
PDIpy

Release 1

Andrew Philip Freiburger

Dec 28, 2022

CONTENTS

1	Installation	3
2	Contents	5
2.1	Usage	5
2.2	PDipy API	6
2.3	PDipy parameter files	11

Antibiotic resistance is a developing medical crisis that is projected to surpass cancer in annual deaths by mid-21st century. Photodynamic Inactivation (PDI) escapes resistance evolution and may therefore be an essential antibiotic method to hamper the growing threat of resistant pathogens. The requisite rate of research to mitigate these somber projections requires computational tools that can improve and expedite experimental research in developing PDI treatments.

PDIpy is the first comprehensive software of PDI that simulates PDI biochemistry from a chemical kinetics model. PDIpy accepts user inputs of the simulated system, constructs and executes a **Tellurium** kinetic system, and processes and exports the results in spreadsheets and SVG figures. Investigation of the simulation data is further supported with a PDIpy function that parses the data based upon either a specified inactivation or time. The **examples directory** of the PDIpy GitHub exemplifies PDIpy through replicating experimental observations and conducting multiple sensitivity analyses. Users and developers are encouraged to critique, improve, and join the open-source project through **GitHub issues** or emailing afreiburger@uvic.ca, respectively.

Note: This project is under active development, and may therefore be subject to incompatible changes in the API. We seek to minimize these types of upgrades.

INSTALLATION

pdipy is installed in a command prompt, Powershell, Terminal, or Anaconda Command Prompt via pip:

```
pip install pdipy
```


CONTENTS

2.1 Usage

```
from pdipy import PDI

# define the simulation conditions
pdi = PDI(
    solution_dimensions = {
        'area_sqr_cm': pi/4,
        'depth_cm': 46,
    },
    printing = False
)
pdi.define_conditions(
    bacterial_specie = 'S_aureus',
    bacterial_cfu_ml = 3e7,
    photosensitizer = 'protoporphyrin IX',
    photosensitizer_characteristics = {
        'formula': {
            'value': 'C34_H34_N4_O4'
        },
        'dimensions': {
            'length (A)': length,
            'width (A)': width,
            'depth (A)': 4,
            'shape': 'rect',
        }
    },
    photosensitizer_molar = 18e-9,
    measurement = {'irradiance': 8},
)

# execute and export the simulation
pdi.simulate(
    export_name = 'Bozja_et_al, 1E4 Lux',
    figure_title = '1E4 Lux, Protoporphyrin IX',
    display_ps_excitation = True,
)

# parse the data and evaluate the PDI object contents
```

(continues on next page)

(continued from previous page)

```
value, unit = pdi.parse_data(log_reduction = 5)
print(dir(pdi))
```

2.1.1 Accessible content

Numerous entities are stored within the PDI object, and can be subsequently used in a workflow. The complete list of content within the PDI object can be identified and printed through the Python function `dir()`, while the following list highlights a few notable contents within the PDI object:

- *raw_data* & *processed_data* `Pandas.DataFrame`: [Pandas DataFrames](#) that contain the raw and processed simulation data, respectively. These files are also exported through the export function.
- *model* & *phrasedml_str* `str`: The kinetic model and its corresponding [SED-ML](#) plot, respectively, composed in a string that is read by Tellurium.
- *bacterium*, *photosensitizer*, & *light* `dict`: Simulation parameters for the bacterium, photosensitizer, and light, respectively, in a for that is read by the code.
- *parameters*, *variables*, & *results* `dict`: Input parameters, calculation variables, and simulation results, respectively, that are subsequently processed into DataFrames and exported as CSVs.
- *figure* & *ax* `Matplotlib.Pyplot.subplots`: The [Matplotlib objects](#) of the simulation figure, which allows the user to externally manipulate the figure without recreating a new figure from the raw or processed data.
- *chem_mw* `chemw.ChemMW`: The `ChemMW` object from the [ChemW module](#), which allows users to calculate the molecular weight from a string of any chemical formula or common name.
- *bacteria* `list`: A list of all the predefined bacteria parameter files, from which a user can easily simulate via the PDI object.
- *light_parameters*, *photosensitizers*, & *solution* `dict`: Dictionaries of the predefined options and parameters for the light sources, photosensitizers, and solution dimensions, respectively.

