus (see also Wolfire et al. (2022) for a comparison of PDR-dominated and X-ray-dominated models). The models cover a wide range of spectral lines that can be observed by many different telescopes (Figure 1). The telescopes listed in Figure 1 are not an exhaustive list. Other telescopes such as the Atacama Pathfinder Experiment, the Kölner Observatorium für Submillimeter Astronomie, the Antarctic Submillimeter Telescope and Remote Observatory, and the Heinrich Hertz Telescope have observed CO and CI, and high-$z$ spectral lines can be redshifted into observable bands of existing instruments.

The Wolfire–Kaufman PDR model code is based on the work of Tielens & Hollenbach (1985) but with many updates since the early versions. It assumes a plane-parallel geometry with a UV radiation field, cosmic rays, and soft X-rays incident

[3] https://github.com/mpound/pdrtpy
[4] https://peps.python.org/pep-0008/
[5] https://jupyter.org
[6] https://github.com/mpound/pdrtpy-nb
[7] https://www.sphinx-doc.org/
[8] https://pdrtpy.readthedocs.io

**Table 1**
Example Parameters of PDR Models

| Parameter | Units | Symbol | WK2020 | KOSMA-tau |
|---|---|---|---|---|
| (1) | (2) | (3) | (4) | (5) |
| Carbon abundance | | $X_C$ | $1.6 \times 10^{-4}$ | $2.34 \times 10^{-4}$ |
| $^{13}$C abundance | | $X_{13C}$ | $3.2 \times 10^{-6}$ | $3.2 \times 10^{-6}$ |
| Oxygen abundance | | $X_O$ | $3.2 \times 10^{-4}$ | $4.47 \times 10^{-4}$ |
| $^{18}$O abundance | | $X_{18O}$ | … | $8.93 \times 10^{-7}$ |
| Silicon abundance | | $X_{Si}$ | $1.7 \times 10^{-6}$ | $3.17 \times 10^{-6}$ |
| Sulfur abundance | | $X_S$ | $2.8 \times 10^{-5}$ | $7.41 \times 10^{-6}$ |
| Iron abundance | | $X_{Fe}$ | $1.7 \times 10^{-7}$ | $1.0 \times 10^{-6}$ |
| Magnesium abundance | | $X_{Mg}$ | $1.1 \times 10^{-6}$ | $3.2 \times 10^{-6}$ |
| Nitrogen abundance | | $X_N$ | 0 | $8.32 \times 10^{-5}$ |
| Fluorine abundance | | $X_F$ | $1.8 \times 10^{-8}$ | $6.68 \times 10^{-9}$ |
| Helium abundance | | $X_{He}$ | 0.1 | $8.51 \times 10^{-2}$ |
| PAH abundance | | $X_{\rm PAH}$ | $2 \times 10^{-7}$ | …[a] |
| Dust and metals with respect to local ISM | | Z | 1 | 1 |
| Dust abundance relative to diffuse ISM | | $\delta_d$ | 1 | 1 |
| FUV dust opacity/visual extinction | | $\tau_{\rm FUV}/\tau_V$ | 1.8 | …[b] |
| Maximum optical depth | | $A_{V,Max}$ | 7 | …[c] |
| Dust visual extinction per H atom | cm$^{-2}$ | $\sigma_V$ | $5.26 \times 10^{-22}$ | …[d] |
| Formation rate of $H_2$ on dust | s$^{-1}$ | $R_{form}$ | $6 \times 10^{-17}$ | …[e] |
| Turbulent Doppler velocity | km s$^{-1}$ | $\delta v_D$ | 1.5 | …[f] |
| Cosmic ray ionization rate per H nucleus | s$^{-1}$ | $\zeta_{CR}$ | $2.0 \times 10^{-16}$ [g] | $2.0 \times 10^{-16}$ |
| Cloud H density | cm$^{-3}$ | $n$ | $10^1 - 10^7$ | $10^3 - 10^7$[h] |
| Incident UV flux[i] | erg cm$^{-2}$ s$^{-1}$ | $F_{\rm FUV}$ | $10^{-3.3} - 10^{3.7}$ | $10^{-2.5} - 10^{3.4}$ |

**Notes.**

[a] Following the Weingartner & Draine (2001a) prescription, the PAH abundance is not specified.

[b] Depends on the dust model; see Table 4 of Rollig et al. (2013).

[c] Depends on the mass and density of the model.

[d] Depends on the dust model: WD01-7: $5.24 \times 10^{-22}$, WD01-21: $5.05 \times 10^{-22}$, WD02-25: $4.88e \times 10^{-22}$.

[e] Computed following Cazaux & Tielens (2004, 2010).

[f] Doppler velocity computed from the Larson (1981) mass–line width relation.

[g] Assumes that the ionization rate falls as $\zeta_{CR}/(1 + N/1.0 \times 10^{21}\ {\rm cm}^{-2})$ with a minimum of $2.0 \times 10^{-17}$ s$^{-1}$.

[h] This is the density at the surface. KOSMA-tau assumes a certain profile, typically leading to a central density ~11 times higher and a mean density that is ~1.9 times the surface density.

[i] Draine (1978) spectral energy distribution.

on one side. The main input parameters are the radiation field strength in units of Habing (Habing 1968) fields ($G_0$) and a constant hydrogen nucleus density $n$. Alternatively, the density can be derived self-consistently from an input pressure. The code finds the gas temperature in thermal equilibrium and abundances of atomic and molecular species in chemical balance. The non-LTE level populations are calculated for the dominant coolants and the emitted line intensities are found using an escape probability formalism. Updates to the code are described in Wolfire et al. (2010), Kaufman et al. (2006), Hollenbach et al. (2012), and Neufeld & Wolfire (2016). More recent updates, and in particular those included in the "wk2020" models, feature the photorates and dependence with the depth as given in Heays et al. (2017), $^{13}$C chemistry and line emission, and O collision rates from Lique et al. (2018) as given in the MOLCAT (Schoier et al. 2005) database.

In addition to the hydrogen density and radiation field strength, a large number of parameters could potentially be varied including the gas-phase metallicity, the abundance of grains and large molecules, the microturbulent line width, and the PDR depth. The values for these parameters are given in Table 1 and are discussed in more detail in previous papers. Note that in the "wk2020" models we adopt a lower PDR depth ($A_{V,Max} = 7$) compared to previous models to avoid possible time-dependent effects in the deeper layers. We also turn off chemistry on grain surfaces—a constraint that will be lifted in future models. Similarly, sets of models with low $A_V$, appropriate for diffuse or translucent clouds, can be computed from this PDR model code.

The KOSMA-tau models and PDR model code are described in Rollig et al. (2013) and Röllig & Ossenkopf-Okada (2022). The geometry of the KOSMA-tau models differ from those of the Wolfire–Kaufman code. Instead of a plane-parallel geometry, KOSMA-tau uses an ensemble of spherical clumps with a spectrum of masses ("clumpy") or a single clump ("nonclumpy"). Further, whereas the Wolfire–Kaufman code has a fixed incident spectral energy distribution (that of the interstellar radiation field) and grain model (interstellar medium grains with $R_V = 3.1$), the KOSMA-tau code can independently vary them (see Table 2). We were provided with FITS files of model spectral line intensity ratios and intensities by the KOSMA-tau authors for use in PDRT.

The choice of PDR model can have significant effects on the predicted line intensities (see Figure 2). This can give physical insight into the PDR conditions, for instance, whether the data are better represented by a clumpy or plane-parallel medium.

## 5. Capabilities

### 5.1. Data Representation

Observations in `pdrtpy` are represented by the `Measurement` class. A `Measurement` consists of a value and an error.

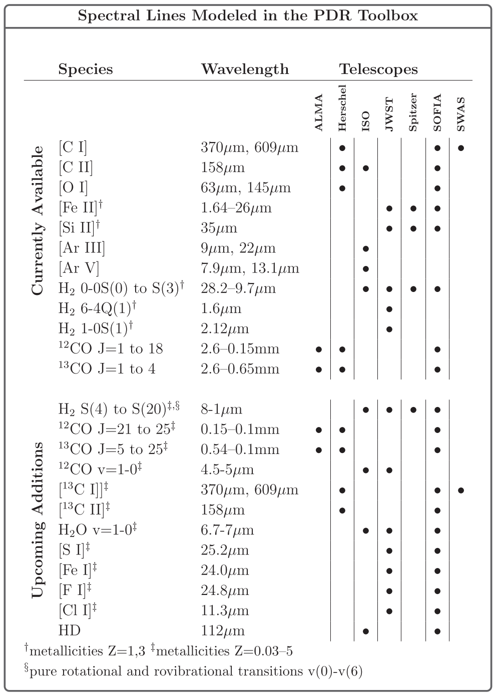
**Table 2**
PDR Models Currently Supported

