# PROFIT: Bayesian profile fitting of galaxy images

## A. S. G. Robotham,[1]★ D. S. Taranu,[1,2] R. Tobar,[1] A. Moffett[1] and S. P. Driver[1]

[1]*ICRAR, M468, University of Western Australia, Crawley, WA 6009, Australia*
[2]*CAASTRO: ARC Centre of Excellence for All-sky Astrophysics, Building A28, School of Physics, The University of Sydney, NSW 2006, Australia*

## ABSTRACT

We present PROFIT, a new code for Bayesian two-dimensional photometric galaxy profile modelling. PROFIT consists of a low-level c++ library (`libprofit`), accessible via a command-line interface and documented API, along with high-level R (PROFIT) and PYTHON (PyProFit) interfaces (available at github.com/ICRAR/libprofit, github.com/ICRAR/ProFit, and github.com/ICRAR/pyprofit, respectively). R PROFIT is also available pre-built from CRAN; however, this version will be slightly behind the latest GitHub version. `libprofit` offers fast and accurate two-dimensional integration for a useful number of profiles, including sérsic, Core-sérsic, broken-exponential, Ferrer, Moffat, empirical King, point-source, and sky, with a simple mechanism for adding new profiles. We show detailed comparisons between `libprofit` and GALFIT. `libprofit` is both faster and more accurate than GALFIT at integrating the ubiquitous sérsic profile for the most common values of the sérsic index $n$ ($0.5 < n < 8$). The high-level fitting code PROFIT is tested on a sample of galaxies with both SDSS and deeper KiDS imaging. We find good agreement in the fit parameters, with larger scatter in best-fitting parameters from fitting images from different sources (SDSS versus KiDS) than from using different codes (PROFIT versus GALFIT). A large suite of Monte Carlo-simulated images are used to assess prospects for automated bulge-disc decomposition with PROFIT on SDSS, KiDS, and future LSST imaging. We find that the biggest increases in fit quality come from moving from SDSS- to KiDS-quality data, with less significant gains moving from KiDS to LSST.

**Key words:** methods: data analysis – methods: statistical – techniques: photometric – galaxies: fundamental parameters – galaxies: statistics – galaxies: structure.

## 1 INTRODUCTION

Galaxy modelling can be broadly separated into two categories: light profile fitting and kinematic dynamical modelling. In dynamical modelling, part or all of the six-dimensional phase space structure is constrained by kinematic data and dynamical constraints such as the Poisson and/or Jeans equations. By contrast, profile fitting (or profiling) quantifies the *projected* structure of a galaxy on the sky, usually by fitting radially varying density profiles to one-dimensional, azimuthally averaged profiles or two-dimensional images. Both methods have been used to infer the presence of distinct structural components in galaxies, such as discs, bulges, bars, and spiral arms (so-called decomposition).

Hubble (1926) is credited as the first to develop a galaxy classification scheme including distinct features or components, including spiral arms and a concentrated central component. As early as de Vaucouleurs (1958), it had been quantitatively demonstrated, using photoelectric measurements along the major and minor axes, that

M31 comprised a distinct bulge and disc component. In the modern era, large samples of thousands of galaxies have been decomposed quantitatively into distinct components (Allen et al. 2006; Gadotti 2009; Simard et al. 2011; Kennedy et al. 2016; Lange et al. 2016). Such data sets have been used to infer distinct evolutionary pathways and mechanisms (Kormendy & Kennicutt 2004) for galaxies of different types and compositions and to develop galaxy formation models (Zavala, Okamoto & Frenk 2008; Driver et al. 2013; Lacey et al. 2016). Improving these theories requires both better data (by volume, depth, and quality) and analysis methods. This paper offers a new route to reliably divide galaxies into their constituent parts using a new publicly released library and interface (`libprofit` and PROFIT, respectively) for galaxy profile modelling and decomposition. These will be used by the authors for near future large-scale studies, and are being made available to the wider community along with the prospect of practical support and future updates.

The earliest efforts of galaxy structural characterization concentrated on simple one-dimensional intensity profile fitting, leading to the early discovery of the de-Vaucouleurs profile (de Vaucouleurs 1948) for describing early-type galaxies and the realization

★ E-mail: aaron.robotham@uwa.edu.au

that disc galaxies had close to exponential profiles (de Vaucouleurs 1959). Sérsic (1963) generalized these profiles into the $r^{1/n}$ law that remains widely popular today, commonly referred to as the sérsic profile (Graham & Driver 2005). Increasingly complicated one-dimensional component fitting came in work by Kormendy (1977) and Kent (1985), where galaxies were decomposed into distinct components rather than treated as having a single light profile.

The earliest efforts at two-dimensional galaxy profiling came with the work of Andredakis, Peletier & Balcells (1995), Byun & Freeman (1995), de Jong (1996), Schade et al. (1995), and Wadadekar, Robbason & Kembhavi (1999). The latter four approaches are broadly similar in application, and are recognizably similar to modern two-dimensional galaxy fitting efforts. The basic philosophy was the same as for previous one-dimensional work, the aim being to find the distinct light components of galaxies, but here the analysis was made using image pixels directly rather than by fitting azimuthally averaged ellipses of light.

A large number of key insights into galaxy properties have been uncovered through both one-dimensional and two-dimensional structural analysis. Early work found that the disc component of galaxies is very well represented by an exponential drop-off in light (e.g. Freeman 1970; Kormendy 1977). Of recent interest in the astronomy literature is the relationship between galaxy mass and size, both globally (Shen et al. 2003; Lange et al. 2015) and for individual components (Lange et al. 2016). There is also a large body of work investigating the luminosity–surface-brightness relation for different classes of galaxy (see reviews by Ferguson 1994; Graham 2013), as well as investigating the distribution of mass for different types of structure (e.g. Dressler 1987; Benson et al. 2007; Driver et al. 2007; Kelvin et al. 2014; Kennedy et al. 2016; Moffett et al. 2016a,b).

Such global morphological properties of galaxies are finally being utilized and predicted by the newest theoretical work, both in the regime of semi-analytic models (Lacey et al. 2016) and cosmological hydrodynamic simulations (Vogelsberger et al. 2014; Schaye et al. 2015). Given the constraints that galaxy morphology can offer theory, it will be increasingly important that well-quantified structural measurements for galaxy properties are extracted from current and future large-area, deep, high-resolution photometric surveys (e.g. Kilo Degree Survey (KiDS), Dark Energy Survey (DES), Hyper Suprime Camera (HSC), and Large Synoptic Survey Telescope (LSST): Kuijken et al. 2015; The Dark Energy Survey Collaboration 2005; Miyazaki et al. 2013; Abell et al. 2009, respectively).

There remains some debate regarding the relative merits of one-dimensional and two-dimensional image analysis (e.g. Savorgnan & Graham 2016). This introduction will not serve as a thorough discussion of the various issues, but we will discuss a few key points. One-dimensional fitting relies heavily on an additional step in the analysis: the galaxy must be collapsed into a one-dimensional profile in some manner, usually using software such as IRAF's ELLIPSE task. It is impossible to do this in an entirely model-independent manner when the galaxy is anything other than smoothly changing isophotes; however, such an approach can easily cope with a smoothly varying orientation in the isophotes. A caveat to this is that it is unclear formally how the atmospheric point-spread function (PSF) should be treated in one-dimensional fitting (note that there are analytic solutions for the specific case of face-on convolution, see Pritchet & Kline 1981). Even a galaxy constructed from perfect concentric ellipses will become artificially circular towards the profile centre when convolved with a circular PSF. Whilst this effect is captured from the one-dimensional profile information, it is non-

trivial to properly propagate this information such that you measure the intrinsic two-dimensional profile properties of the galaxy. Collapsing two-dimensional information into a one-dimensional form is almost always a lossy process (it can be no better than lossless), and real azimuthal profile fluctuations may be lost entirely. Correctly propagating errors is also non-trivial, and in practice these are often not used at all (an approach advocated in Savorgnan & Graham 2016).

There are two other critical aspects of this collapsing process that can have a dramatic effect on galaxy profiling: the centre of the galaxy must be fixed exactly a priori and the components must share the same centre. The freedom to fit for the galaxy centre is often important for capturing the steepest inner parts of a galaxy properly (especially if the centre is sub-pixel) and galaxy components sometimes do not share exactly the same centres (Lange et al. 2015).

Two-dimensional fitting is closer in spirit to Bayesian generative model fitting, e.g. creating a two-dimensional distribution of flux that is then propagated through the atmospheric PSF. As such, it is an attractive approach to quantifying image components. However, the isophotes of popular galaxy profile functions are poorly behaved when integrating over two-dimensional pixels, requiring relatively expensive numerical estimation, while complex geometry (like rotating isophotes) is more difficult to capture fully, with most codes ignoring this issue entirely. In the regime where the true galaxy is well represented by the two-dimensional model being used, a two-dimensional fit *should* be preferable. In principle, effects such as twisting isophotes can be incorporated into a two-dimensional generative model technique (see Peng et al. 2010), but this is rarely attempted in practice.

Because of these relative advantages and disadvantages between one-dimensional and two-dimensional approaches, their application to galaxy profiling has pragmatically diverged. One-dimensional codes are more popular in the regime of highly resolved galaxies where complex geometric effects (such as twisting isophotes) are clearly visible and the issue of PSF convolution is less significant. Because of the more automated nature of two-dimensional codes (they do not require a carefully controlled collapse of the data to one dimensional), they have tended to be more popular for large samples of galaxies where the fitting is made with little user interaction (e.g. Simard et al. 2002; Allen et al. 2006; Häußler et al. 2007; Kelvin et al. 2012; Häußler et al. 2013). Our long-term ambition is to decompose galaxy structures from a large number of imaging data sources for many millions of galaxies, so our use-case is squarely in the highly automated regime. For this reason in particular, we focused our efforts on two-dimensional decomposition codes.

Current publicly available two-dimensional galaxy profile fitting codes include BUDDA (de Souza, Gadotti & dos Anjos 2004), GALFIT (Peng et al. 2010), GALMORPH (Hyde 2009), GIM2D (Simard et al. 2002), and IMFIT (Erwin 2015) – all of which support the commonly used sérsic profile. These codes share their origins in the early two-dimensional profiling efforts of Wadadekar et al. (1999) and Benson, Frenk & Sharples (2002) and have been utilized in a number of notable large-scale galaxy studies: Allen et al. (2006), Gadotti (2009), Hyde & Bernardi (2009), Simard et al. (2011), Kelvin et al. (2012), Häußler et al. (2013), Meert, Vikram & Bernardi (2015), Lange et al. (2016), and Kennedy et al. (2015, 2016). Notable alternatives include GASPHOT (Pignatelli, Fasano & Cassata 2006) which is a one-dimensional fitting code, and MGE (Cappellari 2002) which abandons traditional sérsic profile bulge-disc decomposition for more flexible (but arguably less physical) concentric Gaussians. In all of the other codes, the model is treated generatively by numerically integrating a radially varying profile over a rectangular

pixel grid; in practice, most codes improve integration accuracy by oversampling the native pixel grid, particularly near the centre of the galaxy. Once this idealized model is integrated, it can be convolved with a target PSF, either via brute-force or fast Fourier transform (FFT) techniques. These codes all have their positive features and weaknesses, and differ in fine aspects such as how the likelihood is computed and maximized. However, no existing code achieves the desired design requirements that our new code, PROFIT, aims to address, namely that it:

(i) should be legally open source under a GPL, BSD, MIT license, or similar;

(ii) should be practically collaborative and hosted in a publicly available location such as GitHub or BitBucket;

(iii) should be written in a low-level modern object-oriented language, aiding modularity, extensions, and robustness for complex software;

(iv) has the core image generation code clearly separated from the optimization code, available as a shared library, and accessible via an easy-to-use API;

(v) offers a useful range of built-in models and have a well-documented mechanism for adding more;

(vi) supports sub ~1 per cent flux weighted error model image generation for built-in models, and be testable in this regard;

(vii) should be fast at generating these model images;

(viii) has a range of options to define likelihoods, i.e. Normal, Poisson, Chi-squared, and Student-T statistics;

(ix) should be able to fit a large number of components in a flexible manner, e.g. components being fixed, free, or locked together with other components;

(x) should be untied to a specific optimizer, i.e. able to use a range of (user-selectable) optimization engines;

(xi) should be able to accept prior distributions of an arbitrary type for model parameters (vital for Bayesian model evaluation);

(xii) has the ability to fit parameters in a linear or log space as deemed appropriate by the user for the problem (vital for Bayesian model evaluation);

(xiii) should be able to handle parameter limits and constraints in flexible manner (e.g. prevent some parameters getting larger than other etc.);

(xiv) has a core interface that does not require the use of text file inputs, i.e. there is a completely functional interactive interface to the software (though scripting with files should be allowable);

(xv) is simple to parallelize.

The reasons for desiring the above features are multifaceted, but broadly it is to aid the structural analysis of large amounts of data obtained from new and future imaging surveys such as KiDS, DES, HSC, and LSST. Also, the addition of optional priors and log or linear parameter scaling is paramount for proper Bayesian model evaluation. Automating a historic code is a complicated affair (Kelvin et al. 2012; Häußler et al. 2013), and there are in practice many subtle issues with trying to bootstrap a new code from legacy software. Starting from scratch with a new code offers full flexibility in language choice, design flexibility, and low-level choice regarding pixel integration and likelihood calculations. Our estimation was it would be at least as much effort to adapt any available open-source code to meet the above requirements that PROFIT delivers.

This paper discusses the development and application of the core image generation code (libprofit, Section 2) and the fitting code (PROFIT, Section 3). The PROFIT code is then applied to various examples (Section 4), spanning a detailed case study (Section 4.1), and the analysis of 10 exemplar bulge-disc galaxies that are included

with the PROFIT package (Section 4.2). Estimations of decomposition fidelity is then made for current and future data sets (Section 5). Finally, we discuss and summarize the results of this paper (Section 6).

## 2 IMAGE GENERATION WITH `libprofit`

The image generation code is contained in a shared library written in c++ called `libprofit`. `libprofit` enables the user to construct a *model* to which *profiles* are appended. Each profile can be fine-tuned, and fully describes a component of the resulting image. After profiles are appended, the model is evaluated, and the resulting image can be retrieved. `libprofit` cycles through each of the profiles generating individual images for each of them which are optionally convolved and finally added. Hence, it is trivial to describe a one-, two-, or multicomponent system. This external interface, internal organization, and execution of the `libprofit` library are depicted in Fig. 1 using the Unified Modelling Language.

### 2.1 Galaxy profiles

`libprofit` includes a default set of popular profiles: sérsic, Core-sérsic, broken-exponential, Moffat, Ferrer, modified empirical King, PSF, and sky (see Fig. 2). The first six share some key similarities: they can be fully described by an analytic radial profile that is then evaluated in two dimensions over pseudo-elliptical isophotes. They also share similarities in terms of how they must be evaluated in a given pixel: if the gradient of the model image is varying rapidly over the scale of a pixel, then using a simple trapezoidal integration scheme (effectively taking the flux value predicted at the centre and assuming this to reflect the average for the pixel) can be error prone.

For this reason the sérsic, Core-sérsic, broken-exponential, Moffat, Ferrer, and King profiles all share the same core c++ code that evaluates the flux in a pixel and determines whether more accuracy is required. This means new profiles can be defined quite simply within `libprofit`, effectively reducing to the one-dimensional form of the radial profile and a scheme to calculate the total flux within the profile. For sérsic, Moffat, and Ferrer, the total integration is known (or has been calculated by the authors) analytically, so this makes them particularly simple cases.

All eight profiles are discussed in more detail below, with the caveat that the detail of the integration scheme discussed with respect to the sérsic profile also applies to the Core-sérsic, broken-exponential, Moffat, Ferrer, and King.