2.2 PDipy API

2.2.1 PDI()

The simulation environment is defined:

```
import pdipy
pdi = pdipy.PDI(solution_dimensions = {}, surface_system = False, well_count = 24,
               verbose = False, jupyter = False, printing = True)
```

- *solution_dimensions* `dict`: defines the physical dimensions of the simulated solution, which are used to calculate photonic density and photosensitizer volume proportion.
- *surface_system* `bool`: specifies whether a photodynamic system with a surface-bound, cross-linked, photosensitizer will be simulated.
- *well_count* `int`: specifies the petri dish well count that will be simulated, which begets default dimensions of the simulated solution.
- *verbose* & *printing* `bool`: specifies whether simulation details and simulation results, respectively, will be printed. These are valuable for troubleshooting.

- `jupyter` bool: specifies whether the simulation is being conducted in a Jupyter Notebook, which allows `display()` to illustrate data tables and figures.

define_conditions()

The characteristics of the simulated system are concisely defined in a single function:

```
pdi.define_conditions(bacterial_specie = 'S_aureus', bacterial_characteristics = {},
    bacterial_cfu_ml = 1e6, biofilm = False, photosensitizer = 'A3B_4Zn',
    photosensitizer_characteristics = {}, absorbance_nm = {}, transmittance = {},
    photosensitizer_molar = None, surface_system = False, photosensitizer_g = 90e-9,
    cross_linked_sqr_m = 0.0191134, area_coverage = False,
    light_source = None, light_characteristics = {}, measurement = {})
```

- `bacterial_specie` str: specifies one of the bacteria in the `pdi/parameters/bacteria` directory, where *S. aureus* is imported by default. These imported parameters are overwritten by the `bacterial_characteristics`, and can be used complementarily. The PDIPy parameter files directory of these docs detail the contents and syntax of these files.
- `bacterial_characteristics` dict: provides custom characteristics of the simulated bacterium that supplant imported characteristics from the `bacterial_specie` argument. The expected dictionary keys `shape`, `membrane_thickness_nm`, `cell_mass_pg`, `cell_volume_fL`, `eps_oxidation_rate_constant`, `cellular_dry_mass_proportion_biofilm`, `doubling_rate_constant`, and `biofilm_oxidation_fraction_lysis` are all themselves dictionaries that follow a simple structure:

```
{
    "value": 0.268,
    "reference": "A. G. O'DONNELL, M. R. NAHAIE, M. GOODFELLOW, D. E. MINNIKINI,
    ↪and V. HAJEK. Numerical Analysis of Fatty Acid Profiles in the Identification of
    ↪Staphylococci. Journal of General Microbiology (1989). 131, 2023-2033. https://doi.org/
    ↪10.1099/00221287-131-8-2023",
    "notes": "All saturated SCFAs were summed from Table 2 for all S. aureus
    ↪entries."
}
```

The `reference` and `notes` keys are optional, yet may be important for provenance and reproducibility of the simulation results. The final key of `bacterial_characteristics` is `membrane_chemicals`, which characterizes the chemicals that constitutes the bacterial membrane such as BCFA for branch-chain, and SCFA for straight-chain, saturated fatty acids. The sub-structure of these values are exemplified by the following content for the "SCFA" entry for *S. aureus*:

```
{
    "density_gL": {
        "value": 0.94,
        "reference": ["https://pubchem.ncbi.nlm.nih.gov/compound/Stearic-acid
    ↪#section=Density"],
        "notes": "The density for all saturated fatty acids is estimated as stearic
    ↪acid."
    },
    "formula": [
        "C20_H38_O2",
        "C18_H34_O2",
        "C16_H30_O2"
    ],
}
```

(continues on next page)