| PDR Code | Name | Version | Medium | $Z$ | Mass[a] | $R_V$[b] |
|---|---|---|---|---|---|---|
| | | | | | $M_\odot$ | |
| (1) | (2) | (3) | (4) | (5) | (6) | (7) |
| Wolfire/Kaufman | wk2006 | 2006 | constant density | 1.0 | ⋯ | 3.1 |
| Wolfire/Kaufman | wk2006 | 2006 | constant density | 3.0 | ⋯ | 3.1 |
| Wolfire/Kaufman | wk2020 | 2020 | constant density | 1.0 | ⋯ | 3.1 |
| Wolfire/Kaufman | smc[c] | 2006 | constant density | 0.1 | ⋯ | 3.1 |
| Wolfire/Kaufman | lmc[d] | 2020 | constant density | 0.5 | ⋯ | 3.1 |
| KOSMA-tau | kt2013wd01-7 | WD01-7 | clumpy | 1.0 | 100.0 | 3.1 |
| KOSMA-tau | kt2013wd01-7 | WD01-7 | non-clumpy | 1.0 | 100.0 | 3.1 |
| KOSMA-tau | kt2013wd01-21 | WD01-21 | clumpy | 1.0 | 100.0 | 4.0 |
| KOSMA-tau | kt2013wd01-21 | WD01-21 | non-clumpy | 1.0 | 100.0 | 4.0 |
| KOSMA-tau | kt2013wd01-25 | WD01-25 | clumpy | 1.0 | 100.0 | 5.5 |
| KOSMA-tau | kt2013wd01-25 | WD01-25 | non-clumpy | 1.0 | 100.0 | 5.5 |
| KOSMA-tau | kt2013wd01-7 | WD01-7 | clumpy | 1.0 | 10.0 | 3.1 |
| KOSMA-tau | kt2013wd01-7 | WD01-7 | non-clumpy | 1.0 | 10.0 | 3.1 |
| KOSMA-tau | kt2013wd01-21 | WD01-21 | clumpy | 1.0 | 10.0 | 4.0 |
| KOSMA-tau | kt2013wd01-21 | WD01-21 | non-clumpy | 1.0 | 10.0 | 4.0 |
| KOSMA-tau | kt2013wd01-25 | WD01-25 | clumpy | 1.0 | 10.0 | 5.5 |
| KOSMA-tau | kt2013wd01-25 | WD01-25 | non-clumpy | 1.0 | 10.0 | 5.5 |
| KOSMA-tau | kt2013wd01-7 | WD01-7 | clumpy | 1.0 | 1.0 | 3.1 |
| KOSMA-tau | kt2013wd01-7 | WD01-7 | non-clumpy | 1.0 | 1.0 | 3.1 |
| KOSMA-tau | kt2013wd01-21 | WD01-21 | clumpy | 1.0 | 1.0 | 4.0 |
| KOSMA-tau | kt2013wd01-21 | WD01-21 | non-clumpy | 1.0 | 1.0 | 4.0 |
| KOSMA-tau | kt2013wd01-25 | WD01-25 | clumpy | 1.0 | 1.0 | 5.5 |
| KOSMA-tau | kt2013wd01-25 | WD01-25 | non-clumpy | 1.0 | 1.0 | 5.5 |
| KOSMA-tau | kt2013wd01-7 | WD01-7 | clumpy | 1.0 | 0.1 | 3.1 |
| KOSMA-tau | kt2013wd01-7 | WD01-7 | non-clumpy | 1.0 | 0.1 | 3.1 |
| KOSMA-tau | kt2013wd01-21 | WD01-21 | clumpy | 1.0 | 0.1 | 4.0 |
| KOSMA-tau | kt2013wd01-21 | WD01-21 | non-clumpy | 1.0 | 0.1 | 4.0 |
| KOSMA-tau | kt2013wd01-25 | WD01-25 | clumpy | 1.0 | 0.1 | 5.5 |
| KOSMA-tau | kt2013wd01-25 | WD01-25 | non-clumpy | 1.0 | 0.1 | 5.5 |
| KOSMA-tau | kt2013wd01-7 | WD015.5 | clumpy | 1.0 | 1000.0 | 3.1 |
| KOSMA-tau | kt2013wd01-7 | WD01-7 | non-clumpy | 1.0 | 1000.0 | 3.1 |
| KOSMA-tau | kt2013wd01-21 | WD01-21 | clumpy | 1.0 | 1000.0 | 4.0 |
| KOSMA-tau | kt2013wd01-21 | WD01-21 | non-clumpy | 1.0 | 1000.0 | 4.0 |
| KOSMA-tau | kt2013wd01-25 | WD01-25 | clumpy | 1.0 | 1000.0 | 5.5 |
| KOSMA-tau | kt2013wd01-25 | WD01-25 | non-clumpy | 1.0 | 1000.0 | 5.5 |

**Notes.**
[a] For clumpy models, this is the maximum clump mass. For non-clumpy models, it is the mass of a single spherical clump.
[b] KOSMA-tau models use dust properties from Weingartner (2001b).
[c] Limited set of models for the Small Magellanic Cloud.
[d] Limited set of models for the Large Magellanic Cloud.

These can be single-valued, an array of values, or an image and can be in intensity units (equivalent to erg $\mathrm{cm}^{-2}\,\mathrm{s}^{-1}\,\mathrm{sr}^{-1}$) or in K km $\mathrm{s}^{-1}$, which is typical of millimeter/submillimeter spectral line observations. In the typical case of an image observation, the `Measurement` is a representation of a FITS file with two header data units (HDUs): the first HDU is the spatial map of intensity and the second HDU is the spatial map of the errors. An image-based `Measurement` carries a world coordinate system and traditional FITS-like header. `Measurement` is based on the `CCDData` class of `astropy` with additional properties such as beam parameters and support for arithmetic operators. In arithmetic operations, errors and units are correctly propagated through the underlying `astropy` code. Users identify their `Measurements` with one of `pdrtpy`'s predefined ID strings, e.g., "CII_158" for the [C II] 158 $\mu$m spectral line. These identifiers are used by `pdrtpy` to match observations with models that are similarly identified.

The models in `pdrtpy` are two-dimensional grids of either intensities or ratios of intensities as a function of the hydrogen nucleus volume density ($\mathrm{cm}^{-3}$) and incident FUV field, $F_{\mathrm{FUV}}$ (erg $\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ or equivalent). As these are stored on disk as FITS images, they are also represented internally as `Measurements`, but with no errors. Because of the built-in operator support, this makes straightforward arithmetic operations that involve both observations and models. There are different conventions for the units of $F_{\mathrm{FUV}}$ depending on different approximations to the local interstellar radiation field–cgs units, Habing units (Habing 1968; 1 Habing $= G_0 = 1.63 \times 10^{-3}$ erg $\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$), Draine units (Draine 1978; 1 Draine $= \chi = 2.72 \times 10^{-3}$ erg $\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$), and Mathis units (Mathis et al. 1983; 1 Mathis $= 1.81 \times 10^{-3}$ erg $\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$); `pdrtpy` defines each as an astropy `Quantity` to convert seamlessly between them. This allows the user to, for instance, create plots in their preferred unit (see, e.g., Listing A.1 and Figure 2).



Spectral Lines Modeled in the PDR Toolbox

| | Species | Wavelength | ALMA | Herschel | ISO | JWST | Spitzer | SOFIA | SWAS |
|---|---|---|---|---|---|---|---|---|---|
| Currently Available | [C I] | 370$\mu$m, 609$\mu$m | | ● | | | | ● | ● |
| | [C II] | 158$\mu$m | | ● | ● | | | ● | |
| | [O I] | 63$\mu$m, 145$\mu$m | | ● | | | | ● | |
| | [Fe II]$^{\dagger}$ | 1.64–26$\mu$m | | | | ● | ● | ● | |
| | [Si II]$^{\dagger}$ | 35$\mu$m | | | | ● | ● | ● | |
| | [Ar III] | 9$\mu$m, 22$\mu$m | | | ● | | | | |
| | [Ar V] | 7.9$\mu$m, 13.1$\mu$m | | | ● | | | | |
| | $H_2$ 0-0S(0) to S(3)$^{\dagger}$ | 28.2–9.7$\mu$m | | | ● | ● | ● | ● | |
| | $H_2$ 6-4Q(1)$^{\dagger}$ | 1.6$\mu$m | | | | ● | | | |
| | $H_2$ 1-0S(1)$^{\dagger}$ | 2.12$\mu$m | | | | ● | | | |
| | $^{12}$CO J=1 to 18 | 2.6–0.15mm | ● | ● | | | | ● | |
| | $^{13}$CO J=1 to 4 | 2.6–0.65mm | ● | ● | | | | ● | |
| Upcoming Additions | $H_2$ S(4) to S(20)$^{\ddagger,\S}$ | 8-1$\mu$m | | | ● | ● | ● | ● | |
| | $^{12}$CO J=21 to 25$^{\ddagger}$ | 0.15–0.1mm | ● | ● | | | | ● | |
| | $^{13}$CO J=5 to 25$^{\ddagger}$ | 0.54–0.1mm | ● | ● | | | | ● | |
| | $^{12}$CO v=1-0$^{\ddagger}$ | 4.5-5$\mu$m | | | ● | ● | | | |
| | [$^{13}$C I]$^{\ddagger}$ | 370$\mu$m, 609$\mu$m | | ● | | | | ● | ● |
| | [$^{13}$C II]$^{\ddagger}$ | 158$\mu$m | | ● | | | | ● | |
| | $H_2O$ v=1-0$^{\ddagger}$ | 6.7-7$\mu$m | | | ● | ● | | ● | |
| | [S I]$^{\ddagger}$ | 25.2$\mu$m | | | | ● | | ● | |
| | [Fe I]$^{\ddagger}$ | 24.0$\mu$m | | | | ● | | ● | |
| | [F I]$^{\ddagger}$ | 24.8$\mu$m | | | | ● | | ● | |
| | [Cl I]$^{\ddagger}$ | 11.3$\mu$m | | | | ● | | ● | |
| | HD | 112$\mu$m | | | ● | | | ● | |

$^{\dagger}$metallicities Z=1,3 $^{\ddagger}$metallicities Z=0.03–5

$^{\S}$pure rotational and rovibrational transitions v(0)-v(6)

**Figure 1.** Spectral lines and metallicities currently available in PDRT and the upcoming additions. The Species column lists the spectral line designation; Wavelength gives the rest wavelength range covered by the models for the line (s). A dot in a Telescope column means that spectral line is observable with a given telescope (not including lines highly redshifted into the telescope bands).

Collections of models are managed by the `ModelSet` class. A `ModelSet` represents a coherent collection of models that were created using the same modeling code and physical parameters (e.g., Table 1). The list of currently available `ModelSets` is given in Table 2. Users retrieve individual models from a `ModelSet` using their identifiers. Listing A.1 shows an example of instantiating a `ModelSet`, retrieving individual models from it, and plotting a model (Figure 2).

Another way to visualize the models is through a phase-space diagram that can plot lines of constant $n$ and $F_{\rm FUV}$ as a function of the spectral line intensity or intensity ratio. Adding observed data to the plot lets the astronomer understand the conditions in different regions, as was done by Tiwari et al. (2021) for RCW 49 (Listing A.1, Figure 3). Phase-space diagrams can also be useful for making predictions of the line strength or estimating density and radiation field when the user does not have enough observations to fit with `LineRatioFit`.