#### 2.1.1 Sérsic profile

For galaxy fitting purposes the sérsic profile (Sérsic 1963) is a modern-day work-horse, having enough flexibility in profile characteristics to fit a large variety of galaxy types (see e.g. Kelvin et al. 2012). The intensity profile is usually parametrized as

$$I(R) = I_e \exp \left\{ -b_n \left[ \left( \frac{R_m}{R_e} \right)^{1/n} - 1 \right] \right\}, \tag{1}$$

where $I_e$ is the intensity of the profile at $R_e$ (the radius containing half of the flux), $n$ is the sérsic index (which has special cases of normal/Gaussian with $n = 0.5$, exponential with $n = 1$ and de-Vaucouleurs with $n = 4$), and $b_n$ is a derived parameter that ensures the correct integration properties at $R_e$ and is the quantile at which the Gamma probability distribution function integrates to 0.5 given a shape parameter of $2n$. This can be computed directly
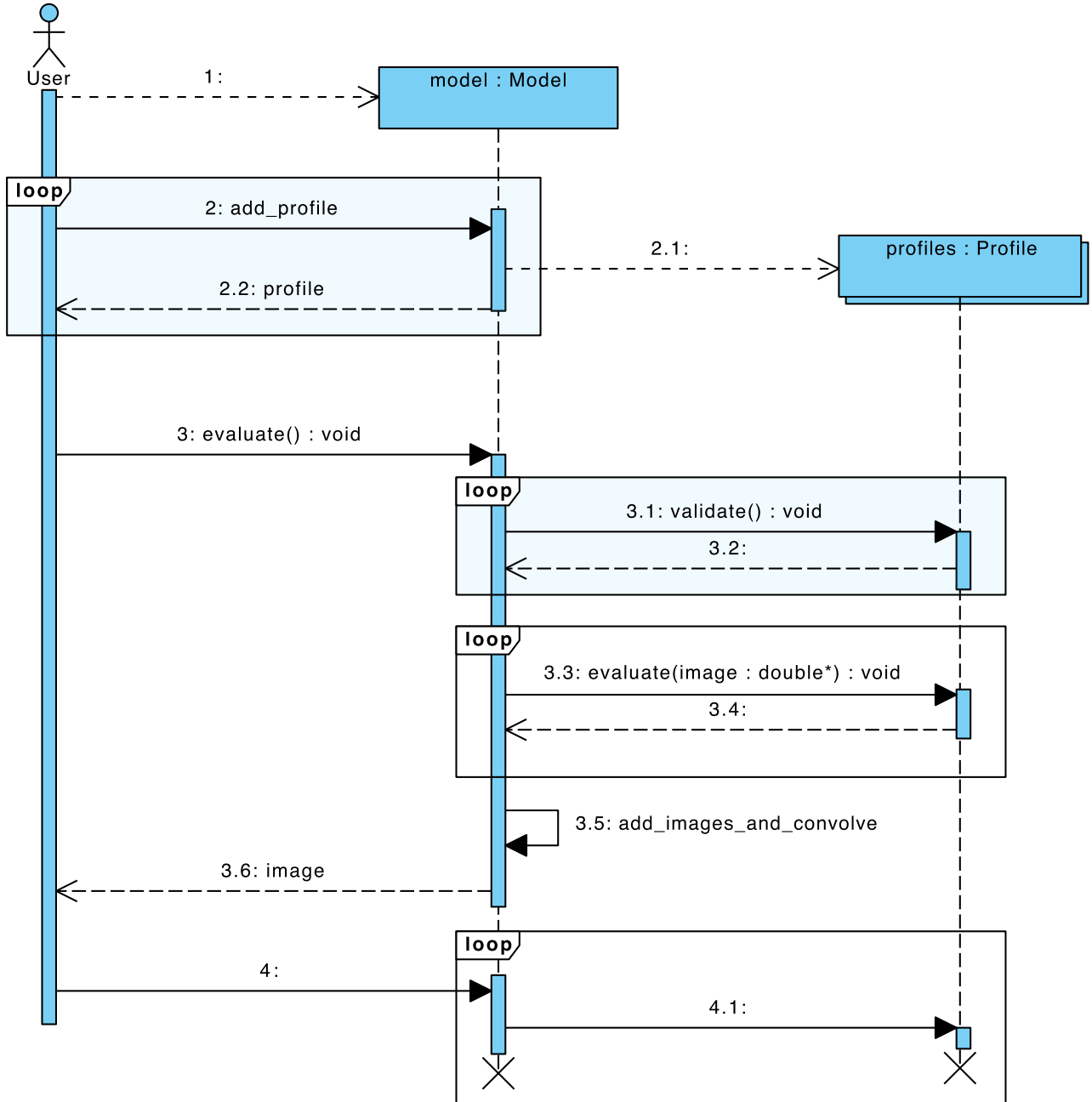
**Figure 1.** Unified Modelling Language sequence diagram showing how to interact with `libprofit` (UML is a standard language commonly used when designing software). Users create a model, append profiles to it, and finally evaluate it.

using high-level interfaces to statical distribution libraries (e.g. in R $b_n = \mathtt{qgamma(0.5, 2*n)}$). $R_m$ is the modified radius where we want to make the evaluation which in one dimension is simply the radius from the profile centre. In two dimensions, $R_m$ has a more complex form:

$$R_m = \sqrt{(x - x_{\mathrm{cen}})^2 + (y - y_{\mathrm{cen}})^2}, \qquad (2)$$

where $x, y$ is the two-dimensional location where we wish to make the evaluation and $x_{\mathrm{cen}}, y_{\mathrm{cen}}$ is the location of the profile centre. When the two-dimensional profile is non-circular, the isophotal contours become elliptical annuli, so calculating $R_m$ becomes a series of computations that rotate and circularize the annuli. When the major-axis angle $\theta$ is defined such that it is $0°$ vertically and

increases positively as the galaxy is rotated counter-clockwise (as per GALFIT) these steps become:

$$R_t = \sqrt{(x - x_{\mathrm{cen}})^2 + (y - y_{\mathrm{cen}})^2}, \qquad (3)$$

$$\theta_t = \arctan\frac{x - x_{\mathrm{cen}}}{y - y_{\mathrm{cen}}} + \theta, \qquad (4)$$

$$R_m = \sqrt{(R_t \sin(\theta_t) A_{\mathrm{rat}})^2 + (R_t \cos(\theta_t))^2}, \qquad (5)$$

where $A_{\mathrm{rat}}$ is the minor-to-major axis ratio (so always a number between 0 and 1, where 0 is an infinitely thin line and 1 is a circle or disc). The final modifier to the standard one-dimensional sérsic profile intensity is to allow for apparent 'boxiness'. Boxiness is often
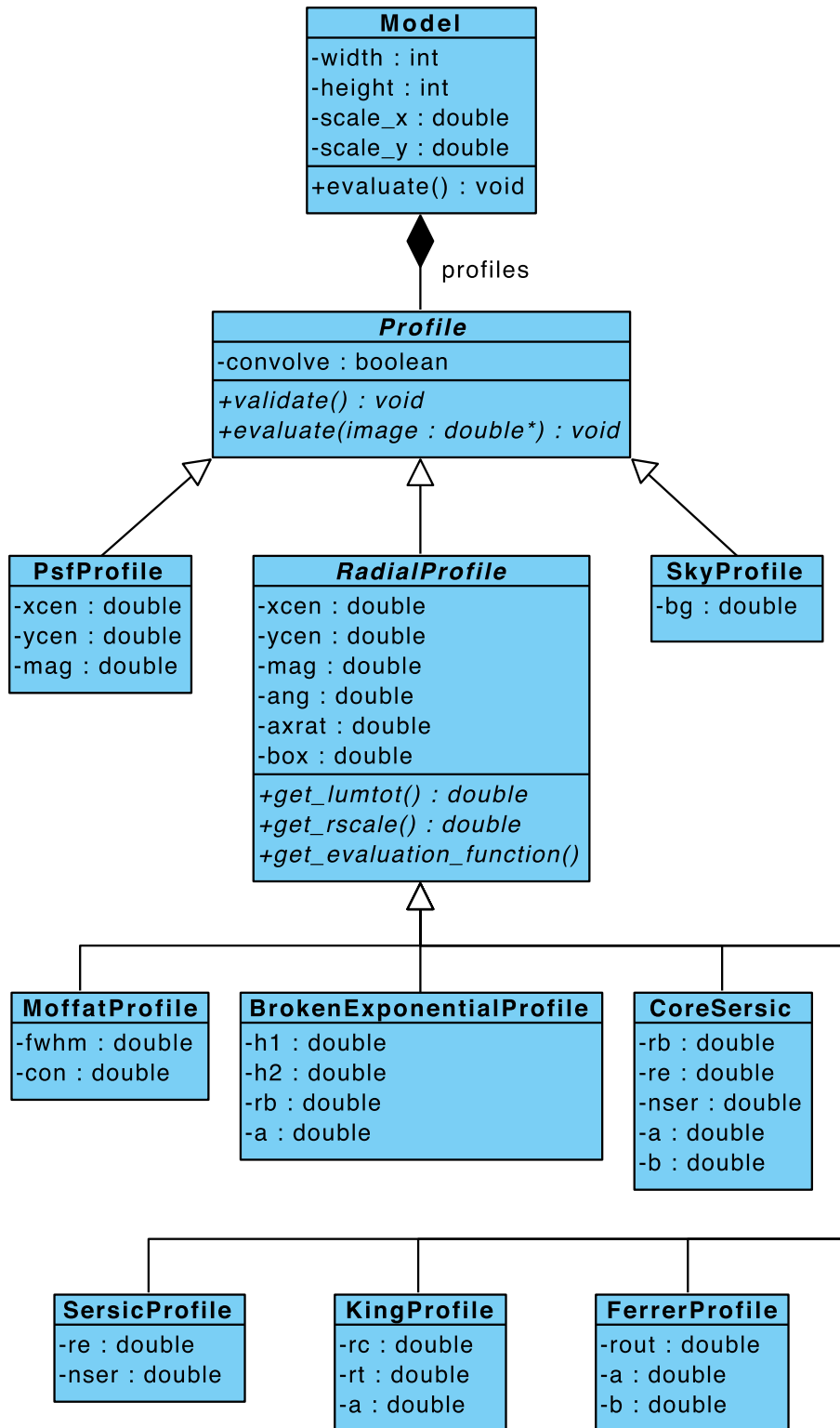
**Figure 2.** UML class diagram showing the eight profile types available in v1.0 of `libprofit` (from left to right, top to bottom): PSF, sky, Moffat, broken-exponential, Core-sérsic , sérsic , King, and Ferrer. The bottom six derive from a common type, the radial profile (centre), a generic form of profile that takes an analytic one-dimensional profile as its input. It is easy to add more profiles of this type if they can be described with a one-dimensional form and the total integral is calculable and finite (even if not analytic). All profile types inherit from a common class and form part of a model.
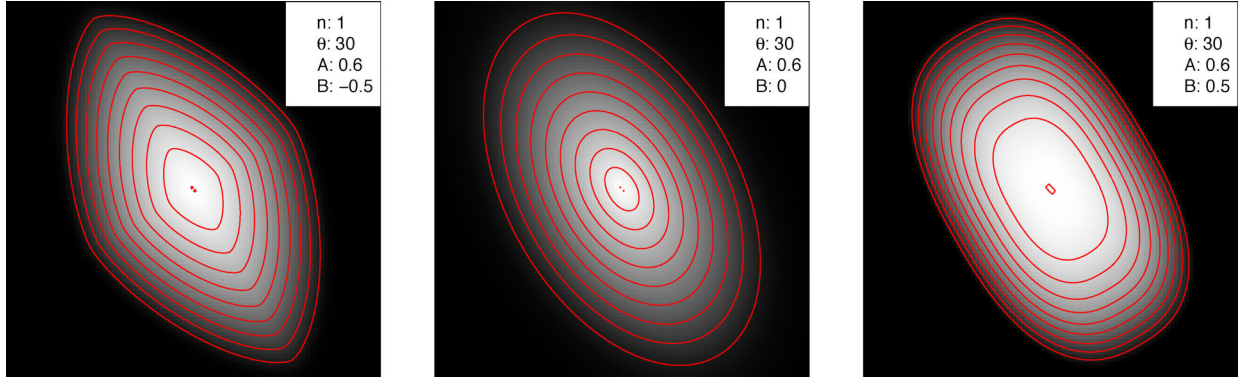
**Figure 3.** Comparison of kite-like negative boxiness (left), normal ellipses with no boxiness (middle), and box-like positive boxiness (right).

used to create the more rectangular visual appearance of galaxy bulges or elliptical galaxies. It effectively modifies the unit circle (or ellipse) away from the standard $L^2$ norm and either towards a diamond/kite appearance ($L^1$ norm) or a boxy appearance ($L^\infty$ norm). This is achieved though a small manipulation of the final $R_m$ calculation above:

$$R_m = \left[ (R_t \sin(\theta_t) A)^{(2+B)} + (R_t \cos(\theta_t))^{(2+B)} \right]^{\frac{1}{2+B}} , \qquad (6)$$

where $B$ is the boxiness which for practical (as opposed to mathematical) purposes is usually allowed to vary between $-1$ and $1$ (thus covering the full range of apparent visual boxiness). Fig. 3 shows the effect of different negative and positive values of boxiness. A consideration for the user is the relative expense of computing the boxiness. Powers that differ from the $L^2$ norm are much more expensive to compute, in particular non-integer values might be a factor of a couple slower when computing a model image. In most standard c math libraries (`libm`), it is faster to make successive calls to `sqrt` and `csqrt` rather than a single call to `pow`. `libprofit` acknowledges this fact and calls the most efficient set of functions as appropriate for certain exponents.

Whilst the above is an accurate description of the computations that need to be made to calculate an arbitrary two-dimensional sérsic profile, they are not optimal. `libprofit` uses rotation matrix arithmetic internally to achieve the above steps efficiently, projecting pixel coordinates on to the major and minor axes of the ellipse. This can be a factor of a few faster than explicitly computing rotations with expensive calls to trigonometric functions.

It is not necessary to parametrize the profile using $I_e$ to define the normalization, as this is rarely intuitive. Instead, the user can specify the total flux contained within a profile in magnitudes ($m$) given a magnitude zero-point ($m_{\rm zero}$), and calculations are made internally to convert this to the correct value of $I_e$. These include appropriate modifications for the axis ratio ($A_{\rm rat}$) and boxiness ($B$):

$$R_{\rm box} = \frac{\pi (2 + B)}{2\beta \left( \frac{1}{2+B}, \frac{1}{2+B} \right)}, \qquad (7)$$

$$L_{\rm tot} = \frac{2\pi A_{\rm rat} R_e^2 n \Gamma(2n) \exp(b_n)}{R_{\rm box} b_n^{2n}}, \qquad (8)$$

$$I_e = \frac{10^{-0.4(m-m_{\rm zero})}}{L_{\rm tot}}, \qquad (9)$$

where $\beta$ is the Beta function defined as $\beta(a, b) = \Gamma(a)\Gamma(b)/\Gamma(a + b)$, and $\Gamma$ is the standard Gamma function. If the profile has no boxiness, then the above is simplified by noting that $R_{\rm box}(B = 0) = 1$ and therefore the denominator term for $L_{\rm tot}$ disappears. To aid flexibility, it is also possible in `libprofit` to

define the profile normalization using the mean surface brightness within $R_e$ ($< \mu >$), where:

$$< \mu > = m + 2.5 \log_{10}(\pi Re^2 A_{\rm rat}) - 2.5 \log_{10}(0.5). \qquad (10)$$

`libprofit` follows the convention that $< \mu >$ is determined for an elliptical radius containing half of the objects flux, i.e. it is not modified for boxiness. This can be useful when parametrizing a galaxy where the galaxy magnitude is entirely unknown since galaxies span a much smaller dynamic range in surface brightness than total magnitude (Driver et al. 2005).