(continued from previous page)

```

    "proportion": {
        "value": 0.268,
        "reference": "A. G. O'DONNELL, M. R. NAHAIE, M. GOODFELLOW, D. E. MINNIKINI,
↪ and V. HAJEK. Numerical Analysis of Fatty Acid Profiles in the Identification of
↪ Staphylococci. Journal of General Microbiology (1989). 131, 2023-2033. https://doi.org/
↪ 10.1099/00221287-131-8-2023",
        "notes": "All saturated SCFAs were summed from Table 2 for all S. aureus.
↪ entries."
    }
}

```

- *bacterial_cfu_ml* float: specifies the bacterial concentration for simulations of solution-based photosensitizers.
- *biofilm* bool: specifies whether a biofilm will be simulated.
- *photosensitizer* str: names the simulated photosensitizer, where predefined names will load the stored parameter values from `pdipy/parameters/photosensitizers.json`. The PDIPy `parameter` files directory of these docs detail the contents and syntax of these files.
- *photosensitizer_characteristics* dict: defines characteristics of the simulation photosensitizer, which supplant the characteristics from the `photosensitizer` parameter. The expected structure of the dictionary are keys with dictionary substructure according to the following example:

```

{
    "e_quantum_yield": {
        "value": 0.6,
        "reference": "Singlet Oxygen Yields and Radical Contributions in
↪ the Dye-Sensitised Photo-oxidation in methanol of esters of polyunsaturated fatty
↪ acids _oleic, linoleic, linolenic, and arachidonic) Chacon et al., 1988"
    },
    "so_specificity": {
        "value": 0.8,
        "reference": null
    },
    "formula": {
        "value": "C76_H48_N16_F12_Zn",
        "reference": null
    },
    "excitation_nm": {
        "value": [
            [400, 430],
            [530, 625]
        ],
        "reference": null
    },
    "ps_decay (ns)": {
        "value": 1.5,
        "reference": "Akimoto et al., 1999, 'Ultrafast ... Porphyrins'"
    },
    "ps_rise (fs)": {
        "value": 50,
        "reference": "Andersson et al., 1999, 'Photoinduced ... State' ;
↪ Gurzadyan et al., 1998, 'Time-resolved ... Zn-tetraphenylporphyrin'"
    },
}

```

(continues on next page)

(continued from previous page)

```

    "ps_charge_transfer (ns)": {
        "value": 100,
        "reference": "Kupper et al., 2002, 'Kinetics ... Oxygen'"
    },
    "photobleaching_constant (cm^2/J)": {
        "value": 1.74e-7,
        "reference": "\"Photobleaching kinetics, photoproduct formation,
↪and dose estimation during ALA induced PpIX PDT of MLL cells under well oxygenated and
↪hypoxic conditions\" by Dysart et al., 2005",
        "notes": "The 0.015 value from literature is divided by 8.64e4 --
↪the quantity of seconds in a day -- to yield a sensible value. A similar value is
↪discovered from \"PHOTBLEACHING OF PORPHYRINS USED IN PHOTODYNAMIC THERAPY AND
↪IMPLICATIONS FOR THERAPY\" by Mang et al., 1987"
    },
    "dimensions": {
        "shape": "disc",
        "length_A": 32.8,
        "width_A": 32.8,
        "depth_A": 1.5,
        "notes": "The depth is atomic thickness, as quantified by this
↪paper https://www.nature.com/articles/ncomms1291."
    }
}

```

The value sub-key in the dictionary substructures, where it is present, is the only necessary sub-key for each parameter.