### 5.2. Fitting Observations and Plotting Results

#### 5.2.1. Intensity Ratios

It has been shown that far-infrared line and continuum observations can be used to determine the physical properties of PDRs including the incident FUV radiation field, the gas density, and the surface temperature (Wolfire et al. 1990; Kaufman et al. 1999, 2006). These authors showed that the ratios of intensities are particularly effective for determining $n$ and $F_{\rm FUV}$ as to the first order the beam filling factors cancel out.[9]

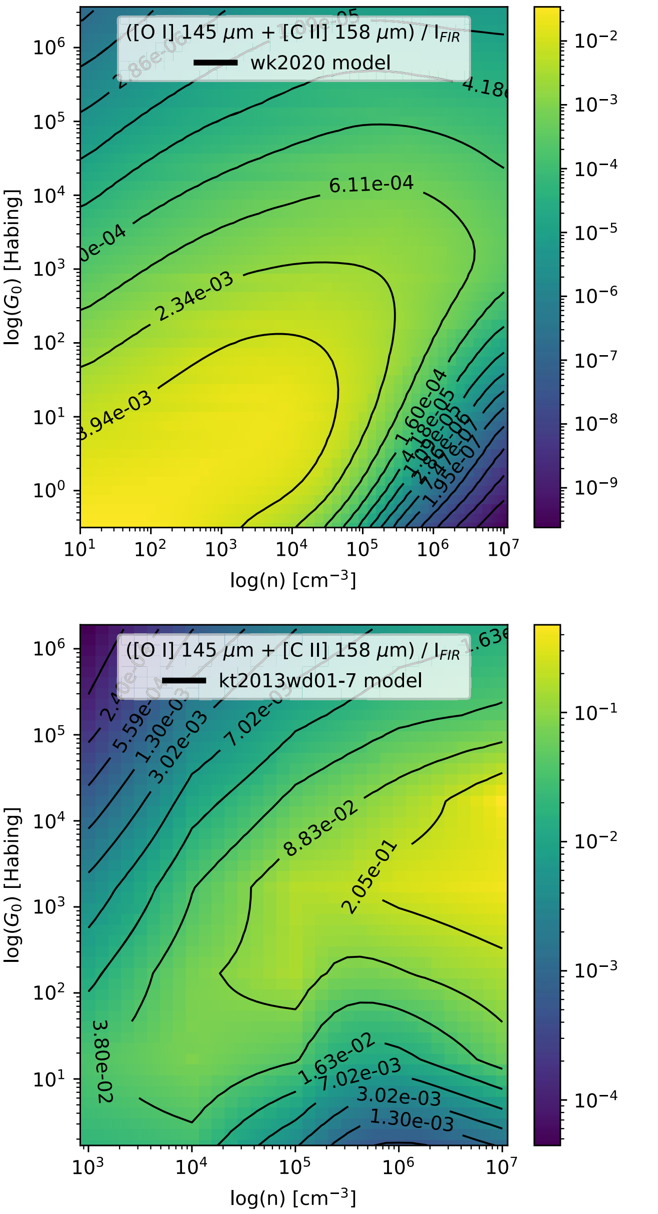


**Figure 2.** Examples of model plots. (top) WK2020 model for the ratio of the sum of the [O I] 145 $\mu$m and [C II] 158 $\mu$m intensities divided by the far-infrared intensity integrated between 8 $\mu$m and 1 mm, $I_{\rm FIR}$, computed as a function of the H nucleus density $n$ and FUV field $G_0$. (bottom) Same model intensity ratio as that computed in the kt2013wd01-7, clumpy, $M = 100\ M_{\odot}$, $R_V = 3.1$ model. The user-friendly flexibility of `pdrtpy` allows the choice of Habing units for $G_0$, log normalization for the image intensities with a color-blind friendly color map, and labeled black contours. See Listing A.1.

In `pdrtpy`, the `LineRatioFit` tool takes intensity `Measurements` and a `ModelSet` as input, computes the intensity ratios that have entries in the `ModelSet`, and finds the best-fit $n$ and $F_{\rm FUV}$. The fit result matches the input—single value, array, or spatial image. The available fitting algorithms are nonlinear least-squares (NLS) minimization or the Monte Carlo Markov Chain (MCMC) to determine the posterior probability density function (PDF) of the fitted parameters. Both are managed through `lmfit`, which capably delegates via easy-to-use high-level interfaces to `scipy.optimize` for NLS or `emcee` for MCMC.

Listing A.2 gives an example of determining $n$ and $F_{\rm FUV}$ from single-pixel (or single-beam) observations using `LineRatiofit`

[9] We note that, for unresolved observations, the beam filling factors may not cancel, and an additional correction to normalize the filling factors may be needed. See the detailed procedure given in Kaufman et al. (2006) and additional caveats in comparing models with observations in Wolfire et al. (2022).

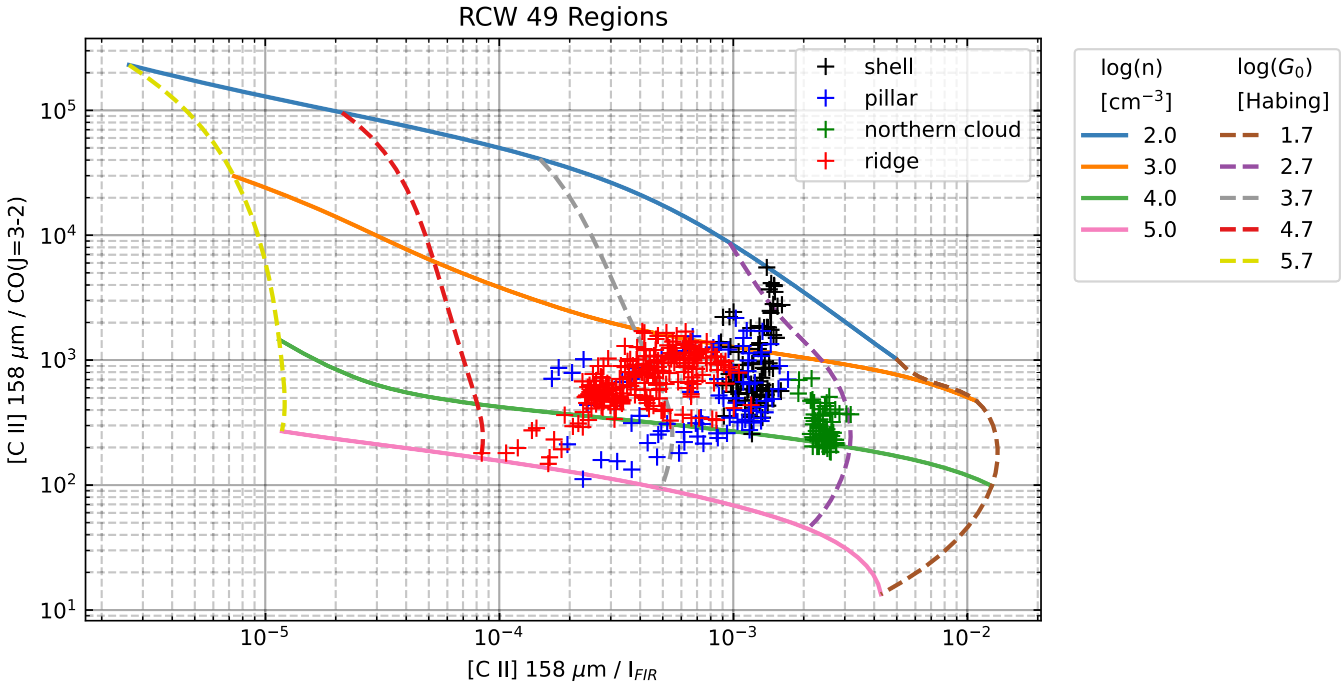


**Figure 3.** An example of a phase-space plot showing lines of constant $n$ and $G_0$ as a function of the spectral line intensity ratios. The plot uses WK2020 models and [C II], CO(3-2), and FIR data in the RCW 49 PDR from Tiwari et al. (2021). The crosses are the observed [C II]/FIR and [C II]/CO(3-2) intensity ratios for four regions in RCW 49. See Listing A.1.

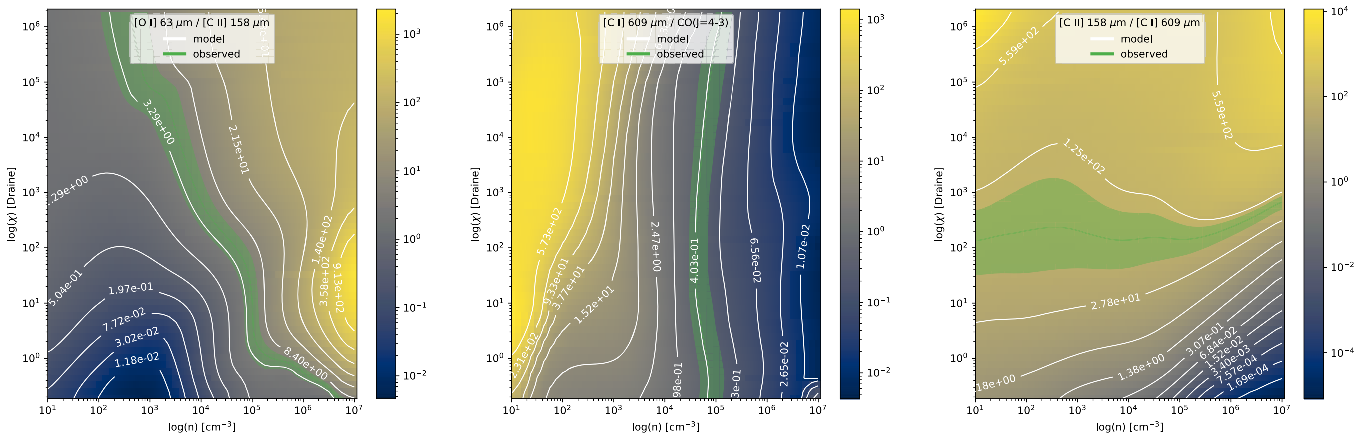


**Figure 4.** Plots created in Listing A.2 using the `LineRatioPlot.ratios_on_models` method showing the observed ratios with errors overlaid on the matching models. The observational errors ($1\sigma$) are shown as shaded regions around the solid observation line. Axis units, colors, contours, and other plot parameters can be modified by the user via the API. The data are values chosen for demonstration purposes.

and plotting the results with `LineRatioPlot`. Integrated intensity observations of [O I] 63 $\mu$m, [C I] 609 $\mu$m, CO($J=$ 4–3), and [C II] 158 $\mu$m are used to create three ratios, and the `run()` method invokes NLS minimization to determine the best-fit quantities. The results can be inspected with print statements, ratio plots (Figure 4), overlay plots, and chi-square plots (Figure 5). In Listing A.3, we show how to fit $n$ and $F_{\rm FUV}$ using MCMC and how to create the traditional corner plot with the desired axes (Figure 6).