In conclusion, the sérsic profile is specified fully through the provision of eight parameters that are parsed into the `libprofit` image-building routine as a list structure of equal length vectors. Below is an example specification of a two-component sérsic model in the R implementation of PROFIT that directly calls the `libprofit` API to generate model images, where $x_{\rm cen} = $ xcen, $y_{\rm cen} = $ ycen, $m = $ mag, $R_e = $ re, $n = $ nser, $\theta = $ ang, $A_{rat} = $ axrat, and $B = $ box:

```
modellist = list(
  sersic = list(
    xcen = c(50, 50),
    ycen = c(50, 50),
    mag = c(12, 13),
    re = c(14, 5),
    nser = c(1, 8),
    ang = c(46, 80),
    axrat = c(0.4, 0.6),
    box = c(0,-0.3)
  )
)
```

This model is shown in Fig. 4. As expected, the high sérsic index bulge component dominates the flux near the galaxy centre as well as at very large radii, due to the shallow slope of an $n = 8$ sérsic profile. This is visible as side lobes in the surface brightness contours jutting out from the visually dominant disc component running diagonally across the image.

### 2.1.2 Core Sérsic

Closely related to the sérsic profile is the core-sérsic profile introduced by Graham et al. (2003). This was designed to parametrize the strong inner core (i.e. suppression compared to pure sérsic) profile of well-resolved elliptical galaxies. The inner components of such galaxies had previously been parametrized well with the Nuker profile (Lauer et al. 1995), at least in the inner parts of the
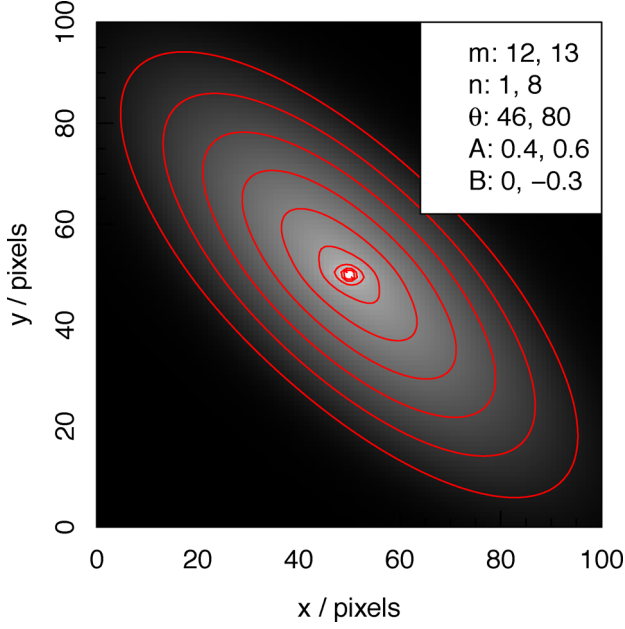
**Figure 4.** Example `libprofit` model image with no convolution by the PSF.

profile. However, the Nuker profile fits poorly at large radii and does not have a well-behaved integral to infinite radius for typical parameters. The core-sérsic addresses these issue by allowing the inner component to behave in a Nuker-like manner, with the outer parts being described by a standard sérsic profile. The intensity profile can be parametrized as

$$I(R_m) = I_r \left[ 1 + \left( \frac{R_m}{R_b} \right)^{-a} \right]^{\frac{b}{a}} \exp\left[ -b_n \left( \frac{R_m^a + R_b^a}{R_e^a} \right)^{\frac{1}{na}} \right], \tag{11}$$

where $I_r$ is the intensity at a parameter-dependent reference point (the centre if $R_b = 0$), $R_b$ is the break radius that controls the point where the outer sérsic-like profile transitions to become an inner Nuker-like profile, $R_e$ is the standard sérsic half light radius (and $b_n$ has the same meaning as for the pure sérsic profile), and $a$ controls the strength of transition (large value is a sharper transition) and $b$ controls the inner power law. $R_m$ is the modified radius where we want to make the evaluation which in one dimension is simply the radius from the profile centre. In two dimensions, $R_m$ has a more complex form where, as with the sérsic profile, we can use this basic form and compute annuli that are modified from pure circular ellipses by introducing geometric transformations that compute an effective $R_m$, as per equation (6).

Accounting for the allowed geometric distortions of axial ratio and boxiness, the value of $I_r$ for a given magnitude can be calculated numerically with

$$L_{\text{tot}} = \frac{2\pi A_{\text{rat}} \int_0^\infty R_m I(R_m; I_r = 1) \mathrm{d}R_m}{R_{\text{box}}}, \tag{12}$$

$$I_r = \frac{10^{-0.4(m - m_{\text{zero}})}}{L_{\text{tot}}}, \tag{13}$$

The integral above has no simple analytic solution, so it is instead calculated numerically inside `libprofit` using either the R or GSL implementation of the QUADPACK library (specifically adap-

tive Gauss–Kronrod quadrature integration). This is a much better solution than attempting to re-normalize the generated image since much of the flux could exist beyond the confines of the image pixels, especially for large values of $n$.

Below is an example of a core-sérsic model in the R implementation of PROFIT that directly calls the `libprofit` API to generate model images, where $x_{\text{cen}} = \text{xcen}$, $y_{\text{cen}} = \text{ycen}$, $m = \text{mag}$, $R_b = \text{rb}$, $R_e = \text{re}$, $n = \text{nser}$, $a = \text{a}$, $b = \text{b}$, $\theta = \text{ang}$, $A_{\text{rat}} = \text{axrat}$, and $B = \text{box}$:

```
modellist = list(
  coresersic = list(
    xcen   = 50,
    ycen   = 50,
    mag = 15,
    rb = 5,
    re = 10,
    nser=4,
    a = 2,
    b=1.3,
    ang   = 30,
    axrat   = 0.4,
    box = 0
  )
)
```

### 2.1.3 Broken exponential profile

The broken-exponential profile is a popular means to parametrize rolling or sharp truncations in the surface profiles of discs which are usually close to exponential in profile until they enter the moderate-to-low surface brightness regime. Erwin, Pohlen & Beckman (2008) introduced a simple modification to the standard exponential profile (which is itself a subset of possible sérsic profiles) in order to capture the range of truncations observed. The intensity profile can be parametrized as

$$S = (1 + e^{-aR_b})^{-\frac{1}{a}\left(\frac{1}{h_1} - \frac{1}{h_2}\right)}, \tag{14}$$

$$I(R_m) = I_0 S e^{\left(\frac{-R_m}{h_1}\right)} \left[ 1 + e^{a(R_m - R_b)} \right]^{\frac{1}{a}\left(\frac{1}{h_1} - \frac{1}{h_2}\right)}, \tag{15}$$

where $I_0$ is the intensity at the centre of the profile, $h_1$ is the scale-length of the inner exponential profile, $h_2$ is the scale-length of the outer exponential profile, $R_b$ is the radius at which the profile transitions from the inner to outer part, and $a$ controls the sharpness of this transition. $R_m$ is the modified radius where we want to make the evaluation which in one dimension is simply the radius from the profile centre. In two dimensions, $R_m$ has a more complex form where, as with the sérsic profile, we can use this basic form and compute annuli that are modified from pure circular ellipses by introducing geometric transformations that compute an effective $R_m$, as per equation (6).

Accounting for the allowed geometric distortions of axial ratio and boxiness, the value of $I_0$ for a given magnitude can be calculated analytically with

$$L_{\text{tot}} = \frac{2\pi A_{\text{rat}} \int_0^{R_t} R_m I(R_m; I_0 = 1) \mathrm{d}R_m}{R_{\text{box}}}, \tag{16}$$

$$I_0 = \frac{10^{-0.4(m - m_{\text{zero}})}}{L_{\text{tot}}}, \tag{17}$$

where $m$, $m_{\text{zero}}$, and $R_{\text{box}}$ have the same meanings as per equation (7).

An example specification of a broken exponential model in the R implementation of PROFIT that directly calls the `libprofit` API to generate model images, where $x_{cen} =$ xcen, $y_{cen} =$ ycen, $m =$ mag, $h_1 =$ h1, $h_2 =$ h2, $R_b =$ rb, $a =$ a, $\theta =$ ang, $A_{rat} =$ axrat, and $B =$ box, is

```
modellist = list(
  brokenexp = list(
    xcen   = 40,
    ycen   = 60,
    h1 = 10,
    h2 = 5,
    rb = 8,
    ang = 30,
    a = 0.2,
    axrat = 0.6,
    box = -0.1
  )
)
```

### 2.1.4 Moffat profile

The Moffat function was designed as a means to parametrize a telescope PSF (Moffat 1969), and is often used on occasions when the PSF is predicable and when you might not have many point sources within the field of view to estimate it empirically (e.g. for space telescopes with small fields of view like the *Hubble Space Telescope*). Whilst this was the purpose behind its design, in practice in modern applications, it is rare to see analytic approximations of the PSF (such as the Moffat) used, the modern preference being for empirically derived estimates. Regardless, for reasons of flexibility, it is included as a profile option in `libprofit`. The Moffat is a re-parametrization of the bivariate Student-T distribution, meaning it creates a normal-like core with Lorentzian-like wings. The intensity profile can be parametrized as

$$I(R_m) = I_0 \left[ 1 + \left( \frac{R_m}{R_d} \right)^2 \right]^{-c}, \tag{18}$$

where

$$R_d = \frac{\text{FWHM}}{2\sqrt{2^{\frac{1}{c}} - 1}}, \tag{19}$$

where $I_0$ is the intensity at the centre of the profile, $c$ is the profile concentration ($c = 1$ is pure Lorentzian, $c = \infty$ is pure normal), and FWHM is the full-width half max of the profile across the major-axis of the intensity profile. $R_m$ is the modified radius where we want to make the evaluation which in one dimension is simply the radius from the profile centre. In two dimensions, $R_m$ has a more complex form where, as with the sérsic profile, we can use this basic form and compute annuli that are modified from pure circular ellipses by introducing geometric transformations that compute an effective $R_m$, as per equation (6). Thus in `libprofit`, the Moffat function need not be circular, which is often the case for the PSF of a wide-field telescope where corner distortions tend to be radial or tangentially aligned with respect to the field centre.

Accounting for the allowed geometric distortions of axial ratio and boxiness, the value of $I_0$ for a given magnitude can be calculated analytically with

$$L_{tot} = \frac{\pi A_{rat} R_d^{\;2}}{(c-1) R_{box}}, \tag{20}$$

$$I_0 = \frac{10^{-0.4(m - m_{zero})}}{L_{tot}}, \tag{21}$$

where $m$, $m_{zero}$, and $R_{box}$ have the same meanings as per equation (7).

An example specification of three Moffat models in the R implementation of PROFIT that directly calls the `libprofit` API to generate model images, where $x_{cen} =$ xcen, $y_{cen} =$ ycen, $m =$ mag, FWHM = fwhm, $c =$ con, $\theta =$ ang, $A_{rat} =$ axrat, and $B =$ box, is

```
modellist = list(
  moffat = list(
    xcen   = c(34,10,85),
    ycen   = c(74,64,13),
    mag  = c(10,13,16),
    fwhm   = c(3, 3, 3),
    con  = c(5, 5, 5),
    ang  = c(45, 45, 45),
    axrat  = c(0.95, 0.95, 0.95),
    box = c(0.1, 0.1, 0.1)
  )
)
```

Since the Moffat profile is almost exclusively used to model the image PSF, `libprofit` is hard-coded to never convolve the profile with another PSF, even if this is provided. This is to help eliminate potential user error.

### 2.1.5 Modified Ferrer/s profile

The modified Ferrer profile (a type of projected potential derived from Ferrers' functions, as discussed in Laurikainen et al. 2005) is a useful parametrization of galaxy features that have very strong drop-offs, most typically bar structures. Whilst it should formally be called the Ferrers profile, it is very common for it to be written as Ferrer (the assumption is that authors have erroneously attributed the final 's' as possessive, but the name derives from Norman Macleod Ferrers). For convenience, the profile can be named as Ferrer or Ferrers in `libprofit` and in higher level interfaces. The intensity can be parametrized as

$$I(R_m) = I_0 \left[ 1 - \left( \frac{R_m}{R_{out}} \right)^{(2-b)} \right]^a, \tag{22}$$

where $I_0$ is the intensity at the centre of the profile, $R_{out}$ is the outer truncation radius (the profile is 0 beyond this radius), $a$ controls the global power-law slope to the profile centre, and $b$ controls the strength of truncation as $R_M$ approaches $R_{out}$. $R_m$ is the modified radius where we want to make the evaluation, which in one dimension is simply the radius from the profile centre. In two dimensions, $R_m$ has a more complex form where, as with the sérsic profile, we can use this basic form and compute annuli that are modified from pure circular ellipses by introducing geometric transformations that compute an effective $R_m$, as per equation (6). For the Ferrer profile, it is common to introduce some positive boxiness to reflect the visually rectangular isophotes of galaxy bars.

Accounting for the allowed geometric distortions of axial ratio and boxiness, the value of $I_0$ for a given magnitude can be calculated analytically with

$$L_{tot} = \frac{\pi A_{rat} R_{out}^{\;2} a \beta \left( a, 1 + \frac{2}{2-b} \right)}{R_{box}}, \tag{23}$$

$$I_0 = \frac{10^{-0.4(m - m_{zero})}}{L_{tot}}, \tag{24}$$

Below is a sample specification of a Ferrer model in the R implementation of PROFIT that directly calls the `libprofit` API to generate model images, where $x_{cen}$ = xcen, $y_{cen}$ = ycen, $m$ = mag, $R_{out}$ = rout, $a$ = a, $b$ = b, $\theta$ = ang, $A_{rat}$ = axrat, and $B$ = box:

```
modellist = list(
 ferrer = list(
   xcen   = 50,
   ycen   = 50,
   mag = 15,
   rout = 20,
   a = 0.5,
   b=0.7,
   ang   = 30,
   axrat   = 0.3,
   box = 0.6
 )
)
```

### 2.1.6 Modified empirical King profile

The modified King profile in `libprofit` shares its origins with the empirically motivated two-dimensional profile presented in King (1962), and remains a popular function for parametrizing globular cluster light profiles. It should be emphasized that it is *not* the same as the equally popular profile presented in King (1966), the latter being parametrized in three dimensions, theoretical in origin, and non-trivial to project. `libprofit` uses the same basic modification to the King (1962) as used in GALFIT, where the intensity can be parametrized as

$$I(R_m) = I_r \left\{ \frac{1}{\left[1 + \left(\frac{R_m}{R_c}\right)^2\right]^{\frac{1}{a}}} - \frac{1}{\left[1 + \left(\frac{R_t}{R_c}\right)^2\right]^{\frac{1}{a}}} \right\}^a , \qquad (25)$$

where $I_r$ is the intensity at a parameter-dependent reference point (the centre if $R_t = \infty$), $R_c$ is the core radius, $R_t$ is the outer truncation radius (the profile is 0 beyond this radius), and $a$ controls the global power law. When $a = 2$, this parametrization is identical to the one presented originally in King (1962). $R_m$ is the modified radius where we want to make the evaluation, which in one dimension is simply the radius from the profile centre. In two dimensions, $R_m$ has a more complex form where, as with the SÉRSIC profile, we can use this basic form and compute annuli that are modified from pure circular ellipses by introducing geometric transformations that compute an effective $R_m$, as per equation (6).