- *absorbance_nm* & *transmittance* dict: define the absorbance or transmittance, respectively, of the simulated PS at the simulated concentrations or area coverage. The absence of these parameters triggers the estimation of the photon absorption from first principles and several assumptions.
- *photosensitizer_molar* float: specifies the photosensitizer molar concentration for simulations of a solution-based photosensitizer.
- *surface_system* bool: signals whether the photosensitizer is surface-bound upon a material substratum.
- *photosensitizer_g* float: specifies the mass of photosensitizer that is surface-bound in cross-linked simulations.
- *cross_linked_sqr_m* float: defines the square-meters area that is coated with the bound photosensitizer. This must be provided in tandem with the *photosensitizer_g* parameter.
- *area_coverage* float: specifies the fraction of a substratum area that is covered with the surface-bound photosensitizer.
- *light_source* str: names the simulated light source, where predefined names will load the stored parameter values from `pdipy/parameters/light_source.json`. The PDIPy parameter files directory of these docs detail the contents and syntax of these files.
- *light_characteristics* dict: specifies custom characteristics of the light source, which overwrite characteristics that are specified from the *light_source* option. The expected structure of the dictionary are keys with dictionary substructure according to the following example:

```

{
    "visible_proportion": {
        "value": 0.1,
        "reference": "Macisaac et al., 1999"
    },
}

```

(continues on next page)

(continued from previous page)

```
"lumens_per_watt": {
    "value": 3,
    "reference": "Michael F. Hordeski. Dictionary Of Energy Efficiency Technologies.↵
↵Fairmont Press. ISBN: 9780824748104"
}
```

where the value sub-key in the dictionary substructures is the only necessary sub-key for each parameter.

- *measurement* dict: provides the unit and quantity of the photonic intensity measurement of the light source in a key-value pair. The supported unit options are: irradiance in mW/cm², exposure in J/cm², lux in lumen/m², and lumens in lumens.

simulate()

The aforementioned system specifications are refined into chemical parameters and are executed in a Tellurium kinetic model:

```
pdi.simulate(export_name = None, export_directory = None, figure_title = None,
             y_label = 'log10', exposure_axis = False, total_time = 720, timestep = 3
             experimental_data = {'x':[], 'y':[]}, display_fa_oxidation = False,
             display_ps_excitation = False, display_inactivation = True, export_content =↵
↵True)
```

- *export_name* & *export_directory* str: specify the name and directory, respectively, to which the simulation contents will be saved, where None defaults to a folder name with simulation parameters **PDlpy-*<photosensitizer_selection>-<bacterial_specie>-<count>*** within the current working directory.
- *figure_title* & *y_label* str: specify the title and y-axis label of the simulation figure, respectively. The y-axis label is general to support multiple overlaid plots in the same figure. The value of None defaults to a figure title of **Cytoplasmic oxidation and inactivation of < bacterial_specie >**.
- *exposure_axis* bool: specifies whether the x-axis of the simulation figure will be defined with cumulative exposure J/cm² over the simulation or in minutes of simulation time, where the latter is default.
- *total_time* float: specifies the total simulated time in minutes. This parameter is over-ridden when the predicted oxidation proportion reaches a singularity, at which point the simulation is determined to conclude.
- *timestep* int: specifies the timestep value in minutes of the simulation.
- *display_fa_oxidation*, *display_ps_excitation*, & *display_inactivation* bool: determine whether the fatty acid oxidation, the photosensitizer excitation, or the inactivation proportions, respectively, will be plotted in the figure.
- *export_content* bool: specifies whether the simulation content will be exported.

parse_data()

The inactivation predictions can be easily investigated through this function:

```
value, unit = pdi.data_parsing(log_reduction = None, target_hours = None)
```

- *log_reduction* float: inquires at what time the specified log-reduction is achieved
- *target_hours* float: inquires what log-reduction is achieved that the specified time

Returns *value* float: The value of the search inquiry, reported in the respective units.

Returns *unit* str: The units of the search inquiry result, being either log-reduction or hours.

2.3 PDlpy parameter files

Parameters may be more succinctly provided through comprehensive JSON files, which are automatically imported by the code, than through function arguments; although, these parameter sources are complementary. Each JSON file pertains to a distinct category of simulation parameters, which are individually detailed in the following sections.