Listing A.3 shows how to fit the `Measurements` from Listing A.2 using the MCMC method by passing the appropriate arguments to `LineRatioFit.run()`. The `emcee` package is used, and keywords specific to `emcee` can be passed in, e.g., *steps*, indicating how many samples to draw from the posterior distribution for each walker. Listing A.3 also shows how to create a custom corner plot of the results (Figure 6) using the `corner` package (Foreman-Mackey 2016).

One of the significant improvements that `pdrtpy` makes over the old web version is the ability to operate on images, creating maps of the best-fit $n$ and $F_{\rm FUV}$. Listing A.4 (Figure 7) shows an example using the [C II] 158 $\mu$m, [O I] 63 $\mu$m, and far-infrared continuum maps in the Small Magellanic Cloud N22 star-forming region from Jameson et al. (2018). In this example, models computed using a low metallicity ($Z = 0.3$) were used to match the conditions of the SMC. To fit the 4768 nonblanked pixels takes ~17 s in a Jupyter notebook on a late-model laptop using a single CPU. We have not yet implemented multithreading speedups. The fit was done with the NLS method; using `emcee` on so many pixels would be prohibitive (about 14 s per pixel or over 18 hr for the entire map).

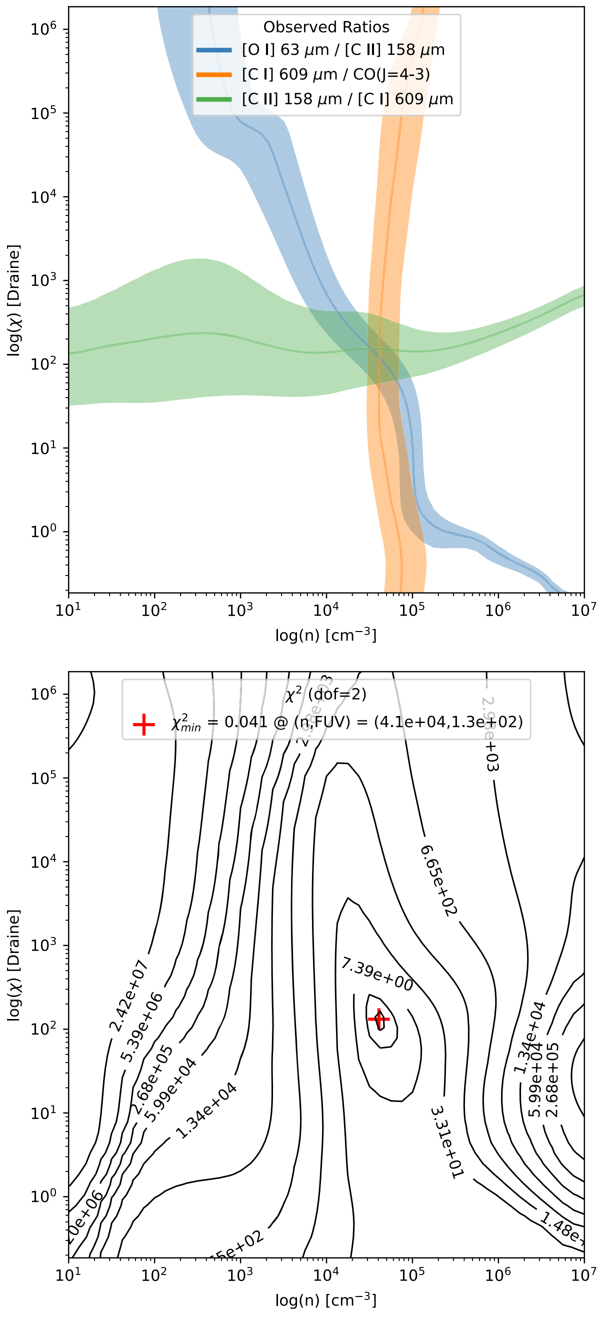


**Figure 5.** Plots created in Listing A.2 using the `LineRatioPlot.overlay_all_ratios` and `LineRatioPlot.chisq` methods. (top) Observed ratios used in the fit overlaid on the model space; solid lines are observed values, and shaded regions are 1$\sigma$ errors. The intersection of the lines indicates the region of the model space where the most likely density and radiation field lie. (bottom) Contour plot of reduced chi-square from the fit. The red cross indicates the minimum $\chi^2$ and best-fit density and radiation field. The plot parameters are modifiable by the user via the API.

### 5.2.2. $H_2$ *Excitation Diagrams*

From the observed, extinction-corrected intensity $I$ of an $H_2$ spectral line we can calculate the column density in the upper state, $N_u$:

$$N_u = 4\pi \frac{I}{A \Delta E}, \qquad (1)$$

where $A$ is the Einstein A coefficient, and $\Delta E$ is the energy difference between the upper and lower states of the transition.

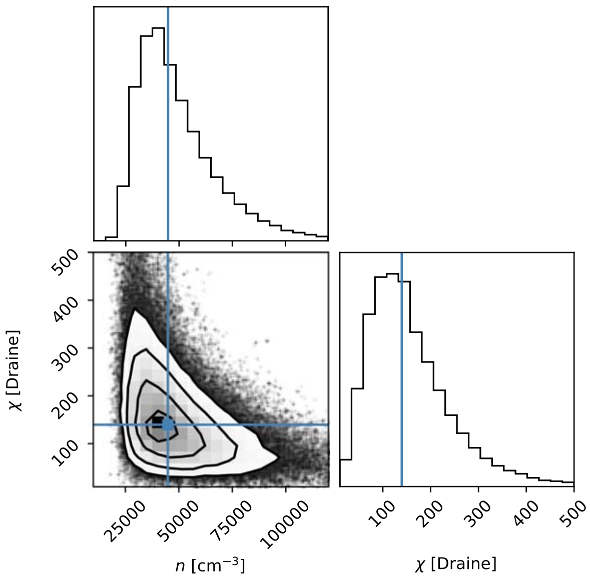


**Figure 6.** Corner plot showing the histogram probability density functions of the density $n$ and radiation field $\chi$ from the MCMC fit in Listing A.3. The blue lines indicate the most probable values.

More typically one is interested in the normalized upper-state column density $N_u/g_u$ for each transition, where $g_u$ is the statistical weight of the upper state.

An excitation diagram plots the upper-state energy of the transition $E_u/k$ on the $x$-axis versus $log(N_u/g_u)$ on the $y$-axis. The statistical weight $g_u = (2I + 1) \times (2J + 1)$, where $I$ is the total nuclear spin and $J$ is the rotational quantum number of the upper level. Ortho hydrogen molecules have the spins of both the nuclei in the same direction $I = 1$, and odd $J$; para molecules have nuclei that spin in opposite directions $I = 0$, and even $J$. In LTE at temperatures $T \gtrsim 200$ K, the ortho-to-para ratio (OPR) of molecules is 3 to 1. In nonequilibrium environments, the OPR can be less than 3, and the actual $N_u/g_u$ will increase over its LTE value (see discussions in Burton et al. 1992 and Sheffer et al. 2011). In such cases, on a traditional plot that assumes OPR = 3, $N_u$ for odd $J$ will be too low. This creates the so-called "zigzag" pattern in the excitation diagram (Fuente et al. 1999; Neufeld et al. 1998; Sternberg & Neufeld 1999).

Often, excitation diagrams show evidence of both "hot" and "cold" gas components, where the "cold" gas dominates the intensity in the low-$J$ transitions and the "hot" gas dominates in the high-$J$ transitions, leading to a curved line in the diagram. Given data over several transitions, one can fit for $T_{\rm cold}$, $T_{\rm hot}$, $N_{\rm total} = N_{\rm cold} + N_{\rm hot}$, and OPR. The $T_{\rm cold}$ is usually a good approximation for the gas kinetic temperature as the lower levels are collisionally excited. The $T_{\rm hot}$ is generally a result of UV fluorescence to the excited levels. One needs at least five data points to fit the temperatures and column densities (slope and intercept × 2), though one could compute (not fit) them with only four points. To additionally fit the OPR six data points are required. The cold, hot, and total column densities are computed using $N_0$ determined from $y$-axis intercepts and the partition function $Z(T) = 0.0247\ T\ [1 - \exp(-6000/T)]^{-1}$, where $T$ is one of $T_{\rm cold}$ or $T_{\rm hot}$ (Herbst et al. 1996). As with the $n$ and $F_{\rm FUV}$ fitting, the user can fit single pixels or full maps. For $H_2$ map inputs, PDRT will fit the excitation diagram at every pixel and produce maps of $T_{\rm cold}$, $T_{\rm hot}$, $N_{\rm total}$, $N_{\rm cold}$, $N_{\rm hot}$, and OPR. The

