

# DOLPHOT/Roman User's Guide

version 3.0

Andrew Dolphin

adolphin@rtx.com

June 2025

<http://americano.dolphotsim.com/dolphot>

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Installation</b>	<b>3</b>
2.1	Makefile . . . . .	3
2.2	PSFs . . . . .	3
<b>3</b>	<b>Preprocessing Steps</b>	<b>4</b>
3.1	<i>romanmask</i> : Convert to DOLPHOT Usable Format . . . . .	4
3.2	<i>calcsky</i> : Calculate Sky for Image . . . . .	4
<b>4</b>	<b>Running DOLPHOT</b>	<b>5</b>
4.1	DOLPHOT Parameters . . . . .	5
4.2	DOLPHOT Output . . . . .	7
<b>5</b>	<b>PSF Utility Reference</b>	<b>8</b>
5.1	<i>romanmakepsf</i> : Compute PSF for Filter . . . . .	8
5.2	<i>romanshowpsf</i> : Display PSF for Filter . . . . .	8
<b>6</b>	<b>Recipe for Roman Photometry</b>	<b>9</b>
<b>7</b>	<b>Changes to DOLPHOT/Roman and WFIRST</b>	<b>10</b>

# 1 Introduction

This manual describes installation and use of the Roman module for DOLPHOT. The Roman module replaces the analytic PSF model with a lookup table computed using Tiny Tim PSFs.

If you have any problems installing or using DOLPHOT or the provided utilities, please let me know.

Note that the current release of the Roman module supports photometry of STIPS generated (simulated) images only. Note that supersampling of exactly 9x may result in overly optimistic predictions of the photometry performance, given that the Roman module's PSF library is generated at 9x supersampling.

Andrew Dolphin  
adolphin@rtx.com

## 2 Installation

Installation of the Roman module is done in addition to the regular DOLPHOT installation. See the DOLPHOT manual for installation instructions for DOLPHOT.

The Roman module requires less than 1Mb for the sources and binaries, plus less than 1Mb per set of PSFs. Note that the PSFs are distributed separately, as you only need PSFs for the filters that you will actually be using.

The .tar.gz file for the Roman module (as well as the PAMs and PSFs) should be expanded from the same directory from which the DOLPHOT distribution was expanded.

### 2.1 Makefile

The Makefile is the same as the regular DOLPHOT installation, but includes several options used by the Roman module. To enable the Roman module, the three lines beginning with the Roman definition should be uncommented.

No customization of the Makefile within the “roman” subdirectory is needed.

After editing the makefile, you should be able to type “make” to compile DOLPHOT without any error or warning messages. Should you need to later recompile the programs, typing “make clean” will remove the original compilations to ensure that all programs get recompiled.

### 2.2 PSFs

You can either make your own PSFs or use the precomputed ones. Generally there is no need to generate your own, as precomputed PSFs are available for all filters. Instructions for computing PSFs are not currently available.

### 3 Preprocessing Steps

For the Roman module to run correctly, data must be in the default format created by the image simulation tool STIPS. This is a single chip of data (stored in an extension), with 4096 x 4096 pixels. Stage 3 files (equivalent to HST drizzled images) are not currently supported.

This section takes you through the necessary steps (in order) for obtaining the best possible photometry from DOLPHOT.

#### 3.1 *romanmask*: Convert to DOLPHOT Usable Format

```
romanmask <fits files>
```

```
romanmask -exptime=<exptime> -rdnoise=<readout noise> <fits files>
```

*Romanmask* takes as input FITS files in the format created by STIPS. (one-extension files with a 4096x4096 image).

The `-exptime` flag will override the exposure time in the header. This should generally not be used with STIPS-created images, unless images have been coadded by another software tool.

The `-rdnoise` flag will override the readout noise in the header. This should generally not be used with STIPS-created images, unless images have been coadded by another software tool.

Before running *romanmask*, make sure you have backed up the original images, since *romanmask* will alter them!

#### 3.2 *calcsky*: Calculate Sky for Image

```
calcsky <fits base> < $r_{in}$ > < $r_{out}$ > <step> < $\sigma_{low}$ > < $\sigma_{high}$ >
```

*Calcsky* is described in the DOLPHOT manual. Recommended parameters are as follows:

$$r_{in} = 10$$

$$r_{out} = 25$$

$step = 2$  for an accurate sky map, or  $step = -64$  for a quick one.

$$\sigma_{low} = 2.25$$

$$\sigma_{high} = 2.00$$

Since I always use non-zero FitSky settings in my photometry, the negative values of *step* are sufficient.

Note that running *calcsky* is currently mandatory, even if using FitSky settings other than 0.

## 4 Running DOLPHOT

### 4.1 DOLPHOT Parameters

The Roman module of DOLPHOT will be invoked whenever processing images that have been run through romanmask. When using the Roman module, several parameters are disabled or have restricted ranges.

img\_aprad is set to 10 pixels

img\_RPSF and img\_RAper cannot exceed 24 pixels

Zero disabled (set to ABMAG zero points)

FPSF, SubPixel, img\_psfa, img\_psfb, img\_psfsc disabled

EPSF, PSFsol, PSFStep disabled

MinS, MaxS, MaxE disabled

Recommended general-use settings for other parameters follow. Note that these were optimized across a variety of environments ranging from uncrowded to extremely crowded. For uncrowded fields, other parameter options involving larger photometry radii (img\_RAper and img\_RChi) and other sky options (FitSky = 1 or FitSky = 3) can produce very good results.