Accounting for the allowed geometric distortions of axial ratio and boxiness, the value of $I_0$ for a given magnitude can be calculated analytically with

$$L_{tot} = \frac{2\pi A_{rat} \int_0^{R_t} R_m I(R_m; I_r = 1) dR_m}{R_{box}}, \qquad (26)$$

$$I_r = \frac{10^{-0.4(m-m_{zero})}}{L_{tot}}, \qquad (27)$$

An example specification of a modified King profile in the R implementation of PROFIT that directly calls the `libprofit` API to generate model images, where $x_{cen}$ = xcen, $y_{cen}$ = ycen,

$m$ = mag, $R_c$ = rc, $R_t$ = rt $a$ = a, $\theta$ = ang, $A_{rat}$ = axrat, and $B$ = box, is

```
modellist = list(
 king = list(
   xcen   = 50,
   ycen   = 50,
   mag = 15,
   rc = 5,
   rt = 30,
   a = 2,
   ang   = 30,
   axrat   = 0.8,
   box = 0
 )
)
```

### 2.1.7 PSF profile

The PSF profile for an unresolved point source (referred to as a `pointsource` within `libprofit` and PROFIT to distinguish it from the PSF) is a simple prescription requiring the $x_{cen}, y_{cen}$ location of the desired point source, the total flux contained in magnitudes $m$ and either an empirical PSF image or analytic PSF model. In the former case, the PSF image is linearly interpolated on to the specified location in the image (thus fractional pixel values can be provided) and is renormalized to ensure the flux contained is correct given the magnitude zero-point ($m_{zero}$); otherwise, the point source is accurately integrated with the specified PSF model using an analytic profile (e.g. the Moffat, as discussed in Section 2.1.4). An example of a model containing three point sources in the R implementation of PROFIT, where $x_{cen}$ = xcen, $y_{cen}$ = ycen, and $m$ = mag, is

```
modellist = list(
 pointsource = list(
   xcen   = c(34,10,85),
   ycen   = c(74,64,13),
   mag = c(10,13,16)
 )
)
```

Numerical integration of an analytic PSF is considerably more accurate than interpolation of an empirical PSF image, so we generally recommend using the former approach if it is at all possible. As mentioned in Section 2.1.4, it is more typical to see astronomers using empirical PSFs. Since the intrinsic PSF is unknown, using such a description will result in a simple linear interpolation to fractional pixel locations. To accommodate this fact, `libprofit` and PROFIT allow for the specification of an oversampled empirical PSF. If PSF modelling is required, this offers both a pragmatic and accurate solution.

### 2.1.8 Sky profile

The simplest profile of all is the sky. In `libprofit`, this is specified as the flux per pixel without any adjustment made for the magnitude zero-point ($m_{zero}$), i.e. it is the flat sky level pedestal directly measured from the image in its native flux units (any subsequent conversions must be made explicitly by the user). An example

sky model in the R implementation of PROFIT requires only the background level bg:

```
modellist = list(
  sky = list(
    bg = 3e-12
  )
)
```

Currently, only flat sky profiles are available in `libprofit` and PROFIT, with the assumption that in typical PROFIT use cases the sky is a pedestal term, with no gradient of curvature. Depending on community feedback additional complexity for the sky profile could be added in future. However, it is clear that the majority of modern survey image processing involves an explicit sky-subtraction stage that reduces the data to a pedestal-only sky profile (e.g. see Kuijken et al. 2015). Like any other profile type, it is possible to fit for the sky profile background within `libprofit`; however, in many use cases, the background will be known a priori and should not be fit. A common issue with the leaving the sky as a fitted parameter is how degenerately it behaves with sérsic-index outer profiles, which also appear close to flat at many multiples of $R_e$. To guard against this effect, it is wise to fit using pixels that are dominated by galaxy flux.

### 2.1.9 A combined model

Combining models in `libprofit` is simple given the nested list nature of object specification. A structure containing the sérsic, Ferrer bar, point-source, and sky models is given by

```
modellist = list(
  sersic = list(
    xcen  = c(50, 50),
    ycen  = c(50, 50),
    mag = c(12, 13),
    re  = c(14, 5),
    nser  = c(1, 8),
    ang  = c(46, 80),
    axrat = c(0.4, 0.6),
    box = c(0,-0.3)
  ),
  ferrer = list(
    xcen = 50,
    ycen = 50,
    mag = 14,
    rout = 12,
    a = 0.3,
    b = 1.5,
    ang = 130,
    axrat = 0.2,
    box = 0.5
  ),
  pointsource = list(
    xcen  = c(34,10,85),
    ycen  = c(74,64,13),
    mag = c(18,15,16)
  ),
  sky = list(
    bg = 3e-12
  )
)
```
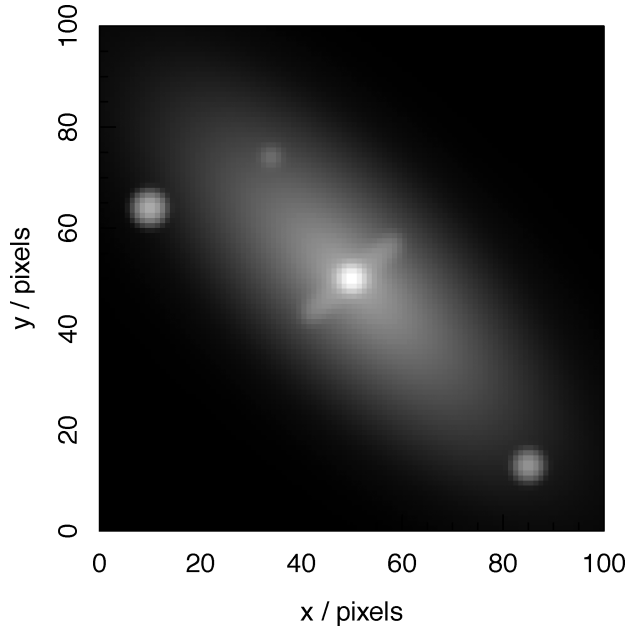


**Figure 5.** Example `libprofit` model image with convolution by the PSF, which is an idealized Gaussian PSF with full-width half max (FWHM) of 1.3 pixels. For clarity, no surface brightness contours are overdrawn. This image has the same two-component sérsic model as shown in Fig. 4, but with the addition of a Ferrer bar profile, three PSFs, and a sky background.

This is the list structure used in R PROFIT to specify models, and in principle an unlimited number of profiles can be provided to create complex model images. In practice, for galaxy image fitting, simple models are made of the region of direct interest. This structure is also the mechanism for providing initial conditions for galaxy image fitting. This structure can be easily manipulated from outside of the R environment, or directly within R.

Fig. 5 shows the image generated by the example model above. This is very similar to Fig. 4 but with the addition of a Ferrer bar profile, three bright point sources, and a sky background (which is not visible due to the image scaling). More details on how `libprofit` generates these images is provided in the following section.

### 2.2 Accurate and fast pixel integration

The main focus of development effort in `libprofit` was optimizing model image generation for speed *and* accuracy, particularly for the sérsic profile (since this is by some margin the most popular general purpose profile for galaxy modelling). The first requirement to achieve rapid image generation was that the core code should be written in a fast low-level language. C++ was chosen for these purposes, with the idea that `libprofit` should provide a generic low-level library that higher level languages (e.g. R and PYTHON) can access easily.

With C++ chosen as the implementation language, the main considerations left were how to best approach the problem of model image generation algorithmically. Two-dimensional modelling routines are vague in describing how they achieve pixel integration, with most published references stating that inner pixels are 'oversampled' in order to achieve reasonable accuracy in the pixel flux (note this is different to the type of oversampling required to ensure accurate image convolution, as discussed in Section 2.3). Depending on the profile being created, the degree of oversampling can vary by many orders of magnitude. Getting accurate solutions for the flux in a pixel containing the peak of a steep (high *n*) sérsic

profile is clearly a very different prospect to estimating the flux of a pixel a large distance from the centre, where the profile is very flat. To combat this problem, `libprofit` uses a unified oversampling scheme (inspired by adaptive quadrature) for all profiles that have an analytic radial intensity profile – initially the sérsic, Core-sérsic, broken-exponential, Moffat, Ferrer, and King profiles discussed above. Here, we concentrate on the evaluation of the sérsic profile, but the basic principles hold true for all radially varying profiles. In `libprofit`, all such profiles inherit from a generic class of radial type profiles (see Fig. 2), where the pixel integration scheme is shared in a common method. The profiles are specified by their radial intensity and total luminosity (as per Section 2.1). This minimizes code duplication and simplifies the addition of new profiles to `libprofit`, where the user only needs to be able to express the one-dimensional form of the profile shape and a method to integrate it to get the total luminosity (which can simply be a numerical integration if there is no convenient analytic solution).

The qualitative version of the pixel integration scheme is that an oversampling resolution, a recursion depth, and an accuracy level are specified for a given pixel. The grid is initially oversampled by the integer factor provided, for each sub-pixel the gradient along the minor axis (this being the steepest descent) is computed and when it is over a tolerance specified by the accuracy value this pixel is selected for recursive oversampling. In this situation, the sub-pixel becomes redefined as the parent pixel and the scheme above is repeated. Fig. 6 is a simple schematic of the central $3 \times 3$ pixels being recursively oversampled by a factor $5 \times 5$ to achieve an accurate numerical integration in the regime where the profile is rising steeply in the core of the galaxy.

The default allowed maximum depth for sub-pixel recursion is 2, but it can be user specified. The default oversampling factor is 8, meaning at most sub-regions of pixels would be oversampled by a factor $(8^2)^2 = 8^4$. A special case is made for the central pixel in the sérsic profile (containing the flux peak). To ensure accurate results even for extreme examples of very steep profiles of barely resolved galaxies, the maximum recursion depth is increased to 10, giving potential oversampling of $(8^{10})^2 \sim 10^9$. In extreme cases, where deep recursion is triggered on the very central pixel of each layer, the practical worst-case scenario is that $(8^2) \times 10 = 640$ calculations are actually required because only the sub-pixels containing the profile peak require deeper recursion.

This combination of accuracy and speed means `libprofit` performs well in model image generation bench-marking whilst maintaining user-definable accuracy with sensible defaults. Detailed comparisons were made with reference to GALFIT, where the target was to achieve similar or better pixel-level accuracy whilst ensuring the model image computation time was no longer for typical values of the sérsic index (*n*). For these comparisons, a modified build of GALFIT was used that did not read or write to disk and instead took arguments and produced outputs in memory (ensuring this relatively large I/O bottleneck was not a factor).

The exact pixel fluxes were computed using the R2CUBA R package that is an interface to the CUBA C++ library that offers deterministic algorithms for multidimensional numerical integration. The algorithm used was CUHRE, which offers a user-definable level of integration accuracy. For these tests, it was set to be better than 0.1 per cent pixel accuracy (in practice achieving an order of magnitude better accuracy in most pixels), so at most this amount of inaccuracy may come from the reference *exact* pixel fluxes. It is worth highlighting why this approach cannot be used in general: to create a typical $20 \times 20$ pixel image (a very small image by galaxy fitting standards) with this level of accuracy takes $\sim 3$ s (and much

longer for some profiles). In comparison, `libprofit` and GALFIT can both create similar image in less than 0.01 s, i.e. they are both at least 300 times faster than using a multidimensional integration library, even a fast one written in C++. It is not clear why using this specialized library is so slow, but it ruled out using it within `libprofit` for general application.

Fig. 7 compares the flux-weighted pixel error of `libprofit` and GALFIT as a function of sérsic index for sérsic profiles (again, sérsic profiles have been highlighted due to their popularity in the galaxy profiling community) with $R_e = 2$pixels, $\theta = 60°$, and $A/B = 0.3$. Two things are immediately clear: `libprofit` has a lower flux-weighted error per pixel (typically 0.1 per cent), and this error does not vary significantly with sérsic index. GALFIT achieves a flux-weighted error per pixel of 0.3 per cent for low values of the sérsic index, but this inflates to >1 per cent for larger values of *n*.

The image generation is shown in more detail for these example profiles in Fig. 8. This shows the actual error as a function of pixel location on the two-dimensional image for the target model for both `libprofit` and GALFIT. Fig. 9 shows the one-dimensional pixel error across the central portion of pixels that incorporate the peak pixel (which is usually the hardest to integrate accurately for the sérsic profile). There are specific regions of pixels where GALFIT has smaller errors, but the general trend is that `libprofit` is more accurate over a large range of pixels, and never experiences large integration error in pixels that contain significant fractions of the profile flux. In the case of both `libprofit` and GALFIT, outer pixels might often have large relative error, but in these cases the pixels contain almost no flux and do not drive the overall model image to a large flux-weighted error.

As established in the design goals of `libprofit`, generating images accurately is important, but there is a trade-off to be made with the required computation time. It is unavoidable that some pixels will be harder to evaluate due to the rapid change in flux within a pixel (especially true for the central pixel containing the peak of the sérsic profile flux), and more computing time will be needed for correctly evaluating the pixel flux. `libprofit` uses adaptive recursion to adjust the integration resources as required in order to maintain a relatively constant level of pixel integration accuracy. This is evident from Fig. 10, which compares the image generation times for `libprofit` and GALFIT for the same sérsic profiles as presented in Fig. 7. It is clear that `libprofit` varies the integration time (which is akin to more resources spent evaluating *hard* pixels) more in order to maintain relatively constant flux-weighted errors (see Fig. 7), whereas GALFIT maintains nearly constant integration times.

The computation times of `libprofit` and GALFIT for sérsic profiles are investigated in more detail in Figs 11 and 12. For all these tests, the same MacBook Pro running El-Capitan with 2.6 GHz i7 processors and 16 GB of RAM was used. In Figs 11 and 12, we see that `libprofit` is factors of a few faster compared to GALFIT for most combinations of model parameters. In Fig. 12, we see there is weak dependence of computation time on $R_e$ and *n*, and a strong dependence on axial ratio (in all three cases larger means longer computation time for `libprofit`). For reasons not immediately clear, GALFIT actually takes longer to compute compact models – perhaps because its integration scheme is not sufficiently adaptive.