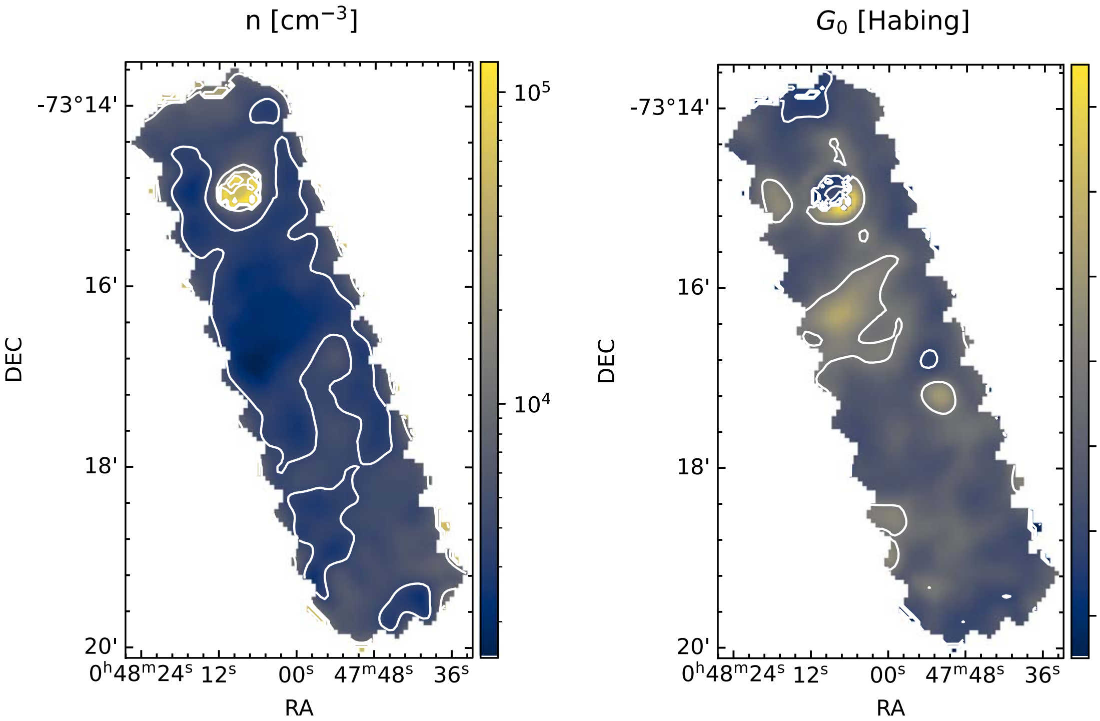


**Figure 7.** Maps of the hydrogen nucleus density $n$ (cm$^{-3}$; left) and radiation field $G_0$ (Habing; right) in the SMC star-forming cloud N22 as determined by `LineRatioFit`. The fit uses a `ModelSet` computed with the metallicity of the SMC ($Z = 0.3$). See Listing A.4.

method `H2ExcitationFit.explore` lets users interactively probe the resultant maps and excitation fits.

Listing A.5 gives an example of fitting the excitation conditions given observations of six $H_2$ rotational lines. The data are of NGC 2023 and taken from Figure 9 of Sheffer et al. (2011), adding an artificial point for the $J = 6$ line to allow for the fitting of the OPR. Figure 8 shows the result of the fit. We find the same temperatures and OPR, within errors, as Sheffer et al. (2011).

For both `LineRatioFit` and `H2ExcitationFit`, the fit results are stored per pixel in an `FitMap` object, which is derived from `astropy.nddata.NDData`. The `FitMap` will contains `lmfit.model.ModelResult` objects for `H2ExcitationFit` or `lmfit.minimizer.MinimizerResult` objects `LineRatioFit`. The user can thus examine in detail the fit at any pixel.

### 5.3. H II Region Diagnostics

Observations of fine-structure line ratios arising in ionized gas can be used to estimate the electron density, $n_e$, and gas temperature, $T_e$, in an H II region. In general, lines that arise from different energies above ground give estimates of the gas temperature, while lines that arise from similar levels above ground but with different collision strengths give estimates of the gas density. We focus on lines that are expected to be bright in JWST observations of Galactic H II regions, namely, those arising from Fe$^+$, Ar$^{+2}$, and Ar$^{+4}$. In particular, Fe$^+$ has great potential for producing diagnostic line ratios due to the large number of levels excited in an H II region, but with caveats as noted here. Low-level Fe$^+$ line emission is also produced in the neutral gas within the PDR, and thus the same species could trace physical conditions continuously from ionized to neutral gas.

We assume that the line emission is in the optically thin limit so that the ratio of intensities is given by the ratio of the volume emissivity. For Ar$^{+2}$ and Ar$^{+4}$ we use CHIANTI (Del Zanna et al. 2021; Dere et al. 1997) using the default values for the A values and collision strengths. For Fe$^+$, we substituted the default values in CHIANTI with A values from Deb & Hibbert (2011) and collision strengths from Smyth et al. (2019). The emissivity ratios are found in the temperature range from $T_e = 10^3$ K to $10^4$ K, and the density ranges from $n_e = 10^2$ cm$^{-3}$ to $10^6$ cm$^{-3}$. FITS files of the resulting values are constructed, and phase-space plots and data overlays can then be made using the tools discussed in Section 5.1. A sample figure is shown in Figure 9. We note, however, that for [Fe II] fine-structure lines the published A values and collision strengths vary between different authors and in some cases do not agree

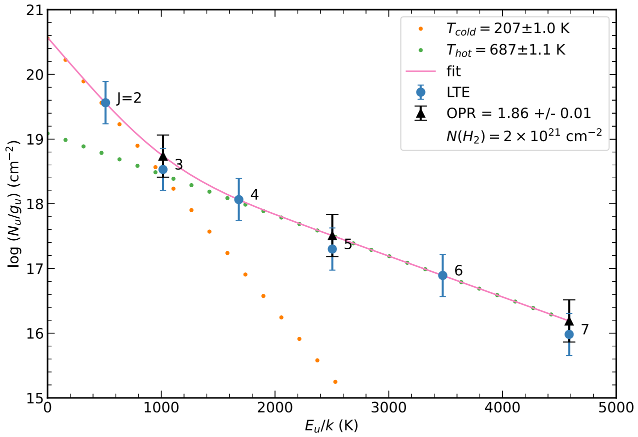


**Figure 8.** Example of PDRT fitting of a molecular hydrogen rovibrational line excitation diagram to determine the temperature and column density for both hot and cold gas components, total column density $N(H_2)$, and OPR. $E_u/k$ is the upper-state energy of the transition, and $N_u/g_u$ is the normalized upper-state column density. The blue dots are data from Sheffer et al. (2011), and the black triangles are the data adjusted for the fitted OPR. See Listing A.5.

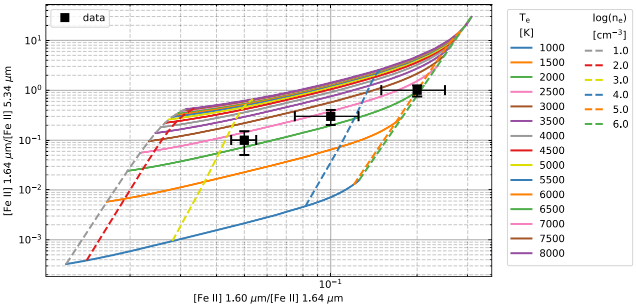


**Figure 9.** Example of PDRT diagnostic plot for electron density, $n_e$, and gas temperature, $T_e$, from the [Fe II] line ratios [Fe II] 1.64 $\mu$m/[Fe II] 5.34 $\mu$m vs. [Fe II] 1.60 $\mu$m/[Fe II] 1.64 $\mu$m. Squares are sample data points with error bars. See Listing A.1.

with the observations (e.g., Koo et al. 2016), so the [Fe II] plots must be considered tentative until the atomic data can be further verified by observations, laboratory work, or quantum calculations.

## 6. Future Development

Below we describe a few future enhancements to the PDR model code, model data, and analysis tools that we intend to undertake. This is not an exhaustive list; we encourage users to submit other requests via GitHub. We also encourage developers to pitch in!

### 6.1. Changes to the Wolfire–Kaufman Model Code

#### 6.1.1. Additional Viewing Angles

The Wolfire–Kaufman PDR models are designed to predict emission line intensities for a face-on geometry in which the line of sight and direction of illumination are parallel. However, there are many PDRs that are observed edge-on in which the line of sight and direction of illumination are perpendicular, and the layers of the PDR are spread across the sky (e.g., Orion Bar). The line intensities are then a function of the position. The peak line intensities and dust continuum can either increase or decrease significantly, compared to face-on, depending on the layer thickness along the line of sight. We will add grids of edge-on models following the prescription given in Pabst et al. (2017), which accounts for optical depth effects in the lines. Using a similar technique we will also provide emitted line intensities for a viewing angle of 45°. Having the three cases, face-on, edge-on, and 45°, will help users better understand the geometries of their sources and how the viewing angle can affect the observed intensities.

#### 6.1.2. Deuterium Chemistry

The lowest HD rotational line at 112 $\mu$m is much easier to excite than $H_2$ due to its 4 times lower energy above ground. When combined with PDR models, HD provides a particularly good measure of the warm molecular gas mass as well as the D/H ratio, which is important for cosmological simulations. It is also a prime target in protostellar disk observations. We will add deuterium chemistry to the PDR model using a simple network (Le Petit et al. 2002) with reaction rates updated, for example, from the Kinetic Database for Astrochemistry[10] and collision rates updated from Wan et al. (2019). The output will be the IR rotational and vibrational line emission as a function of $n$ and $F_{\rm FUV}$.

#### 6.1.3. Improved Treatment of the Metallicities

A large database of PDR observations are available covering a range of metallicities. For example, the Key Insights on Nearby Galaxies: A Far-Infrared Survey with Herschel survey (Kennicutt et al. 2011) mapped galaxies with a metallicity range of 0.04–5 relative to the solar metallicity while SOFIA and Herschel have mapped regions in the LMC (0.5 solar; Chevance et al. 2016; Lebouteiller et al. 2019; Lee et al. 2016, 2019; Okada et al. 2019) and the SMC (0.2 solar; Jameson et al. 2018; Requena-Torres et al. 2012). Although traditionally modelers have used the same scaling for the dust (responsible for extinction and heating) and metal (responsible for gas cooling) abundances, it is now known that these scalings diverge for metallicity $Z < 0.2$ relative to the local ISM (Remy-Ruyer et al. 2014). We will provide a series of models that cover the metallicity range 0.03–5 while accounting for the expected scaling in dust and metals.