Recommended values for other parameters are as follows:

PSFPhot = 1

PSFPhotIt = 2

Force1 = 0

img\_apsky = 20 35

img\_RAper = 5

img\_RChi = 2

img\_RSky = 15 35

img\_RPSF = 15

SkipSky = 1

SkySig = 2.25

SecondPass = 5

SigFindMult = 0.85

```
MaxIT = 25
NoiseMult = 0.10
FSat = 0.999
ApCor = 1
RCentroid = 1
PosStep = 0.25
RCombine = 1.5
SigPSF = 5.0
PSFres = 1
```

Running with `FitSky = 2` is recommended in crowded fields. In this case, recommended parameter changes to the above are as follows:

```
img_RAper = 2
img_RChi = 1.5
img_RSky2 = 3 10
```

If using `FitSky = 3`, the parameters are largely the same except that suggested `img_RAper` values are 8 pixels.

In some cases, blends or extended objects can hamper photometry of nearby stars. Setting `Force1 = 1` will solve this, but of course will also result in false detections of hot pixels and extended objects as single stars. So one may have to choose between a very clean but slightly incomplete CMD and a complete but contaminated one.

Finally, there are four new parameters that can be used:

```
FlagMask = 4
CombineChi = 0
InterpPSFlib = 1
RomanVega = 1
```

`FlagMask` is a bitwise mask that determines what error flags will not be accepted when producing the combined photometry blocks for each filter. Note that error flag values of eight or more (when the “extreme case”) always cause the photometry to be ignored. A value of zero allows photometry with an error flag less than eight to be used. Adding one eliminates stars close to the chip edge, adding two eliminates stars with too many bad pixels, and adding four eliminates stars with saturated cores.

CombineChi also affects the combined photometry blocks. If set to zero (default), photometry will be combined weighted by  $1/\sigma^2$  to maximize signal to noise. If set to one, weights will be  $1/\sigma^2 \max(1, \chi^2)$  to reduce the impact of epochs with bad photometry. Note that using CombineChi of one will require tuning NoiseMult so that well measured stars have  $\chi = 1$  at all magnitudes (plots of chi vs. magnitude should show this). Note also that this will result in larger uncertainties for combined (but not individual image) magnitudes and normalized count rates, as the individual image uncertainties are effectively multiplied by  $\chi$  when calculating combined magnitudes.

If InterpPSFlib is set to 0, the PSF library will use the nearest X,Y position where a precalculated PSF is available rather than interpolating. The impact is  $\approx 1\%$  on the PSF shape but some speed improvement.

If RomanVega is set to 0, instrumental magnitudes will be provided in units of ABmag. Otherwise, when set to 1, Vega magnitudes are used.

## 4.2 DOLPHOT Output

The DOLPHOT/Roman output is virtually identical to the regular DOLPHOT output, with a few exceptions. First, if multiple images exist per filter, it will insert extra sets of photometry for each such filter (in order of wavelength) prior to the individual-image photometry. Following any combined-filter photometry are the single-image photometry blocks, in order that the images are provided in the parameter files.

Within the photometry blocks, there are also some differences. First is that calibrated instrumental magnitudes and transformed magnitudes are provided. Note that transformations are not yet supported, so the second magnitude is always 99.999 in the current DOLPHOT release. Also, a calibrated count rate and uncertainty is provided (effectively the instrumental magnitude and uncertainty expressed as a count rate).

As with all photometry, you will need to trim your detection list to eliminate objects classified as non-stellar, low signal-to-noise, or bad photometry quality. See the DOLPHOT manual for further discussion.



## 5 PSF Utility Reference

The following programs are utilities included with the Roman module. Since there is little need for computing custom PSFs, no documentation is provided – use at your own risk!

### 5.1 *romanmakepsf*: Compute PSF for Filter

```
romanmakepsf <filter>
```

### 5.2 *romanshowpsf*: Display PSF for Filter

```
romanshowpsf <filter> <chip> <X/4096> <Y/4096> <dX> <dY> <<flags>>
```

## 6 Recipe for Roman Photometry

Assuming that an appropriate reference image exists, the recipe for generating photometry is reasonably simple.

1. Run `romanmask` on all Roman images.
2. Run `calsky` on each image to generate `*.sky.fits` files.
3. Run `dolphot`.
4. If artificial star tests are to be run, use `romanfakelist` to generate the star list and re-run `dolphot` with the `FakeStar` parameter used to trigger artificial star mode.

## 7 Changes to DOLPHOT/Roman and WFIRST

August 26, 2017 release

1. Initial release

February 3, 2018 release

1. Changed zero points from ABmag to VEGAMAG
2. Improved calculation of combined magnitudes and uncertainties

September 12, 2020 release

1. Updated zero points based on current STIPS
2. Added spatially varying PSFs
3. Added encircled energy corrections

June 6, 2025 release

1. Revised ROMAN utilities to use current STIPS-created simulated data products.
2. Updated PSFs and photometric calibrations to match current versions of STPSF and STIPS.
3. Switched PSFs to higher resolution. This will require downloading new PSFs, as previous distributions will not work with the current code.