In general, we find a wide range of potential computation times, spanning a factor of $\sim 10$ for `libprofit` versus $\sim 3$ for GALFIT even though the target image is $200 \times 200$ pixels in all cases. This makes it hard to predict in general how long a particular fit might take given the image size. In a pathological situation of the target galaxy having large $R_e$, large *n*, and being circular in projection,
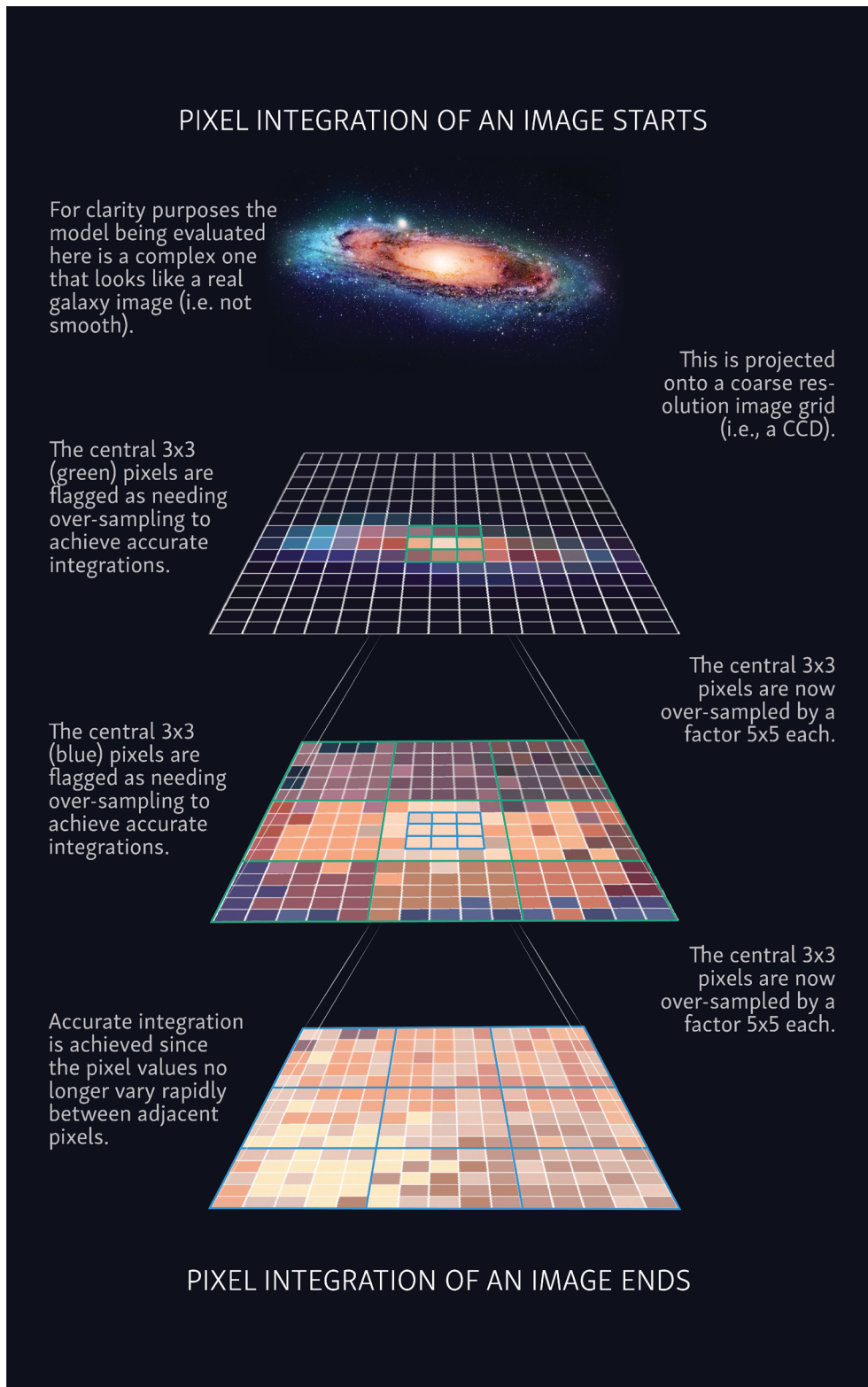
**Figure 6.** Simple schematic of PROFIT recursively upsampling the inner most pixel in order to achieve a desired level of per-pixel integration accuracy. In this example, the inner 3 × 3 pixels are up-sampled by 5 × 5 each. Of this new upsampled grid the inner 3 × 3 pixels are up-sampled by 5 × 5 again, after which the recursion stops. In PROFIT, this assessment is made by the rate-of-change of flux between pixels, i.e. if the change is rapid then pixels are identified as requiring upsampling.
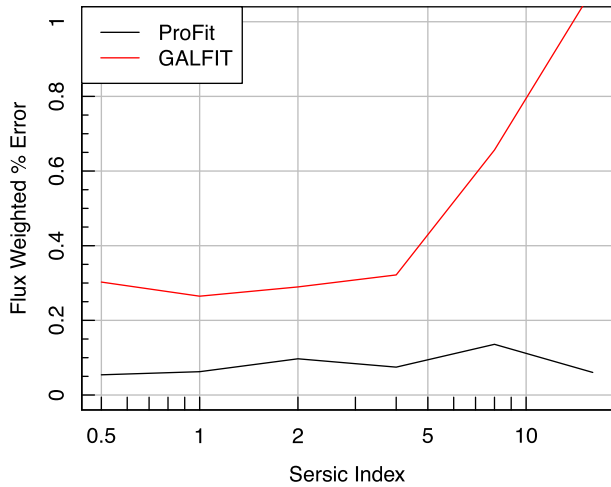
**Figure 7.** `libprofit` versus GALFIT flux-weighted model image error for sérsic indices 0.5–16 for $R_e = 2$ pixels, $\theta = 60°$, and $A/B = 0.3$ in a $20 \times 20$ pixel image. A subset of these results (for $n = 1, 4, 8, 16$) are shown in detail in Fig. 8.

the computation could be a factor 10 longer than for the case of a compact galaxy with low *n* and fairly elongated projection.

It is possible to configure `libprofit` to exceed ∼0.1 per cent flux-weighted image accuracy. This default was deemed appropriate for typical (and certainly our) uses of `libprofit` since it equates to 0.001 mag accuracy which in practice far exceeds the systematic and random measurement uncertainties for even relatively high fidelity photometric images. It also allows for rapid image fitting since small gains in accuracy begin to take factors longer to compute. However, a combination of increasing the oversampling and recursion depth allows for arbitrarily high levels of integration accuracy should this be required by the user. By including a simple interface to the Cuba library in the ʀ PROFIT package for all six radial profile types, it is possible to tune these parameters in order to achieve the desired level of accuracy and speed.

### 2.3 Image convolution

PROFIT supports both brute-force convolution and FFT-based convolution using the FFTW library. In the ʀ implementation, both methods are benchmarked and the optimal method for the given image and PSF size is chosen. Brute-force convolution is often significantly faster than FFT-based methods but scales directly with image *and* PSF size, being $\mathcal{O}(n_{\text{image}} n_{\text{PSF}})$, whereas FFT convolution is $\mathcal{O}(4 n_{\text{image}} \log 4 n_{\text{image}})$, accounting for necessary zero-padding. Testing shows that both methods give nearly identical answers, with fractional differences of less than $10^{-12}$.

One important consideration is that the accuracy of convolution depends on the resolution of the PSF relative to the pixel scale. A poorly resolved PSF with an FWHM ≤3 pixels may give flux errors on the order of a few per cent for the innermost pixels. PROFIT allows for model and PSF oversampling for more accurate convolution, whereby the model pixel grid is subdivided by an integer factor (preferably odd). This is independent of the oversampling for the purposes of accurate model convolution, and requires either an analytic PSF or prior interpolation of the empirical PSF. Interpolating a noisy empirical PSF offers a limited gain in accuracy, so for this reason (and as mentioned in Section 2.1.7), analytic PSFs are generally preferred. When this process is attempted empirically, it is usually via stacking or drizzling of the data (e.g. with the use

of a program like MultiDrizzle[1]). Accounting for pixel covariance when drizzling is a complex problem, and best handled by fitting the native resolution images simultaneously rather than attempting to combine them in an information-lossy process. Simultaneous image fitting of multi-exposure data and covariance likelihoods are high on the list of future extensions to PROFIT, but are not present in v1.0 (although experimental covariance likelihoods are already being tested).

## 3 IMAGE FITTING WITH PROFIT

Having demonstrated that `libprofit` integrates and convolves analytic profiles quickly and accurately, the remaining ingredients for model fitting are the likelihood function and optimization and/or likelihood sampling method. This logic is implemented in the PROFIT ʀ package. ʀ is becoming increasingly widely used in astronomy and other fields of advanced data analysis, and since it features a large number of statistics packages and optimization methods (maximum-likelihood and Bayesian), it is ideally suited for the purpose of Bayesian galaxy image fitting.

Having easy access to a wealth of optimizers was a key design goal of PROFIT. A large subdivision of statistical data science and computer science has invested huge efforts into describing and implementing a multitude of optimizers that have a range of strengths and weaknesses. Depending on the specific problem at hand, one optimizer might be preferred over another for reasons that are very hard for us to intuit. Since this is such a key component of galaxy modelling, where a typical use case might involve fitting dozens of partly degenerate parameters, giving the user flexibility to experiment with optimization engines was deemed critical.

In practice, an initial global parameter search might be made using a fast but biased downhill gradient optimizer, or one which is known to get easily trapped in local minima. From this point, a more expensive but robust Markov Chain Mote Carlo (MCMC) optimizer could be used to accurately refine the parameter posteriors (this is the typical mode of operation recommended by the authors and described in detailed vignettes included with the ʀ package version of PROFIT). In the ʀ implementation alone, the user has easy plug-and-play access to hundreds of optimizers accessible though the Central R Archive Network (CRAN) with hundreds more served from online repositories (e.g. GitHub). We do not wish to be overly prescriptive in how samplers are used to model data, but some typical applications will be discussed in the following sections.

### 3.1 Required observational data

A number of inputs are required to meaningfully fit a profile to a target galaxy observation. For good-quality data, where the whole image is to be used for fitting, the bare minimum is strictly an image where pixels contain integer photoelectron counts and an analytical description of the PSF.

In practice, a more complete set of inputs can include the flux image (*D*, in whatever linear units are appropriate), a segmentation map (that is pixel matched to *D*, where distinct integer values represent different detected structures in the image as per SEXTRACTOR), a binary mask (that is pixel matched to *D*, where TRUE means mask out and ignore for analysis, and FALSE means do not mask), and the $\sigma$ image ($\sigma$ that is pixel matched to *D*, where values represent normal errors in the same units as for *D*).

---

[1] http://stsdas.stsci.edu/multidrizzle/
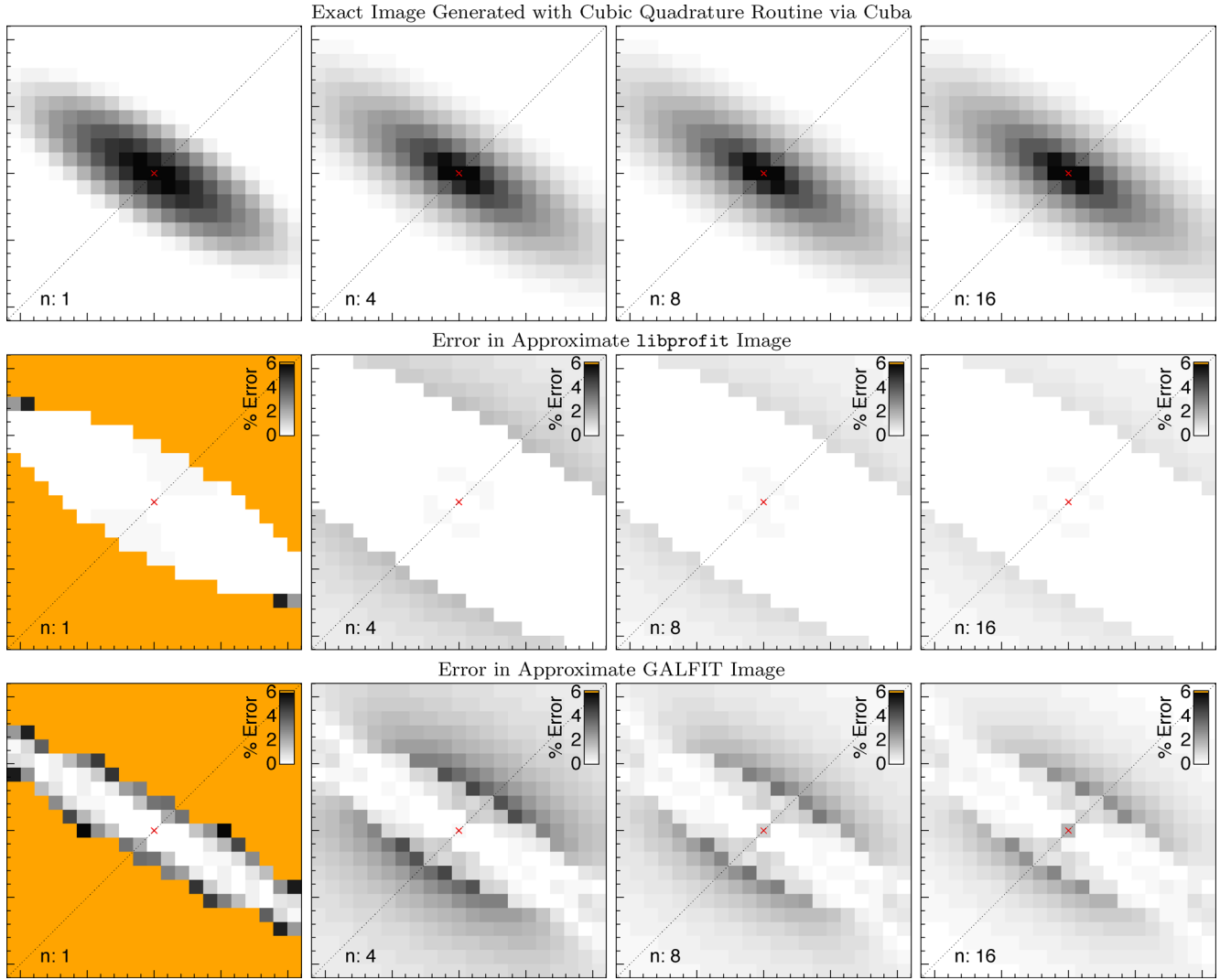
**Figure 8.** A detailed view of model error during image generation in `libprofit` versus GALFIT for sérsic indices 1, 2, 4, and 16 (as labelled), $R_e = 2$ pixels, $\theta = 60°$, and $A/B = 0.3$ in a $20 \times 20$ pixel image. The examples shown here are a subset of the flux-weighted errors shown in Fig. 7. The top row shows the exact Cubic Quadrature derived model being generated, using a sinh scaling for the grey-scale. The second row down shows the `libprofit` error residuals at each pixel location. The third row down shows the GALFIT error residuals at each pixel location. Orange highlights regions that have errors greater than the fixed grey-scale limits, i.e. above 6 per cent. The pixels along the diagonal dotted line are shown in Fig. 9.

The PSF can be specified in two main ways: either via an analytical description using any of the available radial profiles (in practice, the Moffat function is a good profile for this task), or with an image of the PSF that is on the same pixel scale as $D$. PROFIT has a high-level setup function (`profitSetupData`) that can take these basic inputs and create an internal object that is appropriate for fitting a PROFIT model.

### 3.2 Image likelihood

There is a wealth of routes to compute meaningful image versus model likelihoods, depending on the data regime. If many (i.e. hundreds of) photoelectrons are registered in pixels for the observed galaxy then the data are likely to be operating in the normal statistics regime, where PROFIT likelihoods can be computed with

$$x_{i,j} = \frac{(D_{i,j} - M_{i,j})^2}{2\sigma_{i,j}^2}, \qquad (28)$$

$$\ln \mathcal{L} = \frac{\ln 2\pi}{2} - \sum_{i=1}^{N_i} \sum_{j=1}^{N_j} x_{i,j} - \ln \sigma_{i,j} \qquad (29)$$

Here, $D_{i,j}$ represents the observed pixel data at pixel $i, j$, $M_{i,j}$ represents a computed PROFIT model at pixel $i, j$, and $\sigma_{i,j}$ represents the estimated normal uncertainty at pixel $i, j$.