### 6.2. Changes to the `pdrtpy` Code

#### 6.2.1. Correction for the [O I] and [C II] Absorption

It has become increasingly apparent that the [O I] 63 $\mu$m, and [C II] 158 $\mu$m lines can be self-absorbed, thereby affecting the interpretation of the integrated line intensities (Bonne et al. 2022; Goldsmith et al. 2021; Graf et al. 2012; Guevara et al. 2020; Kabanovic et al. 2022). This is most often noticed by a [O I] 145 $\mu$m/63 $\mu$m ratio greater than 0.1 or a central dip in velocity-resolved profiles seen in one isotope but not another (e.g., [$^{12}$C II] versus [$^{13}$C II]). Note that PDR models use an escape probability formalism for the line transfer that accounts for the absorption within the PDR but not for cold foreground absorption. It is not known a priori what the correction for absorption should be, and observers usually adopt a correction factor of 2–3x increase in the observed [O I] line intensity so that other line ratios are physically realistic (e.g., Goldsmith et al. 2021; Schneider et al. 2018). We will assist users by providing plots of appropriate correction factors to use for both the [O I] 63 $\mu$m and [C II] 158 $\mu$m lines as functions of the foreground gas temperature and column density and as functions of the line center optical depth. We realize these are not perfect solutions, but they do provide a better understanding of the source environment and direct users to physically motivated solutions.

#### 6.2.2. Regularization for Image-based Fitting

When determining best-fit density and radiation field maps in instances where the observations are few or are of low signal-to-noise ratio, it is possible for a fitting algorithm to oscillate between two nearly equally good solutions (Figure 10). The best way to break this degeneracy would be to obtain additional observations that further constrain the solution, but this is often not practicable, so another method is needed.

When a fitted solution depends discontinuously upon the initial data, as in our example, it is a symptom of an ill-posed (or ill-conditioned) problem. A common method to resolve an ill-posed problem is to employ a regularization technique (Tikhonov & Arsenin 1977). Simply put, regularization means replacing the problem with a different problem whose solution roughly matches the desired solution (has low bias), is less

[10] https://kida.astrochem-tools.org

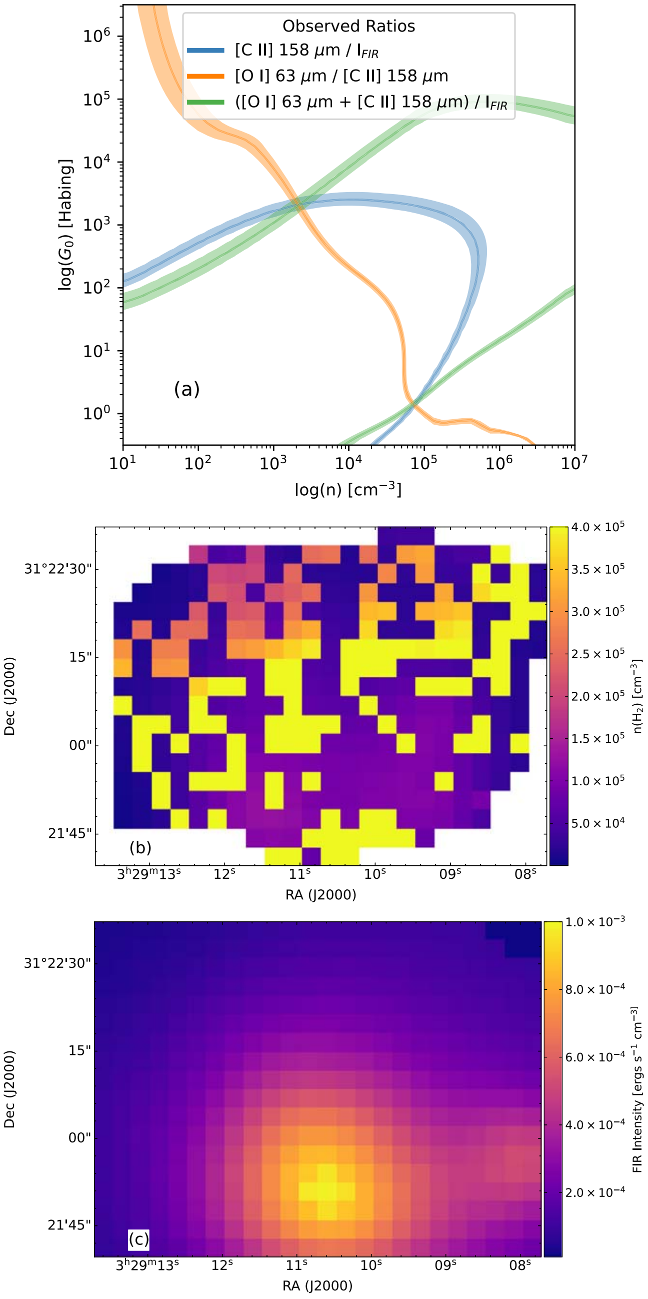


**Figure 10.** Example of when fitted solutions get into trouble. (a) Overlay diagram in the model space ($n$, $G_0$) of the ratios of three common observations: [C II], [O I], and FIR intensity; the width of lines indicate observational uncertainties. Potentially valid solutions appear at crossing points in two very different locations of the model space. (b) Least-squares fitted density map of NGC 1333 from observations. Mirroring the behavior of (a), adjacent pixels can have very different derived densities, despite that the observations (c) are smooth on a larger spatial scale.

sensitive to noise (has low variance), and has a parameter that allows bias-variance tradeoff.

One way to do this is to add a "penalty" to the usual minimization function $M$ that can be used as an additional constraint:

$$M = \underbrace{\sum_i^N \frac{[y_i^{\rm obs} - y_i^{\rm model}(\boldsymbol{v})]^2}{\sigma_i^2}}_{\chi^2} + \underbrace{\lambda [f_i(\boldsymbol{v})]^2}_{\rm penalty}, \qquad (2)$$

where $y_i^{\rm obs}$ is the set of observed data, $y_i^{\rm model}$ is the model calculation, $\boldsymbol{v}$ is the set of variables in the model to be optimized in the fit, $\sigma_i$ is the estimated uncertainty in the data, and $\lambda$ is the bias-variance tuning parameter.

Because observed maps are typically smooth over several pixels (e.g., Figure 10(c)), a reasonable choice would be a penalty function that enforces local smoothness. For instance, an inverse-distance weighted sum of the spatial derivatives of the observed maps could act as a penalty ("the solution cannot be locally less smooth than the observations"). The choice of $\lambda$ is key and can vary with different sets of observations. Another regularization technique is to characterize the spatial structure of the observational and fitted solution maps with a wavelet transform (Allys et al. 2019; Mallat 2012) and to favor solutions that have the wavelet components found in the observations. This method can be computationally quick and has been successful in medical imaging (Guerquin-Kern et al. 2009).

We will explore different regularization methods, test them against real and simulated data, and implement those that perform well (e.g., low-to-modest computational cost, few corner cases, robust to adding or subtracting data). We will provide guidance to users on choosing $\lambda$ as well as explore ways to have PDRT choose it for them, for instance, by examining the Akaike information criterion (Burnham & Anderson 2003), which is already calculated for every pixel by PDRT. The regularization enhancement will also apply to the $H_2$ excitation fitting tool in its map mode.

## 7. Summary

The PDR Toolbox is a mature, versatile package for analysis of photodissociation regions with a wide range of physical conditions. It is applicable to observations from many telescopes from the infrared to the submillimeter regimes. It can also be used to compare models from different PDR codes.

We are grateful for the support from the NASA Astrophysics Data Analysis Program award #80NSSC19K0573, from the SOFIA Legacy Program, FEEDBACK, provided by NASA through award SOF070077 issued by USRA to the University of Maryland, and from JWST-ERS program ID 1288 through a grant from the Space Telescope Science Institute under NASA contract NAS5-03127 to the University of Maryland. We are grateful for helpful conversations with and manuscript comments from the FEEDBACK and PDRs4All teams. Special thanks to Maitraiyee Tiwari for significant beta-testing. We thank Markus Röllig and Volker Ossenkopf-Okada for providing the KOSMA-tau models. Support for the early development of the PDRT CGI version came from NASA Astrophysics Data Program and Applied Information Systems Research Program grants. We thank the anonymous referee for helpful comments.

*Software:* Astropy (Astropy Collaboration et al. 2013, 2018), emcee (Foreman-Mackey et al. 2013), lmfit-py (Newville et al. 2021), matplotlib (Hunter 2007), numpy (van der Walt et al. 2011).