2.3.1 photosensitizers

The chemical attributes of the simulated photosensitizer are articulated in the following format:

```
{
  "A3B_4Zn": {
    "e_quantum_yield": {
      "value": 0.6,
      "reference": "Singlet Oxygen Yields and Radical Contributions in
↳the Dye-Sensitised Photo-oxidation in methanol of esters of polyunsaturated fatty
↳acids _oleic, linoleic, linolenic, and arachidonic) Chacon et al., 1988"
    },
    "so_specificity": {
      "value": 0.8,
      "reference": null
    },
    "so_quantum_yield":{
      "value": 0.48,
      "reference": null
    },
    "formula": {
      "value": "C76_H48_N16_F12_Zn",
      "reference": null
    },
    "excitation_nm": {
      "value": [
        [400, 430],
        [530, 625]
      ],
      "reference": null
    },
    "ps_rise (fs)": {
      "value": 50,
      "reference": "Andersson et al., 1999, 'Photoinduced ... State' ;
↳Gurzadyan et al., 1998, 'Time-resolved ... Zn-tetraphenylporphyrin'"
    },
    "ps_decay (ns)": {
      "value": 1.5,
      "reference": "Akimoto et al., 1999, 'Ultrafast ... Porphyrins'"
    },
    "ps_charge_transfer (ns)": {
```

(continues on next page)

(continued from previous page)

```

        "value": 100,
        "reference": "Kupper et al., 2002, 'Kinetics ... Oxygen'"
    },
    "photobleaching_constant (cm2/(J*M))": {
        "value": 600,
        "reference": "Dysart et al., 2005, 'Calculation of Singlet Oxygen_
↪Dose ... and Photobleaching During mTHPC Photodynamic Therapy of MLL Cells'"
    },
    "dimensions": {
        "shape": "disc",
        "length_A": 32.8,
        "width_A": 32.8,
        "depth_A": 1.5,
        "notes": "The depth is atomic thickness, as quantified by this_
↪paper https://www.nature.com/articles/ncomms1291."
    }
}

```

- *e_quantum_yield*, *so_specificity*, & *so_quantum_yield* float: quantum yields of the photosensitizer (PS). The *e_quantum_yield* and *so_specificity* quantum yields correspond to the excitation of a PS after collision with a photon and the generation of singlet oxygen (SO) by that excited PS, respectively. The *so_quantum_yield* quantum yield is the collective probability, and thus the product, of the two aforementioned quantum yields, and is disaggregated into estimates of the first two quantum yields by simply taking its square-root.
- *formula* str: defines the chemical formula of the PS, where underscores may optionally separate elements to improve readability.
- *excitation_nm* 2D-array: specifies the excitation range(s) of the photosensitizer in nanometers.
- *ps_rise (fs)* & *ps_decay (ns)* float: specifies the excitation and fluorescence times in units of femtoseconds and nanoseconds, respectively, for the PS.
- *ps_charge_transfer (ns)* float: specifies the average time after a photon collision when the excited PS relays excitation energy to another molecule, ideally triplex oxygen.
- *photobleaching_constant (cm2/(J*M))* float: specifies the rate constant of the oxygen-dependent photobleaching reaction for the PS.
- *dimensions* dict: specifies the physical dimensions of the PS molecule in angstroms and the approximate shape of the simulated PS, which are used in the absence of absorption data for the PS at the simulated concentration to calculate the molecular volume and then estimate the absorption properties of the chemical in the system.

The reference and notes keys are optional, and other keys of metadata may be added as well at the user's discretion.

2.3.2 wells

The dimensions of the solution in which the PDI simulation is conducted may be described through the `wells.json` file, in addition to the function argument. The default entries of this file reflect standard dimensions of individual wells in the 6-, 12-, 24-, 48-, and 96-well bioassay plates; however, any solution can be defined that provides the `area_sqr_cm`, `depth_cm`, and `extinction_coefficient` (1/m) parameters.

```

{
    "12": {
        "area_sqr_cm": 3.85,

```

(continues on next page)

(continued from previous page)

```

        "depth_cm": 1.766,
        "extinction_coefficient (1/m)": 0.013,
        "dimensions_reference": "https://ca.vwr.com/assetsvc/asset/en_CA/id/
↪25423331/contents/vwr-essential-products-for-tissue-culture.pdf",
        "coefficient_reference": "The effect of photocarrier generating light on
↪light scattering in the Sea. Lorenzen, 1972"
    }
}

