

---

# **svo\_filters Documentation**

*Release 0.2.16*

**Joe Filippazzo**

**Aug 12, 2022**



---

## Contents

---

<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Load a Photometric Filter</b>	<b>5</b>
<b>3</b>	<b>Load a Grism</b>	<b>7</b>
<b>4</b>	<b>Apply a Filter to a Spectrum</b>	<b>9</b>
<b>5</b>	<b>Contents</b>	<b>11</b>
5.1	svo module . . . . .	11



The [Spanish Virtual Observatory \(SVO\) Filter Profile Service](#) is a great resource for homogenized photometric filter curves and metadata. With *svo\_filters*, I tried to create a lightweight and flexible package to incorporate these filters into Python applications.



# CHAPTER 1

---

## Installation

---

To install **svo\_filters**, do:

```
pip install svo_filters
```

Alternatively, you can clone from Github with:

```
git clone https://github.com/hover2pi/svo_filters.git
python svo_filters/setup.py install
```





---

### Load a Photometric Filter

---

The actual filters are stored locally as XML files and can be viewed with:

```
from svo_filters import svo
svo.filters()
```

To create a filter object, pass a bandpass name to the `svo.Filter()` class:

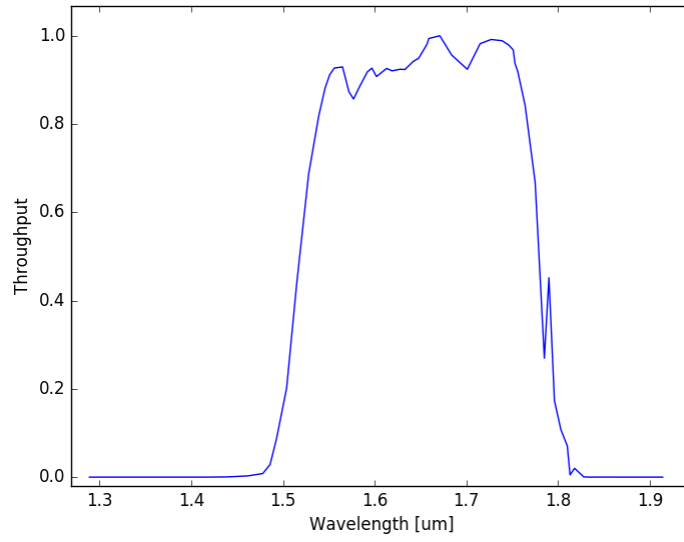
```
H_band = svo.Filter('2MASS.H')
```

You can see some information about the filter with:

```
H_band.info()
```

And you can plot the bandpass like so:

```
H_band.plot()
```



## CHAPTER 3

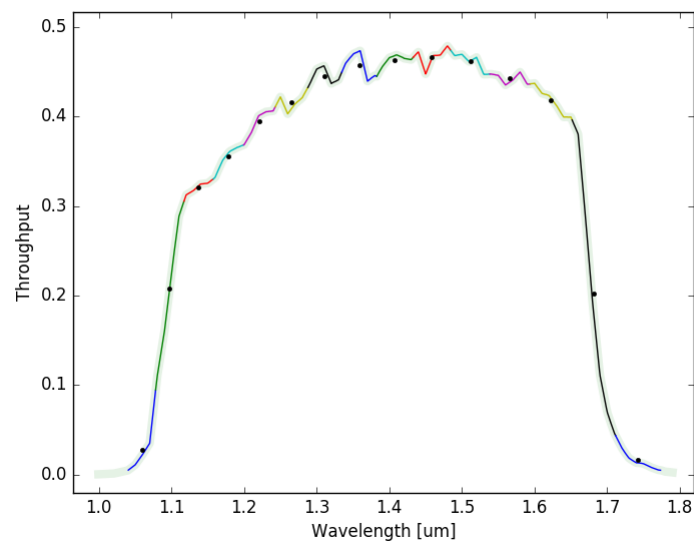
---

### Load a Grism

---

Filters can also be binned arbitrarily, for use with grisms. We can pass integers to the `n_bins` or `pixels_per_bin` arguments to specify the number of wavelength bins or pixels per bin, respectively:

```
G141 = svo.Filter('WFC3_IR.G141', n_bins=15)
```





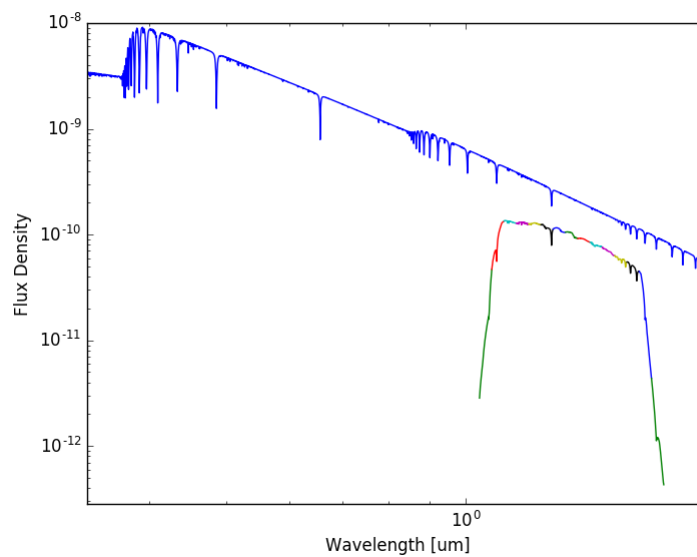
---

## Apply a Filter to a Spectrum

---

Filters can be applied to a spectrum by passing a sequence of [W, F] or [W, F, E] with astropy units to the `apply()` method:

```
filtered = G141.apply(spec, plot=True)
```





## 5.1 svo module

This is the main module for *svo\_filters*.