# Appendix
# Code Listings

## *A.1. Models, Modelsets, and ModelPlot*

```
############################################
###           Listing A.1: models, ModelSet, and ModelPlot           ###
############################################
from pdrtpy.modelset import ModelSet
from pdrtpy.plot.modelplot import ModelPlot]
from pdrtpy.measurement import Measurement
import pdrtpy.pdrutils as utils
import astropy.units as u
from astropy.nddata import StdDevUncertainty
# Get the Wolfire-Kaufman 2020 Z = 1 models
ms = ModelSet(’wk2020’,z = 1)
# Get KOSMA-tau R = 3.1 model
mskt = ModelSet(’kt2013wd01-7’,z = 1,mass = 100,medium = ’clumpy’)
# Example of how to fetch a given model, the [OI] 63 micron/[CII] 158 micron intensity ratio.
# The returned model type is pdrtpy.measurement.Measurement.
model = ms.get_model(’OI_63/CII_158’)
modelkt = mskt.get_model(’OI_63/CII_158’)
# Find all the models that use some combination of CO(J = 1-0), [C II] 158 micron,
#  [O I] 145 micron, and far-infrared intensity. This example gets both intensity
# and ratio models, though one can specify model_type = ’intensity’
# or model_type = ’ratio’ to get one or the other.
# The models are returned as a dictionary with the keys set to the model IDs.
mods = ms.get_models([’CII_158’,’OI_145’, ‘CO_10’, ‘FIR’],model_type = ’both’)
print(list(mods.keys()))
# Output of above: [’OI_145’, ’CII_158’, ’CO_10’, ’CII_158/OI_145’, ’CII_158/CO_10’,
#       ’CII_158/FIR’, ’OI_145+CII_158/FIR’]
# Plot a selected model and save it to a PDF file. Note in this example,
# we request Habing units for the FUV field.
# WK
mp = ModelPlot(ms)
mp.plot(’OI_145+CII_158/FIR’,yaxis_unit = ’Habing’,
   label = True, cmap = ’viridis’, colors = ‘k’,norm = ’log’)
mp.savefig(’example1a_figure.pdf’)
# KT
mpkt = ModelPlot(mskt)
mpkt.plot(’OI_145+CII_158/FIR’,yaxis_unit = ’Habing’,
   label = True, cmap = ’viridis’, colors = ‘k’,norm = ’log’)
mpkt.savefig(’example1b_figure.pdf’)
rcw49 = []
label = ["shell","pillar","northern cloud","ridge"]
format_ = ["k+","b+","g+","r+"]
# The data files are in the testdata directory of the pdrtpy installation
for region in ["shell","pil","nc","ridge"]:
   f1 = utils.get_ testdata(f"cii-fir-region.tab")
   f2 = utils.get_ testdata(f"cii-co-region.tab")
   rcw49.append(Measurement.from_table(f1))
   rcw49.append(Measurement.from_table(f2))
mp.phasespace([’CII_158/FIR’,’CII_158/CO_32’],nax1_clip = [1E2,1E5]*u.Unit(’cm-3’),
     nax2_clip = [1E1,1E6]*utils.habing_unit, measurements = rcw49,label = label,
     fmt = format_,title = "RCW 49 Regions")
mp.savefig(’example1c_figure.pdf’)
# Example ionized gas line diagnostic diagram
i1 = Measurement(identifier = ’FEII_1.60/FEII_1.64’,
        data = [0.1,0.05,0.2],
        uncertainty = StdDevUncertainty([0.025,0.005,0.05]),unit="")
i2 = Measurement(identifier = ’FEII_1.64/FEII_5.34’,
        data = [0.3,0.1,1.0],
        uncertainty = StdDevUncertainty([0.1,0.05,0.25]),unit = "")
mp.phasespace([’FEII_1.60/FEII_1.64’,’FEII_1.64/FEII_5.34’],
     nax2_clip = [10,1E6]*u.Unit(’cm-3’),nax1_clip=[1E3,8E3]*u.Unit(’K’),
    measurements = [i1,i2],errorbar = True)
mp.savefig(’example1d_figure.pdf’)
```

### *A.2. Fitting Intensity Ratios for Single-pixel Observations*

```
##############################################
###      Listing A.2: Fitting intensity ratios for single-pixel observations      ###
##############################################
from pdrtpy.measurement import Measurement
from pdrtpy.modelset import ModelSet
from pdrtpy.tool.lineratiofit import LineRatioFit
from pdrtpy.plot.lineratioplot import LineRatioPlot
import pdrtpy.pdrutils as utils
from astropy.nddata import StdDevUncertainty
from lmfit import fit_report
# Example using single-beam observations of [OI] 163 micron, [CI] 609 micron, CO(J = 4-3),
# and [CII] 158 micron lines. You create ‘Measurements‘ for these using the constructor
# which takes the value, error, line identifier string, and units. The value and the error
# must be in the same units. You can mix units in different Measurements; note we use
# K km/s for the CO observation below. The PDR Toolbox will convert all ‘Measurements‘
# to a common unit before using them in a fit. You can also add optional beam size
# (bmaj,bmin,bpa), however the tools requires all ‘Measurements‘ have the same beam size
# before calculations can be performed. (If you don’t provide beam parameters for any of
# your Measurements, the Toolbox will assume they are all the same).
myunit = ’erg s-1 cm-2 sr-1’ # default unit for value and error
m1 = Measurement(data = 3.6E-4,uncertainty = StdDevUncertainty(1.2E-4),
      identifier = ’OI_63’,unit = myunit)
m2 = Measurement(data = 1E-6,uncertainty = StdDevUncertainty([3E-7]),
      identifier = ’CI_609’,unit = myunit)
m3 = Measurement(data = 26,uncertainty = StdDevUncertainty([5]),
      identifier = ’CO_43’,restfreq = ’461.04077 GHz’, unit = ’K km/s’)
m4 = Measurement(data = 8E-5,uncertainty = StdDevUncertainty([8E-6]),
      identifier = ’CII_158’,unit = myunit)
observations = [m1,m2,m3,m4]
ms = ModelSet(’wk2020’,z = 1)
# Instantiate the LineRatioFit tool giving it the ModelSet and Measurements
p = LineRatioFit(ms,measurements = observations)
p.run()
# Print the fitted quantities using Python f-strings and the fit report via lmfit
print(f"n = p.density:.2e\nX = utils.to(’Draine’,p.radiation_field):.2e")
print(fit_report(p.fit_result[0]))
# Create the plotting tool for the LineRatioPlot,
# then make plots of the observed ratios overlayed on the model ratios
plot = LineRatioPlot(p)
plot.ratios_on_models(yaxis_unit = ’Draine’,colorbar = True,norm = ’log’,
cmap = ’cividis’,label = True,ncols = 3,figsize = (23,7))
plot.savefig(’example2_figure.pdf’)
plot.overlay_all_ratios(yaxis_unit = ’Draine’,figsize = (6,7))
plot.savefig(’example3_figure.pdf’)
# Plot the reduced chisquare, with only contours and legend
plot.chisq(image = False,colors = ’k’,label = True,legend = True,yaxis_unit = ’Draine’,
figsize = (6,7))
plot.savefig(’example4_figure.pdf’)
```

### *A.3. Fitting Intensity Ratios for Single-pixel Observations with MCMC*

```
##############################################
###   Listing A.3: Fitting intensity ratios for single-pixel observations with MCMC   ###
##############################################

from pdrtpy.measurement import Measurement
from pdrtpy.modelset import ModelSet
from pdrtpy.tool.lineratiofit import LineRatioFit
from pdrtpy.plot.lineratioplot import LineRatioPlot
import pdrtpy.pdrutils as utils
from astropy.nddata import StdDevUncertainty
from copy import deepcopy
import corner
import numpy as np
myunit = ’erg s-1 cm-2 sr-1’ # default unit for value and error
m1 = Measurement(data = 3.6E-4,uncertainty = StdDevUncertainty(1.2E-4),
    identifier = "OI_63",unit = myunit)
m2 = Measurement(data = 1E-6,uncertainty = StdDevUncertainty([3E-7]),
     identifier = "CI_609",unit = myunit)
m3 = Measurement(data = 26,uncertainty = StdDevUncertainty([5]),
    identifier = "CO_43",restfreq = "461.04077 GHz", unit = "K km/s")
m4 = Measurement(data = 8E-5,uncertainty = StdDevUncertainty([8E-6]),
     identifier = "CII_158",unit = myunit)
observations = [m1,m2,m3,m4]
ms = ModelSet(’wk2020’,z = 1)
# Instantiate the LineRatioFit tool giving it the ModelSet and Measurements
p = LineRatioFit(ms,measurements = observations)
p.run(method = ’emcee’,steps = 2000)
res = p.fit_result[0]
# the value of the Draine unit in cgs
scale = utils.draine_unit.cgs.scale
# copy the results table
rescopy = deepcopy(res.flatchain)
# scale the radiation _ field column of the table to Draine since it is in cgs
 rescopy[’radiation_ field’] / =  scale
# now copy and scale the ‘best fit’’ values where the cross hairs are plotted.
 truths  =  np.array(list(res.params.valuesdict().values()))
truths[1] / = scale
fig = corner.corner(rescopy, bins = 20,range = [(1E4,1.2E5), (10,500)],
      labels = [r’$n~{\rm [cm^{-3}]}$’,r’$\chi~{\rm [Draine]}$’]],
      truths = truths)
fig.savefig(’example5_figure.pdf’,facecolor = ’white’,transparent = False)
```

### *A.4. Fitting Intensity Ratios for Map Observations*

```
####################################
###   Listing A.4: Fitting intensity ratios for map observations   ###
####################################

from pdrtpy.measurement import Measurement
from pdrtpy.modelset import ModelSet
from pdrtpy.tool.lineratiofit import LineRatioFit
from pdrtpy.plot.lineratioplot import LineRatioPlot
import pdrtpy.pdrutils as utils
# Get the input filenames of the FITS files in the testdata directory
# utils.get_testdata() is a special method to locate files there.
# These are maps from Jameson et al. 2018.
print(’Test FITS files are in: %s’%utils.testdata_dir())
 cii_flux = utils.get_testdata(’n22_ cii_flux.fits’) # [C II] flux
 cii_err = utils.get_testdata(’n22_ cii_error.fits’)  # [C II] error
oi_flux = utils.get_testdata(’n22_oi_flux.fits’)  # [O I] flux
```

```
oi_err=utils.get_testdata('n22_oi_error.fits')  # [O I] error
FIR_flux=utils.get_testdata('n22_FIR.fits')  # FIR flux
# Output file names
 cii_combined='n22_ cii_flux_error.fits'
oi_combined='n22_oi_flux_error.fits'
FIR_combined='n22_FIR_flux_error.fits'
# create the Measurements and write them out as FITS files with two HDUs.
Measurement.make_ measurement(cii_flux, cii_err,
      outfile= cii_combined, overwrite=True)
Measurement.make_measurement(oi_flux, oi_err,
      outfile=oi_combined, overwrite=True)
# Assign a 10% error in FIR flux
Measurement.make_measurement(FIR_flux, error='10%',
      outfile=FIR_combined, overwrite=True)
# Read back in the FITS files to Measurements
 cii_meas= Measurement.read(cii_combined, identifier="CII_158")
FIR_meas=Measurement.read(FIR_combined, identifier="FIR")
oi_meas=Measurement.read(oi_combined, identifier="OI_63")
# Here we will use the Small Magellanic Cloud ModelSet that have Z =0.1
# These are a limited set of models with just a few lines covered.
smc_ms=ModelSet('smc',z=0.1)
p=LineRatioFit(modelset=smc_ms, measurements= [cii_meas,FIR_meas,oi_meas])
p.run()
plot=LineRatioPlot(p)
plot.density(contours= True,norm="log",cmap='cividis')
plot.savefig('example6_n_figure.pdf')
plot.radiation_field(units='Habing',contours= True,norm='simple',cmap='cividis')
plot.savefig('example6_g0_figure.pdf')
# Save the results to FITS files.
p.density.write('N22_density_map.fits',overwrite=True)
p.radiation_field.write('N22_G0_map.fits',overwrite=True)
```