```

Other key:value pairs may be defined to specify references or other notes about the system. A sample entry in the wells.json file is provided below:

- *area_sqr_cm* float: specifies the area on the bottom of the simulated solution, in units of cm².
- *depth_cm* float: specifies the depth (height) of the simulated solution, in units of cm.
- *extinction_coefficient (1/m)* float: specifies the rate constant for the scattering of light through the solution, as a function of depth, via the light attenuation equation: remaining fraction = $e^{(-k \cdot z)}$.

2.3.3 bacteria

This folder contains a different JSON file for each bacterial specie. The only bacterium that is specified by default is *Staphylococcus aureus* (S_aureus.json), however, other organisms can be defined by replicating the structure of the default parameter file. The values of each sub-dictionary are stored in the value key.

```

{
  "membrane_chemicals": {
    "BC_SFA": {
      "density_gL": {
        "value": 0.9,
        "reference": ["https://pubchem.ncbi.nlm.nih.gov/compound/Stearic-acid
↪#section=Density", "https://pubchem.ncbi.nlm.nih.gov/compound/445639"],
        "notes": "The density is estimated to be between stearic acid and oleic acid"
      },
      "formula": ["C18_H34_O2", "C16_H30_O2"],
      "proportion": {
        "value": 0.662,
        "reference": "A. G . O'DONNELL, M. R . NAHAIE, M. GOODFELLOW, D. E. MINNIKINI,
↪and V . HAJEK. Numerical Analysis of Fatty Acid Profiles in the Identification of
↪Staphylococci. Journal of General Microbiology (1989). 131, 2023-2033. https://doi.org/
↪10.1099/00221287-131-8-2023",
        "notes": "All BCFAs were summed from Table 2 for all S. aureus entries."
      }
    },
    "SC_SFA": {
      "density_gL": {
        "value": 0.94,
        "reference": ["https://pubchem.ncbi.nlm.nih.gov/compound/Stearic-acid
↪#section=Density"],
        "notes": "The density for all saturated fatty acids is estimated as stearic
↪acid."
      },
      "formula": ["C20_H38_O2", "C18_H34_O2", "C16_H30_O2"],

```

(continues on next page)

(continued from previous page)

```

    "proportion": {
      "value": 0.268,
      "reference": "A. G. O'DONNELL, M. R. NAHAIE, M. GOODFELLOW, D. E. MINNIKINI,
↪and V. HAJEK. Numerical Analysis of Fatty Acid Profiles in the Identification of
↪Staphylococci. Journal of General Microbiology (1989). 131, 2023-2033. https://doi.org/
↪10.1099/00221287-131-8-2023",
      "notes": "All saturated SCFAs were summed from Table 2 for all S. aureus
↪entries."
    },
    "SC_UFA": {
      "density_gL": {
        "value": 0.94,
        "reference": ["https://pubchem.ncbi.nlm.nih.gov/compound/Stearic-acid
↪#section=Density"],
        "notes": "The density for all saturated fatty acids is estimated as stearic
↪acid."
      },
      "formula": ["C20_H38_O2", "C18_H34_O2", "C16_H30_O2"],
      "proportion": {
        "value": 0.07,
        "reference": "A. G. O'DONNELL, M. R. NAHAIE, M. GOODFELLOW, D. E. MINNIKINI,
↪and V. HAJEK. Numerical Analysis of Fatty Acid Profiles in the Identification of
↪Staphylococci. Journal of General Microbiology (1989). 131, 2023-2033. https://doi.org/
↪10.1099/00221287-131-8-2023",
        "notes": "All saturated SCFAs were summed from Table 2 for all S. aureus
↪entries."
      }
    },
    "membrane_thickness_nm": {
      "value": 4,
      "reference": "W.Rawicz, K.C.Olbrich, T.McIntosh, D.Needham, E.Evans (2000). Effect
↪of Chain Length and Unsaturation on Elasticity of Lipid Bilayers. Biophysical Journal,
↪79(1), 328-339. https://doi.org/10.1016/S0006-3495(00)76295-3 ; "The electrical
↪capacity of suspensions with special reference to blood" by Fricke, 1925"
    },
    "cell_mass_pg": {
      "value": 1.048,
      "reference": "Lewis, C. L., Craig, C. C., & Senecal, A. G. (2014). Mass and density
↪measurements of live and dead Gram-negative and Gram-positive bacterial populations.
↪Applied and environmental microbiology, 80(12), 3622-3631. https://doi.org/10.1128/AEM.
↪00117-14"
    },
    "cell_volume_fL": {
      "value": 0.9357,
      "reference": "Lewis, C. L., Craig, C. C., & Senecal, A. G. (2014). Mass and
↪density measurements of live and dead Gram-negative and Gram-positive bacterial
↪populations. Applied and environmental microbiology, 80(12), 3622-3631. https://doi.
↪org/10.1128/AEM.00117-14"
    },
    "eps_oxidation_rate_constant":{