An alternative view is that cumulative sum of squared residuals should follow a $\chi^2$ distribution, with a PROFIT likelihood computed by

$$\chi^2 = \sum_{i=1}^{N_i} \sum_{j=1}^{N_j} \frac{(D_{i,j} - M_{i,j})^2}{\sigma_{i,j}^2}, \qquad (30)$$

$$\ln \mathcal{L} = -\frac{k}{2} \ln 2 - \ln \Gamma(k/2) + \left(\frac{k}{2} - 1\right) \ln \chi^2 - \frac{\chi^2}{2}. \qquad (31)$$

Here, $k$ represents the degrees of freedom of the $\chi^2$ distribution. For fitting of this type, this will usually be determined by the number of pixels containing galaxy flux (here, $N_i N_j$) minus the number of

Error Across Central Diagonal Pixels
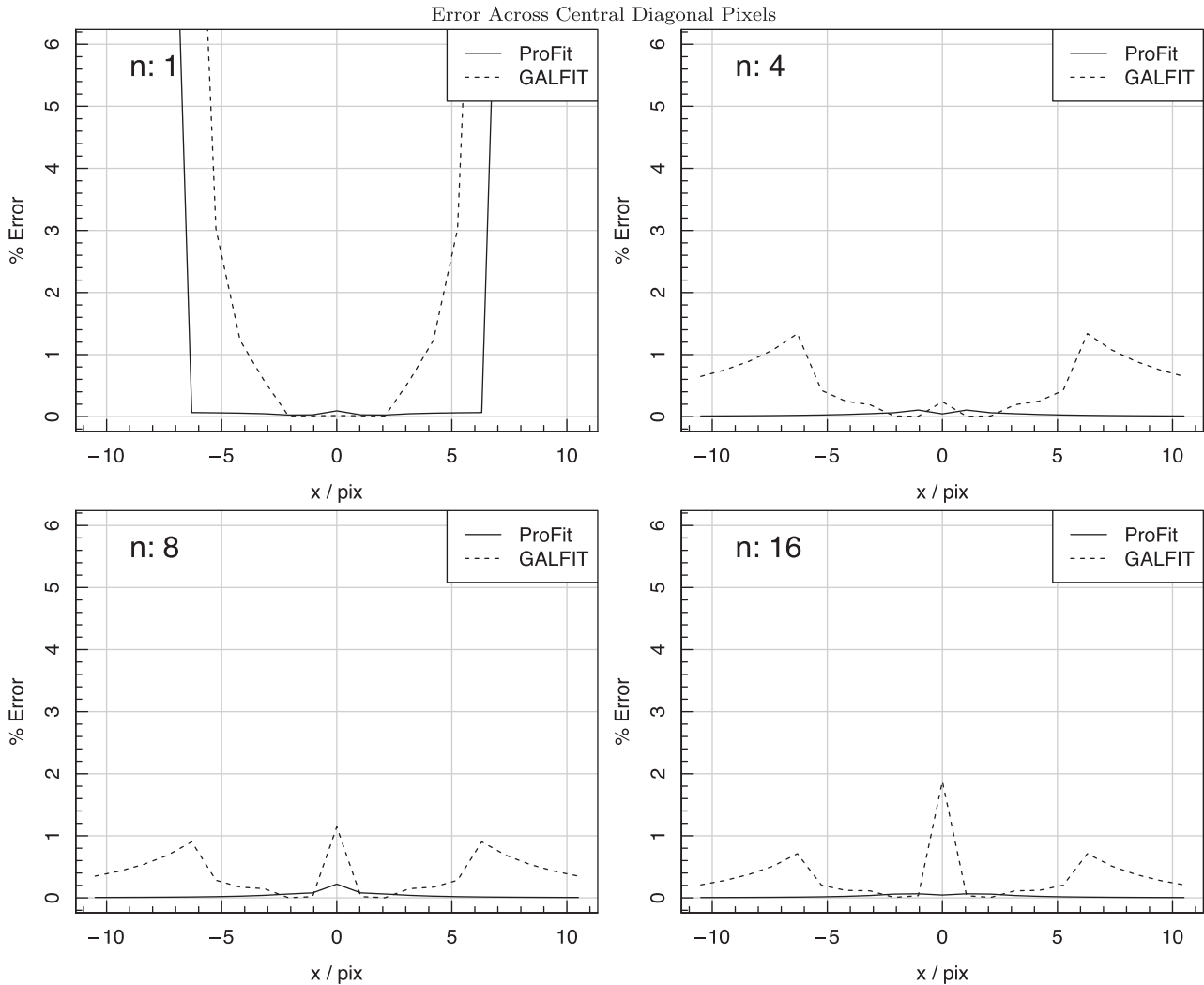


**Figure 9.** The pixel errors for pixels on the diagonal dotted lines in panel rows 1, 2, and 3 of Fig. 8 (i.e. close to the minor-axis centre of the image, where pixel errors tend to be largest).
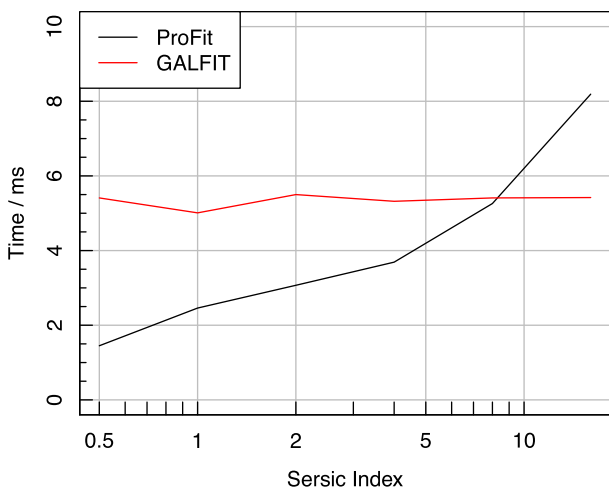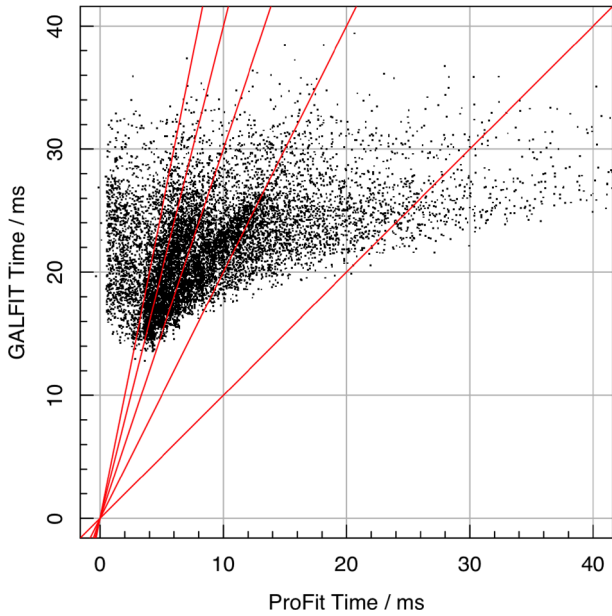


**Figure 10.** `libprofit` versus GALFIT model image generation time for sérsic indices 0.5–16 for $R_e = 2$pixels, $\theta = 60°$, and $A/B = 0.3$ in a $20 \times 20$ pixel image.

parameters being used to fit the model to the data (up to eight in the case of a single component sérsic model). In the regime of the statistics being truly normal and the model fitting well (i.e. a reduced $\chi^2$ near unity), the normal and $\chi^2$ distribution likelihoods converge to similar results. The chief distinction is that the $\chi^2$ statistic penalizes overfitting, since extremely small residuals across all pixels are highly unlikely. Both statistics are sensitive to residuals significantly above the shot noise ($\chi^2$ somewhat more so), such as non-axisymmetric features such as bars, spiral arms, and dust lanes. Nonetheless, if the dominant source of uncertainty in the image is truly shot noise rather than systematics, and the signal to noise is high, the $\chi^2$ statistic is appropriate.

However, if the count rates are relatively low, then the Poisson statistics must be used. In this case, PROFIT likelihoods are computed with

$$\ln \mathcal{L} = \sum_{i=1}^{N_i} \sum_{j=1}^{N_j} D_{i,j} \ln \lambda_{i,j} - \lambda_{i,j} - D_{i,j}!. \tag{32}$$

Here, $\lambda_{i,j}$ is the expectation for the number of photoelectrons counted in at pixel $i, j$, and $D_{i,j}$ is the integer number of photo-electrons recorded. Dropping the $D_{i,j}!$ term (since it depends only

**Figure 11.** `libprofit` versus GALFIT model image generation times for a large grid of 200 × 200 pixel images. Galaxy size (10 $R_e$), sérsic index (11 $n$), rotation angle (10 $\theta$), and axial ratio (10 $A/B$) were sampled on a grid, with 11 000 model images generated in total. The red lines show 1:1, 1:2, 1:3, 1:4, and 1:5 relative speeds for the same target model image, `libprofit` being substantially faster in almost all situations.

on the data) applies a global offset to log-likelihood, which makes no difference to any posterior inference since this is all computed in a relative manner. Expressed in this form and multiplied by a factor of −2 we recover the 'Cash' statistic $C$ (Cash 1979), which is similar to $\chi^2$ but for Poisson statistics. In the same manner that $-2\ln\mathcal{L} = \chi^2 + a$ for normal statistics, $-2\ln\mathcal{L} = C + a$ for Poisson statistics.

It is worth highlighting that returning images to true photoelectron counts is non-trivial and in our experience many published data sets do not produce enough ancillary information to return the images to true counts to better than an accuracy of a few factors. A particularly pernicious issue is that counts must be known within the CCD itself, not for photons at the top of the atmosphere. The latter is relatively trivial to compute after the fact if we know intrinsic properties of the source spectral energy distribution; however, the former requires estimates of atmospheric transmission, instrumental losses, detector gain, and quantum efficiency, etc. Sometimes this information is easy to obtain at least approximately (e.g. for the Sloan Digital Sky Survey, SDSS: Ahn et al. 2014), but this level of ancillary meta-data is in general rare. Added to this is the fact that optical survey data generally have copious counts per pixel: typically thousands of photoelectric counts. For these reasons, Poisson likelihoods should seldom be used for optical survey data, even if quite shallow. It is more appropriate for fitting X-ray data, where the counts are low and have the required meaning for Poisson statistics. Indeed, X-ray astronomy is the main subdivision that uses Poisson (usually in the guise of Cash) statistics.

If the data are nominally distributed in the normal statistics regime (i.e. many photoelectron counts per pixel), but in practice have a significant number of data points that are poorly represented by our attempted model (typical when trying to fit smooth two-dimensional sérsic profiles to well-resolved galaxies with asymmetries) then more robust Student-T distribution statistics might be appropriate. These distributions are approximately normal in the core but have

broad Lorentzian wings, which puts more likelihood mass at large distances from the normal core. This behaviour is controlled by the 'degrees-of-freedom', which in practice is not free and is instead estimated via maximum-likelihood from the data directly. The PROFIT Student-T likelihood is computed by

$$x_{i,j} = \frac{(D_{i,j} - M_{i,j})^2}{\sigma_{i,j}^2}, \tag{33}$$

$$\ln\mathcal{L} = -\ln\left(\sum_{i=1}^{N_i}\sum_{j=1}^{N_j}\frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\,\Gamma\frac{\nu}{2}}\left(1 + \frac{x_{i,j}}{\nu}\right)^{-\frac{\nu+1}{2}}\right). \tag{34}$$

PROFIT gives the users easy access to all four types of commonly used likelihood statistics, and it is for the user to decide the most appropriate given the quality and type of data at hand. For high S/N images of galaxies, the Student-T statistic behaves well over a broad regime given its robustness to outlier flux values. For this reason, it is selected as the default option when setting PROFIT up for fitting. In future, we will add covariance likelihoods (more appropriate for data that exhibit large amounts of pixel correlation, e.g. NIR, see Andrews et al. 2014), but these are not present in v1.0.

### 3.3 Parameter optimization

Once the observational data have been correctly assembled and the most appropriate type of likelihood statistic has been chosen, the user is free to fit a model image. PROFIT requires the user to suggest an initial parametrization of the model, which need not be close to the optimal parameters with robust optimizers. The user is free to fix some components and fit others (e.g. they might wish to fix the disc sérsic index to 1). They are also free to provide prior distributions for each parameter (important for formal Bayesian analysis), hard limits, and/or constraints on the allowed fitting region, to specify whether parameters are optimized in log or linear space and to specify additional constraints between parameters (e.g. pair parameters together). The detailed process for specifying these options is provided in the PROFIT manual and example vignettes, so we will not repeat the full description here.

Once the fitting structure has been specified within PROFIT (`profitSetupData`, as mentioned above) the user can interface with a number of popular optimization routines. The R PROFIT package provides examples of using the base `optim` function, the open source LaplacesDemon package that gives access to gradient optimizers via `LaplaceApproximation` and a large suite of Markov Chain Monte Carlo (MCMC) samplers via `Laplaces-Demon`, and the genetic algorithm CMA-ES package.[2] These four routes to optimization offer over a hundred distinct optimizers. The CRAN collates packages that tackle particular problems types: optimization[3] and Bayesian[4] analysis task views suggest over a hundred additional packages that themselves potentially contain multiple algorithms.

To the uninitiated this can seem daunting, but the reality is that a core few packages and optimizers give the user a solid foundation for a diverse range of fitting problems. Sample vignettes provided in

---

[2] Versions of `LaplacesDemon` and CMA-ES implementing runtime limits for use on shared (super)computers are available at https://github.com/taranu/LaplacesDemon and https://github.com/taranu/cmaeshpc, respectively.

[3] cran.r-project.org/web/views/Optimization.html

[4] cran.r-project.org/web/views/Bayesian.html

Time to create $200 \times 200$ pixel images
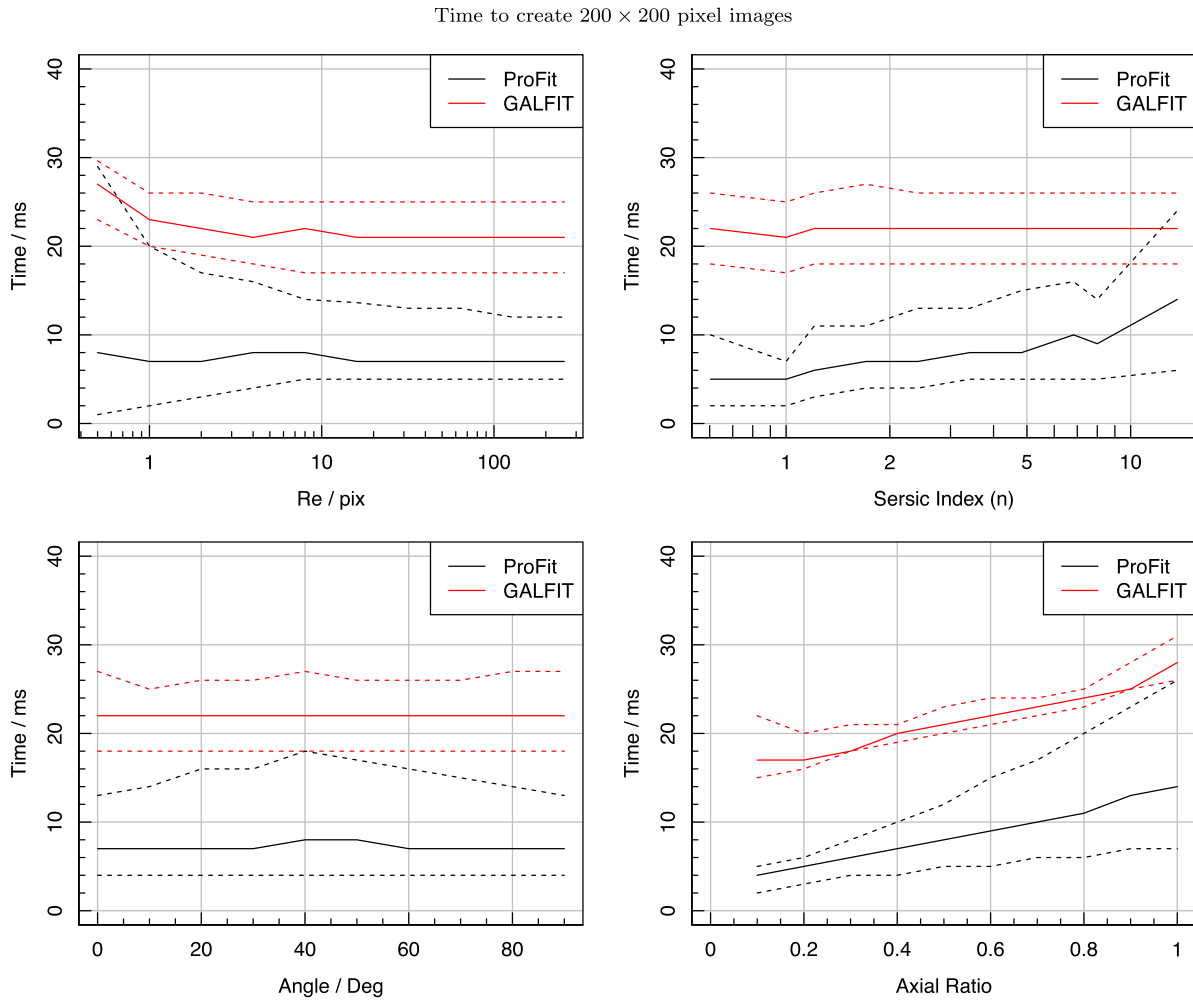


**Figure 12.** A large grid of model parameters were used to create $200 \times 200$ pixel images with centred single sérsic component galaxies (see Fig. 11 for details). The marginalized median image generation time are shown in these plots for `libprofit` and GALFIT, with the effective $1\sigma$ computation time spread shown as dashed lines.

the R PROFIT package goes through some example fitting problems in detail, but we will summarize the approach here. The basic strategy for preparing the inputs for galaxy fitting is

(i) REQUIRED: read in an image of an observed galaxy (in principle any format, but e.g. FITS).