### *A.5. Creating and Fitting H2 Excitation Diagrams, Including Ortho-to-para Ratio (OPR)*

```
#########################################################
###   Listing A.5: Creating and fitting H2 excitation diagrams including ortho-to-para ratio (OPR)   ###
#########################################################
from pdrtpy.measurement import Measurement
from pdrtpy.tool.h2excitation import H2ExcitationFit
from pdrtpy.plot.excitationplot import ExcitationPlot
from astropy.nddata import StdDevUncertainty
intensity=dict()
intensity['H200S0']=3.003e-05
intensity['H200S1']=3.143e-04
intensity['H200S2']=3.706e-04
intensity['H200S3']=1.060e-03
intensity['H200S4']=5.282e-04
intensity['H200S5']=5.795e-04
observations=[]
for i in intensity:
 m=Measurement(data=intensity[i],
 uncertainty=StdDevUncertainty(0.75*intensity[i]),
 identifier=i,unit="erg cm-2 s-1 sr-1")
 observations.append(m)
# Create the tool to run the fit
hopr=H2ExcitationFit(observations)
# Instantiate the plotter
hplot=ExcitationPlot(hopr,"$H_2$")
# Set some plot parameters appropriate for manuscript figure;
# these pass through to matplotlib
```

```
hplot._plt.rcParams["xtick.major.size"] = 7
hplot._plt.rcParams["xtick.minor.size"] = 4
hplot._plt.rcParams["ytick.major.size"] = 7
hplot._plt.rcParams["ytick.minor.size"] = 4
hplot._plt.rcParams[’font.size’] = 14
hplot._plt.rcParams[’axes.linewidth] = 1.5
hplot.ex_diagram(ymax = 21)
hplot.savefig(’example9_figure.png’,dpi = 300)
# Fit a two temperature model allowing OPR to vary
hopr.run(fit_opr = True)
hplot.ex_diagram(show_fit = True,ymax = 21)
hplot.savefig(’example10_figure.png’,dpi = 300)
```

## ORCID iDs

Marc W. Pound https://orcid.org/0000-0002-7269-342X
Mark G. Wolfire https://orcid.org/0000-0003-0030-9510

## References

Allys, E., Levrier, F., Zhang, S., et al. 2019, A&A, 629, A115
Astropy Collaboration, Price-Whelan, A. M., Sipocz, B. M., et al. 2018, AJ, 156, 123
Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, A&A, 558, A33
Bonne, L., Schneider, N., García, P., et al. 2022, ApJ, 935, 171
Burnham, K., & Anderson, D. 2003, Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach (New York: Springer)
Burton, M. G., Hollenbach, D. J., & Tielens, A. G. G. 1992, ApJ, 399, 563
Cazaux, S., & Tielens, A. G. G. M. 2004, ApJ, 604, 222
Cazaux, S., & Tielens, A. G. G. M. 2010, ApJ, 715, 698
Chevance, M., Madden, S. C., Lebouteiller, V., et al. 2016, A&A, 590, A36
Deb, N. C., & Hibbert, A. 2011, A&A, 536, A74
Del Zanna, G., Dere, K. P., Young, P. R., & Landi, E. 2021, ApJ, 909, 38
Dere, K. P., Landi, E., Mason, H. E., Monsignori Fossi, B. C., & Young, P. R. 1997, A&AS, 125, 149
Draine, B. T. 1978, ApJS, 36, 595
Foreman-Mackey, D. 2016, JOSS, 1, 24
Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. 2013, PASP, 125, 306
Fuente, A., Martin-Pintado, J., Rodriguez-Fernandez, N. J., et al. 1999, ApJL, 518, L45
Goldsmith, P., Langer, W. D., Seo, Y., et al. 2021, ApJ, 916, 6
Graf, U. U., Simon, R., Stutzki, J., et al. 2012, A&A, 542, L16
Guerquin-Kern, M., Van De Ville, D., Vonesch, C., & Baritaux, J.-C. 2009, in IEEE Int. Symp. on Biomedical Imaging: From Nano to Macro, ed. Stanley Reeves (Piscataway, NJ : IEEE), 193
Guevara, C., Stutzki, J., Ossenkopf-Okada, V., et al. 2020, A&A, 636, A16
Habing, H. J. 1968, Bull. Astron. Inst. Netherlands, 19, 421
Heays, A. N., Bosman, A. D., & van Dishoeck, E. F. 2017, A&A, 602, A105
Herbst, T. M., Beckwith, S. V. W., Glindemann, A., et al. 1996, AJ, 111, 2403
Herrera-Camus, R., Bolatto, A. D., Wolfire, M. G., et al. 2015, ApJ, 800, 1
Hollenbach, D., Kaufman, M. J., Neufeld, D., Wolfire, M., & Goicoechea, J. R. 2012, ApJ, 754, 105
Hunter, J. D. 2007, CSE, 9, 90
Jameson, K. E., Bolatto, A. D., Wolfire, M., et al. 2018, ApJ, 853, 111
Kabanovic, S., Schneider, N., Ossenkopf-Okada, V., et al. 2022, A&A, 659, A36
Kaufman, M. J., Wolfire, M., & Hollenbach, D. J. 2006, ApJ, 644, 283
Kaufman, M. J., Wolfire, M. G., Hollenbach, D. J., & Luhman, M. L. 1999, ApJ, 527, 795
Kennicutt, R. C., Calzetti, D., Aniano, G., et al. 2011, PASP, 123, 1347
Koo, B.-C., Raymond, J. C., & Kim, H.-J. 2016, JKAS, 49, 109
Larson, R. B. 1981, MNRAS, 194, 809
Le Petit, F., Roueff, E., & Le Bourlot, J. 2002, A&A, 390, 369
Lebouteiller, V., Cormier, D., Madden, S. C., et al. 2019, A&A, 632, A106
Lee, M.-Y., Madden, S. C., Lebouteiller, V., et al. 2016, A&A, 596, A85
Lee, M.-Y., Madden, S. C., Le Petit, F., et al. 2019, A&A, 628, A113
Lique, F., Kłos, J., Alexander, M. H., Le Picard, S. D., & Dagdigian, P. J. 2018, MNRAS, 474, 2313
Mallat, S. 2012, Communications on Pure and Applied Mathematics, 65, 1331
Mathis, J. S., Mezger, P. G., & Panagia, N. 1983, A&A, 128, 212
Neufeld, D. A., Melnick, G. J., & Harwit, M. 1998, ApJ, 506, L75
Neufeld, D. A., & Wolfire, M. G. 2016, ApJ, 826, 183
Newville, M., Otten, R., Nelson, A., et al. 2021, Lmfit: Non-Linear Least-Squares Minimization and Curve-Fitting for Python, 1.0.3, Zenodo, doi:10.5281/zenodo.5570790
Okada, Y., Gusten, R., Requena-Torres, M. A., et al. 2019, A&A, 621, A62
Pabst, C. H. M., Goicoechea, J. R., Teyssier, D., et al. 2017, A&A, 606, A29
Pound, M. W., & Wolfire, M. G. 2008, in ASP Conf. Ser. 394, Astronomical Data Analysis Software and Systems XVII, ed. R. W. Argyle, P. S. Bunclark, & J. R. Lewis (San Francisco, CA: ASP), 654
Remy-Ruyer, A., Madden, S. C., Galliano, F., et al. 2014, A&A, 563, A31
Requena-Torres, M. A., Gusten, R., Weiß, A., et al. 2012, A&A, 542, L21
Röllig, M., & Ossenkopf-Okada, V. 2022, A&A, 664, A67
Rollig, M., Szczerba, R., Ossenkopf, V., & Gluck, C. 2013, A&A, 549, A85
Roussel, H., Helou, G., Hollenbach, D. J., et al. 2007, ApJ, 669, 959
Schneider, N., Rollig, M., Simon, R., et al. 2018, A&A, 617, A45
Schneider, N., Simon, R., Guevara, C., et al. 2020, PASP, 132, 104301
Schoier, F. L., van der Tak, F. F. S., van Dishoeck, E. F., & Black, J. H. 2005, A&A, 432, 369
Sheffer, Y., Wolfire, M. G., Hollenbach, D. J., Kaufman, M. J., & Cordier, M. 2011, ApJ, 741, 45
Smyth, R. T., Ramsbottom, C. A., Keenan, F. P., Ferland, G. J., & Ballance, C. P. 2019, MNRAS, 483, 654
Sternberg, A., & Neufeld, D. A. 1999, ApJ, 516, 371
Tielens, A. G. G. M., & Hollenbach, D. 1985, ApJ, 291, 722
Tikhonov, A. N., & Arsenin, V. Y. 1977, Solutions of Ill-posed problems (Washington, DC: V. H. Winston & Sons)
Tiwari, M. 2022, in Our Galactic Ecosystem: Opportunities and Diagnostics in the Infrared and Beyond (UCLA Lake Arrowhead Lodge, CA: SOFIA)
Tiwari, M., Karim, R., Pound, M. W., et al. 2021, ApJ, 914, 117
Tiwari, M., Wolfire, M., Pound, M. W., et al. 2022, AJ, 164, 150
van der Walt, S., Colbert, S. C., & Varoquaux, G. 2011, CSE, 13, 22
Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, NatMe, 17, 261
Visser, R., Kristensen, L. E., Bruderer, S., et al. 2012, A&A, 537, A55
Wan, Y., Balakrishnan, N., Yang, B. H., Forrey, R. C., & Stancil, P. C. 2019, MNRAS, 488, 381
Weingartner, J. C., & Draine, B. T. 2001a, ApJS, 134, 263
Weingartner, J. C., & Draine, B. T. 2001b, ApJ, 548, 296
Wolfire, M. G., Hollenbach, D., & McKee, C. F. 2010, ApJ, 716, 1191
Wolfire, M. G., McKee, C. F., Hollenbach, D., & Tielens, A. G. G. M. 2003, ApJ, 587, 278
Wolfire, M. G., Tielens, A. G. G. M., & Hollenbach, D. 1990, ApJ, 358, 116
Wolfire, M. G., Vallini, L., & Chevance, M. 2022, ARA&A., 60, 247