```

(continues on next page)

(continued from previous page)

```

    "value": 37.75,
    "reference": null,
    "notes": "This rate constant was empirically determined after calibrating the_
    predictions with the Beirao et al., 2014 paper that constituted one of our examples"
  },
  "cellular_dry_mass_proportion_biofilm":{
    "value": 0.1,
    "reference": "The biofilm matrix; Flemming et al.; 2010"
  },
  "doubling_rate_constant":{
    "value": 0.00038441,
    "reference": "Baines et al.; mBio; 2015"
  },
  "biofilm_oxidation_fraction_lysis":{
    "value": 0.0014125,
    "note": "empirically derived through training with Beirao et al."
  }
}

```

- *membrane_chemicals* dict: specifies the fatty acid constituent the phospholipids of the bacterial cytoplasmic membrane. Each fatty acid (FA) entry is defined with sub-dictionaries of its chemical formula, its density_gL density, and its proportion of total FAs in the cytoplasmic membrane.
- *membrane_thickness_nm* dict: specifies the thickness of the cytoplasmic membrane in nanometers.
- *cell_mass_pg* & *cell_volume_fL* dict: specifies the mass and volume of the bacterial cell in picograms and femtoliters, respectively.
- *eps_oxidation_rate_constant* dict: defines the rate constant of oxidizing the extracellular polymeric substance for biofilm simulations of this organism.
- *cellular_dry_mass_proportion_biofilm* dict: defines the ratio of biofilm mass that is comprised of cellular dry mass.
- *doubling_rate_constant* dict: specifies the rate constant at which the bacteria doubles
- *biofilm_oxidation_fraction_lysis* dict: specifies the threshold of membrane oxidation that manifests in cellular lysis.

The reference and notes keys are optional in each sub-dictionary, as are other keys of metadata at the user's discretion.

2.3.4 light

The emission spectrum and visual intensity per energy input of the simulated light source are defined through the `light.json` file, complementarily to the function argument. The default light sources are incandescent, LED, and fluorescent; however, other light sources can be defined by emulating the same syntactic structure. The values of each sub-dictionary are stored in the value key.

```

{
  "incandescent": {
    "visible_proportion": {
      "value": 0.1,
      "reference": "Macisaac et al., 1999"
    }
  },

```

(continues on next page)

(continued from previous page)

```
"lumens_per_watt": {
  "value": 3,
  "reference": "Michael F. Hordeski. Dictionary Of Energy Efficiency Technologies.↵
↵Fairmont Press. ISBN: 9780824748104"
}
}
}
```

- *visible_proportion* dict: specifies proportion of the emission spectrum that resides within the visible region.
- *lumens_per_watt* dict: specifies the visual lumens that are emitted per watt of energy. This is used to convert between parameterized light intensity in units of lux or lumens into watts, which is the necessary unit for subsequent PDIpy calculations.