(ii) OPTIONAL: read in a mask with the same size, pixel scale, and astrometry as the image.

(iii) OPTIONAL: read in a sigma image with the same size, pixel scale, and astrometry as the image.

(iv) OPTIONAL: read in a segmentation map with the same size, pixel scale, and astrometry as the image.

(v) OPTIONAL: read in a PSF image with the same pixel scale as image.

(vi) REQUIRED: read in an initial model list (as detailed above).

(vii) OPTIONAL: read in a list with the same structure as initial detailing which parameters PROFIT will fit (the others will be fixed at their initial values).

(viii) OPTIONAL: read in a list with the same structure as initial detailing which parameters PROFIT will fit in log space (the others will be fitted in linear space).

(ix) OPTIONAL: read in a function detailing the calculation of the prior log-likelihood.

(x) OPTIONAL: read in a list with the same structure as initial detailing the allowed limits for each parameter when fitting.

(xi) OPTIONAL: read in a function detailing the calculation of additional constraints to apply to parameters, e.g. pairing parameters together or making them scale relative to each other in a fixed manner.

(xii) REQUIRED: all of the above two-dimensional matrix images, list structures, and functions are provided to the high-level `profitSetupData` function that organizes the information into an object of class profit.data ready for fitting with third-party optimization engines.

Once the user has created the profit.data object, which includes some additional options for choosing the type of likelihood and setting the verbosity of the fitting process, a basic strategy for fitting might look like as follows:

(i) Initially find a maximum likelihood galaxy model solution using the base R `optim` function using the Broyden, Fletcher, Goldfarb, and Shanno (BFGS: Fletcher 1970) algorithm (which is fairly robust to spurious minima).

(ii) Take the solution from the BFGS optimization as a starting point and use `LaplacesDemon` component-wise hit-and-run metropolis (CHARM) to make MCMC samples of the likelihood

space (this is very robust to local minima, and can move large distances from poor solutions; hence hit-and-run).

(iii) Use diagnostic tools provided with the LaplacesDemon package to test for the quality and convergence of the model fit, including tests for auto-correlation between parameters, the number of effective stationary samples and traditional parameter triangle plots.

(iv) Assuming the fit is flagged as being well converged and well behaved, the log marginal likelihood (LML) can be computed to aid comparisons between models.

The above fitting process can be carried out for models of varying complexity. If they are well converged, then the LML computed can be used to calculate Bayes factors between simpler and more complex models, offering an objective route for deciding how complex the model needs to be.

# 4  PROFIT IN PRACTICE

## 4.1  Single galaxy case study of G266033

The PROFIT package includes 10 example data sets with all the required inputs for fitting. These examples use public Sloan Digital Sky Survey (SDSS; Ahn et al. 2014) and the Kilo Degree Survey (KiDS; Kuijken et al. 2015) data for galaxies selected from the Galaxy And Mass Assembly (GAMA; Driver et al. 2011; Liske et al. 2015) survey that are at reasonably low redshift and are deemed to have multiple components (as found in Lange et al. 2016). The vignettes included with the software give examples of fitting these data using high-level R and PYTHON interfaces.

As a case study, Fig. 13 shows the quality of MCMC fit and residuals that are possible using PROFIT for G266033. A qualitative assessment of the residuals shows that we are able to remove the major bulge and disc structural components, but that non-axisymmetric components remain (the spiral arms in particular). The three rows of panels show the bulge (top), disc (middle), and combined (bottom) components. The outputs of the CHARM MCMC sampling are shown in Fig. 14. Here, only stationary samples are shown (i.e. there is no 'burn-in' of the sampling), and whilst there is evidence of some weak covariance between some parameters (e.g. disc magnitude (Dmag) and effective radius (Dre)), in general the posteriors look well sampled and consistent with multivariate normal distributions. This latter characteristic is useful when estimating marginalized parameter errors.

Using the R interface to PROFIT, more quantitative information can be extracted from the fit. Fig. 15 shows the output of the `prof-itMakePlots` function, which as well as producing the standard residuals, also outputs how they compare to a reference normal and Student-T distribution. The $\chi^2$ excess is also demonstrated with respect to a $\chi^2$ distribution with one degree of freedom. Finally, a residual image scaled by the pixel error $\sigma$ is provided. This shows that there are some significant residual spiral arm structures that are highly unlikely from the point of view of the idealized generative model we are fitting.

As discussed in the Introduction, it is common in the field of galaxy modelling to collapse two-dimensional data into a one-dimensional form. Whilst PROFIT does not evaluate likelihood using this one-dimensional form, it does provide functions for producing such isophotal data and the associated plots (`profitEllipse` and `profitEllipsePlot`). In brief, these functions use the geometric parameters of the disc and bulge components of a simple two-component model to extract isophotal annuli, allowing for the standard geometric distortions such as ellipticity and boxiness (if present). The one-dimensional profile of G266033 with overlaid model fits is shown in Fig. 16.

Given the residual flux not replicated by the model, the question remains how representative any posterior error distributions might be. To estimate the impact an imperfect model can have on the returned posterior error estimates, the residual is subtracted from the initial image. Fig. 17 shows the effect of fitting such an input where the image noise has been added back via Poisson sampling of the convolved model image. The residuals almost entirely disappear, and the $\sigma$ scaled residuals show no radial structure at all. Comparing the posteriors to the original input data and the residual-removed input data, the former has typical marginalized posterior errors for each parameter that are a factor of $\sim 2$ *larger* (spanning the range 1.3–3) with best-fitting values that agree to within 1 per cent. This is encouraging, since it suggests in this particular case, where PROFIT is correctly identifying the major structural components, the presence of non-axisymmetric components creates posteriors with larger errors. How such fitting behaves will depend on the data at hand (i.e. this should not be assumed to be true in general), but such a test is easy to implement, and should be part of a standard PROFIT workflow.

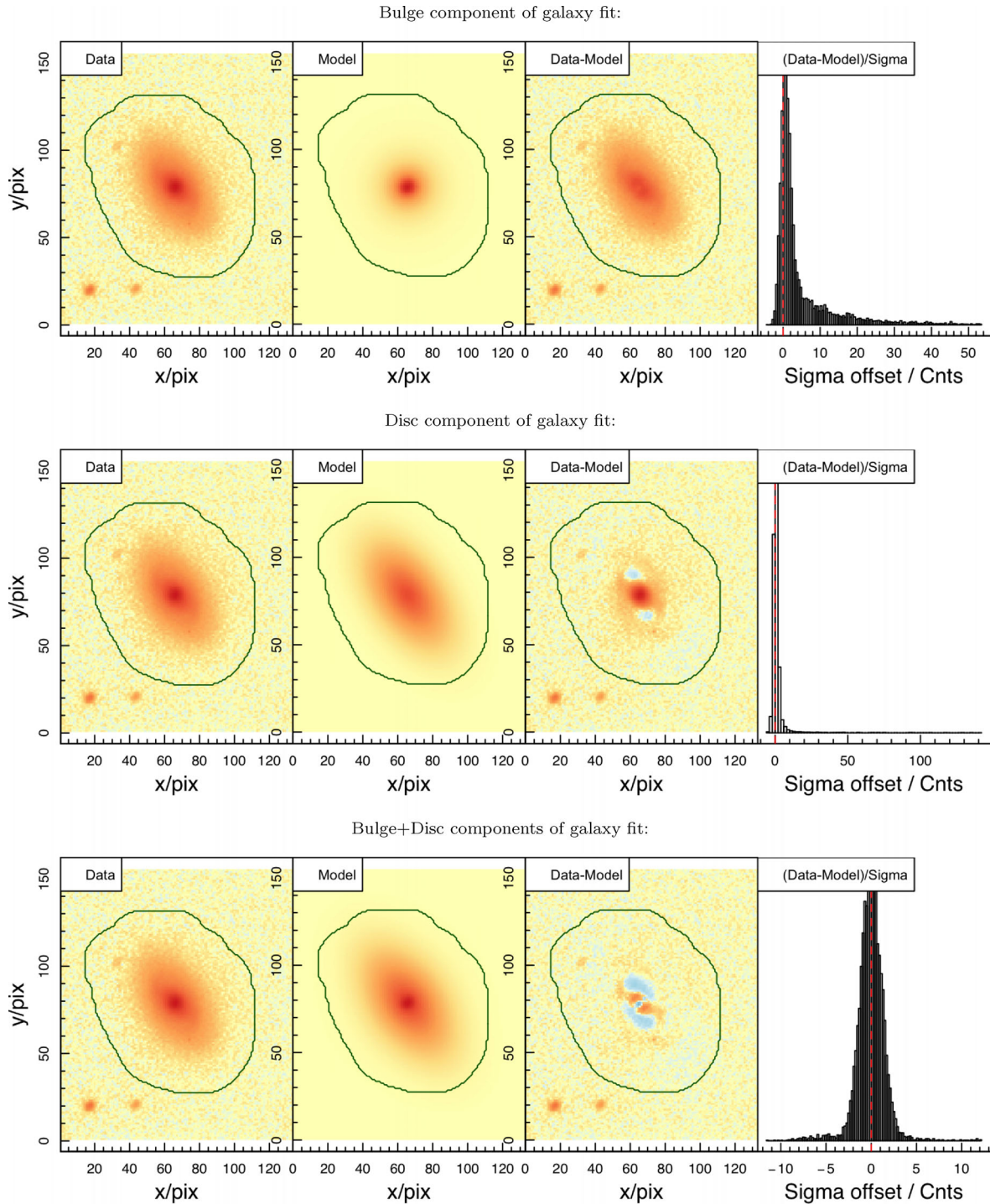## 4.2  Running PROFIT and GALFIT on a small sample from SDSS and KiDS

As mentioned above, the R PROFIT package comes with 10 fairly isolated example galaxies included that have imaging from SDSS and KiDS. All of these are well-resolved galaxies taken from the GAMA survey where we already have well-converged bulge and disc fits from Lange et al. (2016). To create this sample, 40 galaxies from Lange et al. (2016) were randomly sampled and these were ranked in terms of fit-ability (removing galaxies with more complex backgrounds and nearby sources). Of these, the 10 largest were chosen as example galaxies. As such this sample, whilst small, is broadly representative of how common different classes of bulge-disc systems are (e.g. we are dominated by lower $B/T$ systems). Consistent sky subtractions were made using LAMBDAR (Wright 2016), and segmentation maps and PSFs were created using an updated version of SIGMA (Kelvin et al. 2012). We then proceeded to fit these 10 galaxies using both SDSS and KiDS inputs, and using PROFIT and GALFIT. The free parameters used throughout were $x$ and $y$ centres, bulge mag, $R_e$ and $n$ ($A/B$ for the bulge was fixed to be 1), and disc mag, $R_e$, $A/B$ and $\theta$ ($n$ for the disc was fixed to be 1, i.e. exponential).

The input parameters were taken from the fits in Lange et al. (2016), although it should be noted that PROFIT (used in full MCMC mode) is largely insensitive to the inputs used, as long as the total input magnitude is approximately correct (i.e. within a couple of magnitudes of the correct value). Used in pure downhill gradient mode PROFIT suffers from similar local minima issue to GALFIT (see the discussion of GALFIT convergence in Lange et al. 2016).

The full range of comparison Figs A1–A6 are included in Appendix A. Here, we compare how well these different combinations of decomposition codes and data sources affect the returned values for bulge and disc magnitudes.

### 4.2.1  Comparison of bulge magnitudes

Fig. 18 compares the agreement between estimated bulge magnitudes. The one major disagreement between codes is the faintest bulge measured. In PROFIT, the decomposition preferred removing the bulge entirely, often hitting our specified lower limit of 30 mag. Using GALFIT, the code does not allow the solution to move huge

Bulge component of galaxy fit:



Disc component of galaxy fit:



Bulge+Disc components of galaxy fit:



**Figure 13.** Result of the *r*-band MCMC decomposition of G266033. The top row shows the data (left), the bulge model (second left), the data-model (second right), and the histogram of residuals. The second row shows the data (left), the disc model (second left), the data-model (second right), and the histogram of residuals. The bottom row shows the data (left), the bulge-disc model (second left), the data-model (second right), and the histogram of residuals. Even with an optimal bulge-disc model, there is still evidence of residual structure.

distances from the initial estimates unless the likelihood terrain is very smooth, which occurs only when the model reproduces the data with no clear residuals. Fig. 19 shows the fit and residuals for the PROFIT fit for this galaxy, suggesting there is little need for any bulge at all.

Otherwise the main conclusion that can be drawn is that the codes correlate tightly, and so do the data sets, but in all cases,

the scatter is much larger than the errors suggested by GALFIT or PROFIT. Although the errors returned by PROFIT were consistently larger for these galaxies, they are still smaller than the points in the panels.

A general remark is that we find more consistency between codes than between data sets. Removing the outlier galaxy G266035, we measured the intrinsic scatter using the HYPER-FIT code of Robotham
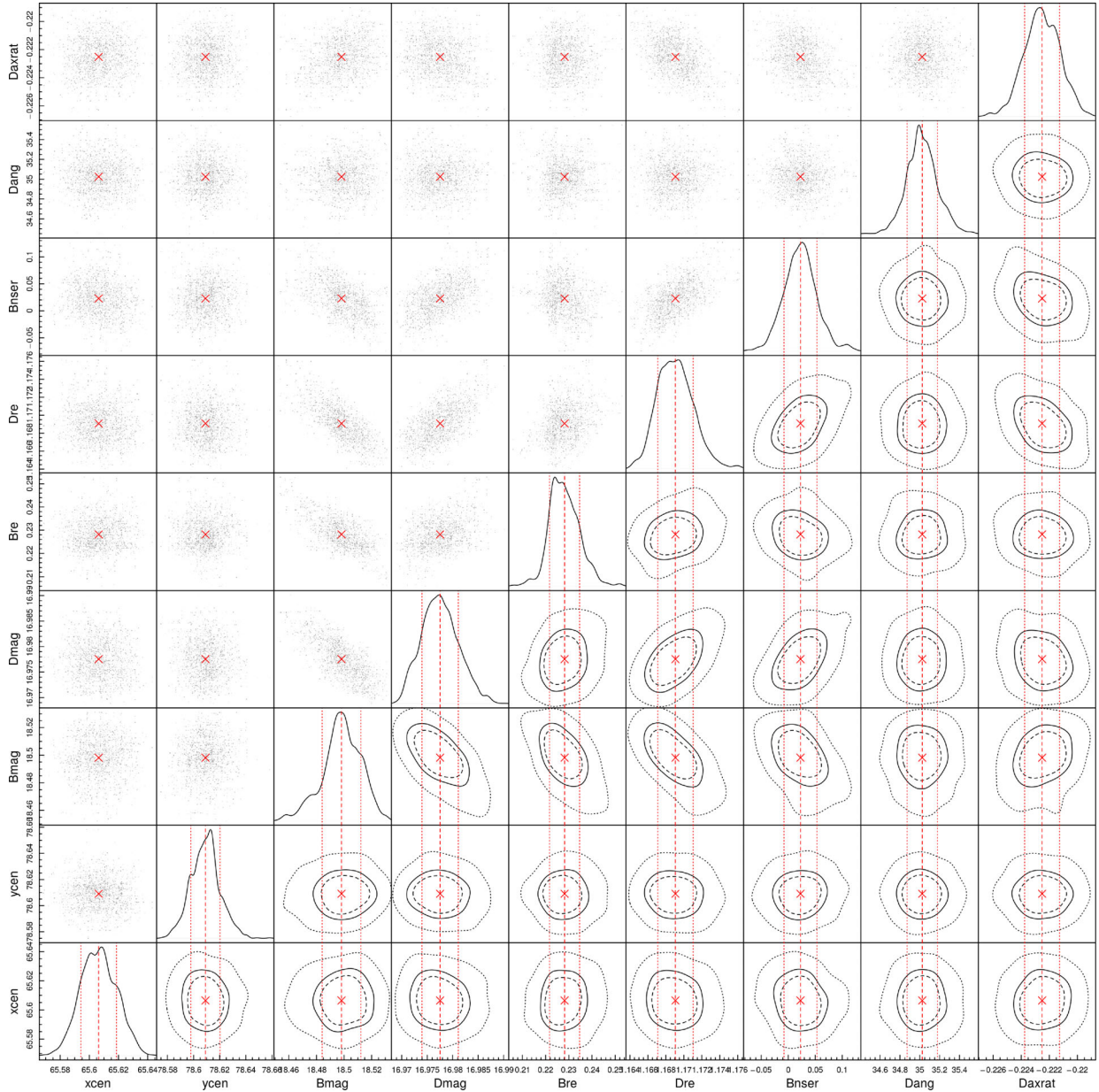
**Figure 14.** The triangle plot of the stationary MCMC chains for the model fit of G266033 shown in Fig. 13. The top-left of the triangle shows the raw samples, the bottom right shows the contoured version of the samples, with dashed/solid/dotted lines containing 50 per cent/68 per cent/95 per cent of the samples. The diagonal one-dimensional density plots show the marginalized distributions for each parameter. In this case, nine parameters were fitted using CHARM in the `LaplacesDemon` function: $x_{cen}$ (xcen), $y_{cen}$, the bulge magnitude (Bmag), the disc magnitude (Dmag), $R_e$ of the bulge (Bre), $R_e$ of the disc (Dre), the sérsic index of the bulge (Bnser), the rotation angle $\theta$ of the disc (Dang), and the axial ratio of the disc (Daxrat). From the triangle plot, it is clear that in this case the bulge magnitude (Bmag) and disc magnitude (Dmag) show the most covariance with other parameters, but in general the fit is well converged. Note the covariances seen tend to be galaxy specific, and might not be present in generic galaxy modelling cases.

