

MockGals

Make mock stars and galaxies in a FITS image with noise.
Manual for MockGals version 0.1

Mohammad Akhlaghi

This manual is for MockGals, a program to make mock astronomical objects in a FITS image and add the appropriate noise.

Copyright © 2014 Mohammad Akhlaghi: `akhlaghi@astr.tohoku.ac.jp`

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is available online at GNU FDL webpage (<http://www.gnu.org/copyleft/fdl.html>).

Table of Contents

1	About	1
1.1	Objective/critical science and its software	1
1.2	Your rights as a user of MockGals	2
1.3	Manual conventions and summary	2
1.4	Manual Formats and reading recommendations.....	3
2	A Quick start	4
3	Introduction	5
4	Installation.....	6
5	Configuring MockGals	7
6	Mock parameters	8
7	How mockgals works	9
	Index.....	10

1 About

To get started with MockGals please see Chapter 2 [A Quick start], page 4. The following sections in this chapter are more general discussions.

In the first section we will give a short review of the way modern science has become intertwined with software and how important it is for scientists to have very good understanding of the software they use. After that your rights as the reader of this document and user of MockGals are explained and in the end of this chapter we give a short review of how this manual is available for your use and a short summary of the manual.

1.1 Objective/critical science and its software

We believe that if a software (a technique) is to be used in scientific research, it has to be open source and transparent (with a good manual) to be open to scrutiny. Any critical/interested user should easily be able to follow all the steps and understand all the fine details and assumptions. History of science shows us that no interpretation, model or technique (software) is perfect. There are always inevitably unseen faults, hidden assumptions, simplifications and approximations in any of our constructs. It is precisely these that will ultimately allow future generations to find/correct them and advance our experimental and theoretical knowledge of the universe we live in. By keeping software free and transparent for all scientists, we can most accurately convey our experiences to users who can build on those or find their shortcomings. Viewed more generally, this will help critical and objective science thrive.

With the progress of technology, scientific results are increasingly becoming more and more dependent on hardware and software techniques. Therefore obscure software is arguably the biggest roadblock in our aim to achieve this goal and a modern scientist cannot claim to be curious in understanding the universe and our relation to it, whatever their specific field, without accurately understanding the software techniques they use to quench that curiosity. We thus believe that the wide use of free and transparent software is one step in a larger scheme to save the sciences...

...from narrow specialization and from an obscurantist faith in the expert's special skill and in his personal knowledge and authority; a faith that so well fits our post-rationalistic and post critical age, proudly dedicated to the destruction of the tradition of rational philosophy and of rational thought its self.

—Karl Popper. *The logic of scientific discovery*. 1959.

To reach this goal, namely to let the users of a software be able to understand its intricate details at their will, a software has to be accompanied by a good and up to date manual. The user of a software cannot claim to understand how it works only based on the experience they have gained by frequently using it. This kind of subjective experience is prone to very serious misunderstandings that only help in producing dogmas and the above mentioned “obscurantist faith in the expert's special skill”. Unfortunately in the commonly used astronomical softwares, most are either closed source or don't have a good manual. The scientific results that are deduced from such software are thus not objective, because in the former case scientists have to look at it as a black box and trust its results on faith. In the latter, it is very hard to find enough time to understand the programming of an open source project without a clearly written and up to date manual explaining all the ideas behind the major functions, data structures or objects.

It is certainly time consuming for the author of a software to make such a manual. If personal greed is put aside and our greater common experience is considered, that time is a time well invested. Because through sharing of these fundamental ideas, that shape all our scientific results, and keeping them open to criticism, we can make them more robust and thus take better steps in our common goal to tackle the riddles of nature.

Bjarne Stroustrup (creator of the C++ language) says: “Without understanding software, you are reduced to believing in magic”. Ken Thomson (the designer of the Unix operating system) has said very nicely that “I abhor a system designed for the “user” if that word is a coded pejorative meaning “stupid and unsophisticated”.” Both these cases naturally apply even more seriously to systems written for scientific applications and as scientists we should be loyal to such principles if we want to remain critical and be objective. After all, all our data and models are produced, archived and processed using hardware and software. We cannot and should not accept any result through faith in the programmer who wrote that particular program we are using to publish our papers.

1.2 Your rights as a user of MockGals

MockGals is “free software”; this means that everyone is free to use it and free to redistribute it on certain conditions. MockGals is not in the public domain; it is copyrighted and there are restrictions on its distribution, but these restrictions are designed to permit everything that a good cooperating citizen would want to do. What is not allowed is to try to prevent others from further sharing any version of MockGals that they might get from you.

Specifically, we want to make sure that you have the right to give away copies of the programs that relate to MockGals, that you receive source code or else can get it if you want it, that you can change these programs or use pieces of them in new free programs, and that you know you can do these things.

To make sure that everyone has such rights, we have to forbid you to deprive anyone else of these rights. For example, if you distribute copies of the MockGals related programs, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

Also, for our own protection, we must make certain that everyone finds out that there is no warranty for the programs that relate to MockGals. If these programs are modified by someone else and passed on, we want their recipients to know that what they have is not what we distributed, so that any problems introduced by others will not reflect on our reputation.

The precise conditions of the licenses for the programs currently being distributed that relate to MockGals are found in the General Public license¹ that accompany them. This manual is covered by the GNU Free Documentation License².

The paragraphs above, in this section, belong to the GNU Texinfo manual and are not written by us! We have just changed the name “Texinfo” to “MockGals” because it and this manual are released under the same licenses and it is beautifully written to inform you of your rights.

1.3 Manual conventions and summary

In this manual commands to be input into the command line as normal user begin with **\$**, and for those that to be input as root (or super-user), the command begins with **#**. Note that you need the root password in order to execute the latter commands. The commands finish where the special type face used to show them finishes. Be careful not to include text punctuation marks that might follow the inline commands, they have a different font. After typing the commands on the command-line you have to press the **Enter** or **Return** key on your keyboard for the command to start executing.

The first chapter is a quick start, explaining the generalities of MockGals so the user can begin using it very fast. The following chapters give a much more detailed review of each step.

¹ <https://www.gnu.org/copyleft/gpl.html>

² <https://www.gnu.org/copyleft/fdl.html>

1.4 Manual Formats and reading recommendations

This manual is available in plain text, HTML, Info, Docbook, PDF and finally the Texinfo format. The last one is the basis for creating all the other formats. As an initial introduction the PDF and HTML formats might be easier to begin using MockGals, but if you would like to understand the code while reading Chapter 7 [How mockgals works], page 9, it is much easier to do it in the info format, where for example in Emacs, you can have one window on the code and another on the info file.

Info is a very convenient format particularly useful for software documentation. We strongly recommend becoming familiar with it if you are not already because it can significantly enhance your experience in programming and in your GNU based operating systems user experience. Operating systems like Ubuntu, Fedora, Linux Mint, Debian, OpenSUSE, Red Hat, CentOS or Scientific linux among many others. Info is very trivial and made for reading manuals. It provides facilitates to navigate easily in long manuals. To learn more about it in any GNU/linux terminal you can run: `$ info info`. It does a fantastic job in explaining its self in practice interactively.

By default, the PDF version of this manual is prepared in the A4 paper size (210cm by 297cm). In case you prefer a letter size (8.5in by 11in), you can simply change it by following the steps below. You must have Texinfo and T_EX installed on your system, for an explanation on how to install them see Chapter 4 [Installation], page 6.

1. In the downloaded source directory (lets call it SRC/), open the SRC/doc/mockgals.texinfo Texinfo source file of this manual.
2. Delete the line that says `@afourpaper`. It is in the first few lines of the file.
3. run `$./run.sh` in that same directory, this will re-make all the documenation formats.

2 A Quick start

Once MockGals is installed, if you simply run `./mockgals` in the installed directory, 45 mock galaxies and 5 stars will be randomly positioned in a FITS image of size 200 * 200.

3 Introduction

Making mock galaxies is very important in the process of understanding our data. MockGals was initially made with this exact intent. Certain astronomical targets, for example elliptical galaxies, are very sharp in their central regions, this makes a simple calculation of the profile in the center of each pixel unrealistic for such cases. The main advantage of MockGals is that it integrates the central parts of profiles until a given accuracy. It does this without any sorting or ordering and in a very fast manner.

A summary of the advantages of MockGals includes:

1. Integration of the center of the profile.
2. Very efficient in CPU usage, resulting in a very fast processing.
3. Written in the C programming language, which is easy to understand and modify or contribute to by any interested user.
4. [To be added] Can make profiles in any dimensions.

4 Installation

MockGals relies on only 3 packages: GSL (GNU scientific library, for mathematical functions), FFTW (for convolution) and cfitsio (for reading to and from FITS files).

5 Configuring MockGals

Currently configuration parameters into MockGals can only be given through single letter commandline options. The options are explained below.

6 Mock parameters

The parameters of the profiles you want to build into a FITS image are fed to it through a table. In this section we will review how MockGals reads that table and how best to prepare it.

7 How mockgals works

In this section we will give a complete review on how MockGals works and how this is implemented in its coding in order to facilitate your reading of the code and possibly modifying it and making it better.

Index

(Index is nonexistent)