& Obreschkow (2015) and find it to be reduced by more than a factor of 2. It is at a minimum for the SDSS data (0.14 dex versus 0.20 dex scatter orthogonal to the 1–1 line), suggesting that some aspect of our data processing is more internally consistent for the SDSS data. This might be the background subtraction or the PSF determination.

### 4.2.2 Comparison of disc magnitudes

Fig. 20 compares the agreement between estimated disc magnitudes. In general, we find more self-consistent fits for discs both between codes and between data. Whilst the smallest intrinsic scat-

ter is again found for SDSS using PROFIT and GALFIT, the results for PROFIT using SDSS and KiDS are not far behind. The intrinsic scatter for KiDS using PROFIT and GALFIT, and GALFIT using SDSS and KiDS are notably worse (the latter showing the most scatter).

For all these fits, we find that the PROFIT MCMC solution provides a *more likely* fit both when forcing the PROFIT parameters through GALFIT, and when forcing the GALFIT parameters through PROFIT. Usually the increases in likelihood when using PROFIT are substantial, suggesting the difference is due to local convergence issues with the Levenberg–Marquardt (Marquardt 1963) algorithm
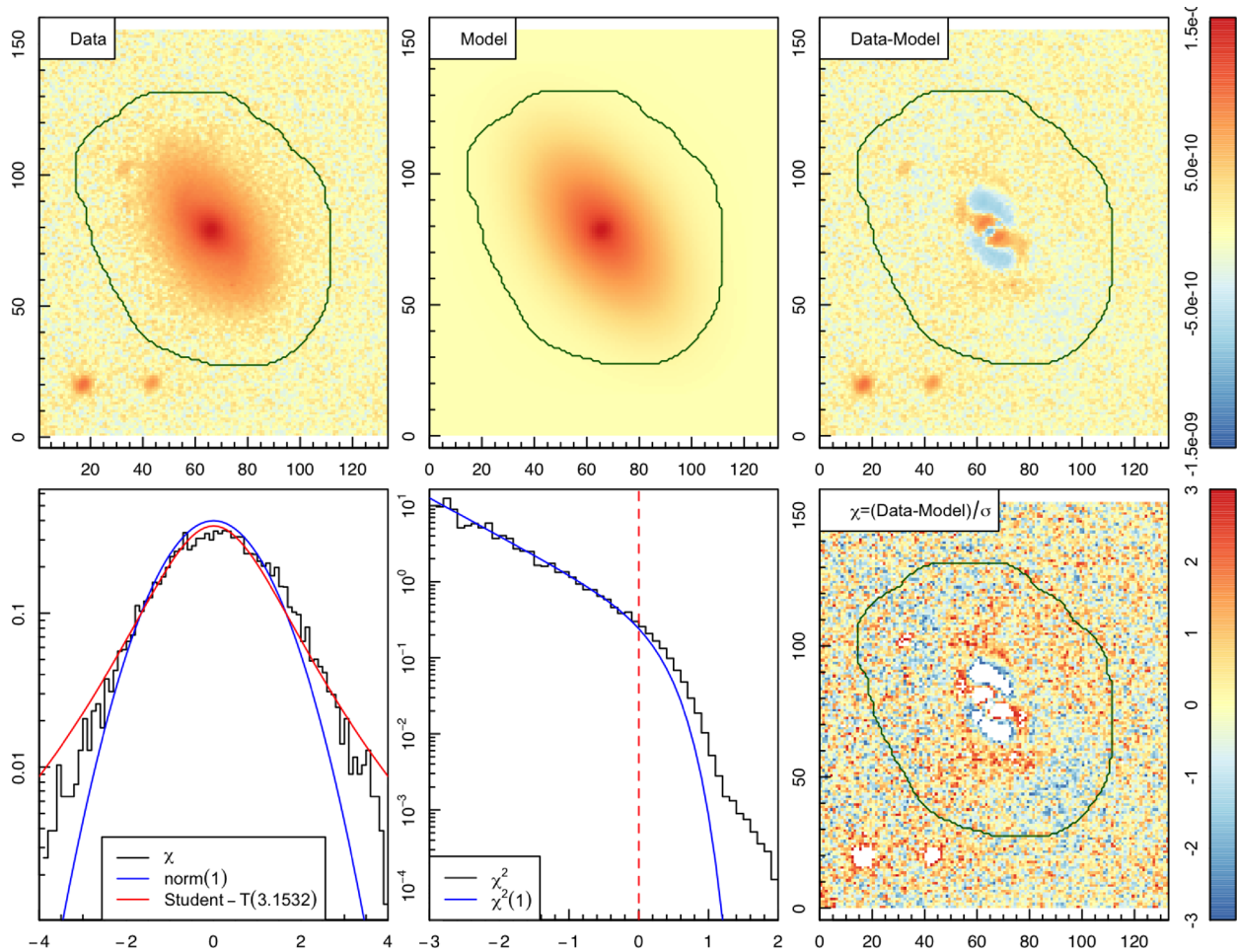
**Figure 15.** The output of the provided high-level `profitMakePlots` function included in the R version of PROFIT. The top panels are the same as Fig. 13 for G266033. The bottom panels show the histogram of residuals on a scale of $\sigma$ (left), the $\chi^2$ residuals compared to a $\chi^2$ distribution with one degree of freedom (middle), and the two-dimensional residuals shown in terms of $\sigma$ significance. In this case, the part of the galaxy that is well approximated by the model has well-behaved Normal distribution errors, but there are substantial deviations for residuals in regions containing non-smooth structure (e.g. spiral arms etc.).

built into GALFIT, rather than issues to do with model accuracy that we explored in Section 2.2.

A general conclusion is that we appear to be able to recover more self-consistent results for disc properties compared to bulge properties. Further evidence for this is shown in Appendix A, where disc $R_e$ is found to be more reliably measured than bulge $R_e$. Bulge $n$ is notably the hardest property to measure consistently, a finding in agreement with recent work by Savorgnan & Graham (2016). We investigate in more detail how well we expect to be able to recover various bulge-disc properties for different quality and depth data in the next section.

## 5 APPLYING PRoFIT TO SIMULATED CURRENT AND FUTURE IMAGING SURVEYS

Having described and tested our new PROFIT decomposition code in some detail, we finally investigate how it performs on data of different depth and quality. For the purposes of these tests, we create and fit bulge-disc systems in $40 \times 40$ arcsec image stamps. We investigate three sources of survey imaging that are indicative of the types of data that PROFIT will be used on en-masse in future: SDSS, KiDS, and LSST. Since these results are purely indicative, we make some

simplifying assumptions for the three surveys. Table 1 presents the assumptions made when simulating and fitting the galaxies. These are not precisely the values advertised for the various surveys, but are indicative given the inevitable variations in data quality. For LSST, these estimates are for the stacked 5-yr survey. Since we know the precise PSF used to convolve the model and the images themselves are precisely background subtracted, the following results are the upper limits on the decomposition performance we can hope to expect.

The simulations themselves sample 10 000 profiles from a uniform grid of total magnitude ($15 \leq m_T \leq 22$), bulge fraction ($0 \leq B/T \leq 1$), logarithmic bulge effective radius ($0 \leq \log_{10}(R_e) \leq 1$), logarithmic disc effective radius ($0 \leq \log_{10}(R_e) \leq 1$), logarithmic bulge sérsic index ($0 \leq \log_{10}(n) \leq 1$), and disc axial ratio ($0 \leq A/B \leq 1$). The bulge axial ratio was assumed to have $A/B = 1$ and the disc sérsic index was fixed to $n = 1$. PROFIT never appears to struggle to find the correct disc angle $\theta$, so we only simulated galaxies with the minor axis varying on the $x$-axis of the image. Note we do not impose internal correlations between parameters, so these results span an overly generous range of structural parameter space.

Fig. 21 shows the results of these simulations in a compact manner, comparing the fraction of fits that are found to converge (which
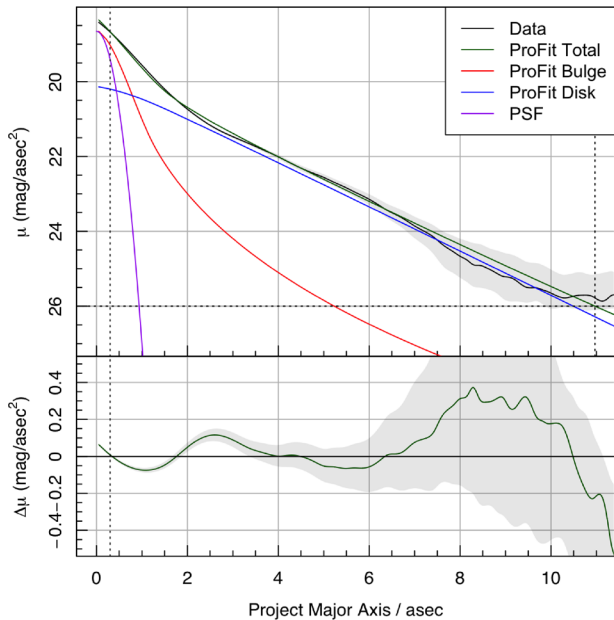
**Figure 16.** The approximate projected one-dimensional profile for G266033. The grey shaded region shows the $1\sigma$ error region for the data pixels, and ideally the total model (green line) should sum to follow this distribution quite closely. At a few radii, the data and model are in some tension (at the 0.1 mag level in surface brightness), but for the majority of the profile they agree very well.

means they find solutions that are not hard limits imposed within PROFIT) and the typical average separation between all input and output parameters in log-space. These properties are plotted in Figs B1–B7 for all major observables in Appendix B, but here we show the results versus the most observable input parameter: the total magnitude of the galaxy. Unsurprisingly, all the results are best when the galaxy is intrinsically bright and the source signal dominates over the sky background. Of perhaps more surprise is that we see a large improvement in decomposition performance when moving from SDSS to KiDS quality data, but a relatively minor improvement when we move from KiDS to LSST. This appears to be due to the major improvement being limited to surface brightness depth, the pixel scale, and the typical PSF FWHM is not expected to be much (if at all) improved between KiDS and LSST (one caveat being the latter has a much larger field of view, so further unsimulated sky-subtraction gains could be expected).

To briefly summarize the other trends presented in the Appendix B: lower bulge and disc $R_e$, lower disc $A/B$, lower bulge $n$, and a moderate $B/T \sim 0.5$ all lead to improved fitting results. For $R_e$, this is simply because lower $R_e$ means high surface brightness for a given magnitude, and the components become easier to distinguish against the background. We start our simulations at an $R_e$ of 1 arcsec, so the caveat to make is that much below this ($R_e$ less than half the PSF FWHM), and the parameters become indistinguishable structurally. Even in this regime, we can still expect the component magnitude to be well recovered, it is just hard to resolve structure inside the PSF scale.

It is clear from these results that only in the most favourable regimes can we expect complete *and* robust bulge–disc decompositions, even using the best current and next-generation deep survey data. This is true even when we are operating in the regime of moderately low-redshift galaxies extracted from surveys such

as GAMA. A future low-redshift focused WAVES-Wide survey (Driver et al. 2016), being inherently lower redshift due to its photo-$z$ pre-selection (with typical redshift likely to be similar to SDSS DR10), is the best prospect for such future studies given the target of high-redshift completeness, and a guarantee it will have deep good-quality imaging data (since KiDS will act as the primary input source for targets).

## 6 DISCUSSION AND CONCLUSIONS

In this paper, we have presented our new publicly available galaxy modelling and decomposition code PROFIT. The core code comes in a standalone c++ library (`libprofit`) that allows for easy access to higher level languages (e.g. R and PYTHON) through a simple API.

The core advances that PROFIT offers over currently available software are that it:

(i) is fully open-source with multiple active developers,

(ii) offers a standalone library (`libprofit`) for accurate and fast pixel integrations when generating a model,

(iii) can be extended with new profiles in a simple well-documented manner,

(iv) allows for simple or complex priors on parameters (an important aspect of Bayesian analysis),

(v) offers a range of likelihood calculations,

(vi) is untied to any specific optimizers but has easy access to downhill minimization, genetic algorithm, and MCMC routines,

(vii) can fit parameters in log or linear space,

(viii) allows for simple or complex additional constraints between parameters,

(ix) offers brute-force and FFT PSF convolution options, with automatic benchmarking to select the fastest strategy.

Initially PROFIT comes in three varieties: a fully featured R package (ProFit, discussed in detail in this paper), a basic PYTHON wrapper (pyprofit), and a command line terminal interface (profit-cli). The R package is the most advanced in terms of features, and we have a longer term aim to bring the PYTHON package up to the same level of sophistication. The command line interface only exists for the easy generation of model images via calling the underlying `libprofit` library, so it does not provide any built-in optimization. By separating the model image generation (which should be an objective black-box task achieved through an API) into a separate `libprofit` library, users are not tied to our high-level solution for the much more subjective problem of galaxy fitting (with many caveats over masking, convolution, likelihoods etc.). We are aware of at least one community fork of `libprofit` already that uses diffusive nested sampling (DNEST; Brewer, Pártay & Csányi 2011) in order to overcome sampling of highly multimodal data (Huijser et al., in preparation). Extensions to other languages using the library API are encouraged, but simpler workarounds using the command line interface are also possible for less programming-savvy users. To avoid community confusion, we request that any forks that are released append the name in some manner (e.g. ProFit_STAN). The aim would be to pull any substantial and useful changes into the main PROFIT branch in the longer term.

The first versions of the PROFIT code stemmed from preliminary work by ASGR. Following this, DST and RT have added many new features and have heavily modified and expanded the functionality of the software. From the early stages, PROFIT has been designed in a highly modular manner. This is important to prevent the user becoming limited to our default choice of profile, likelihood
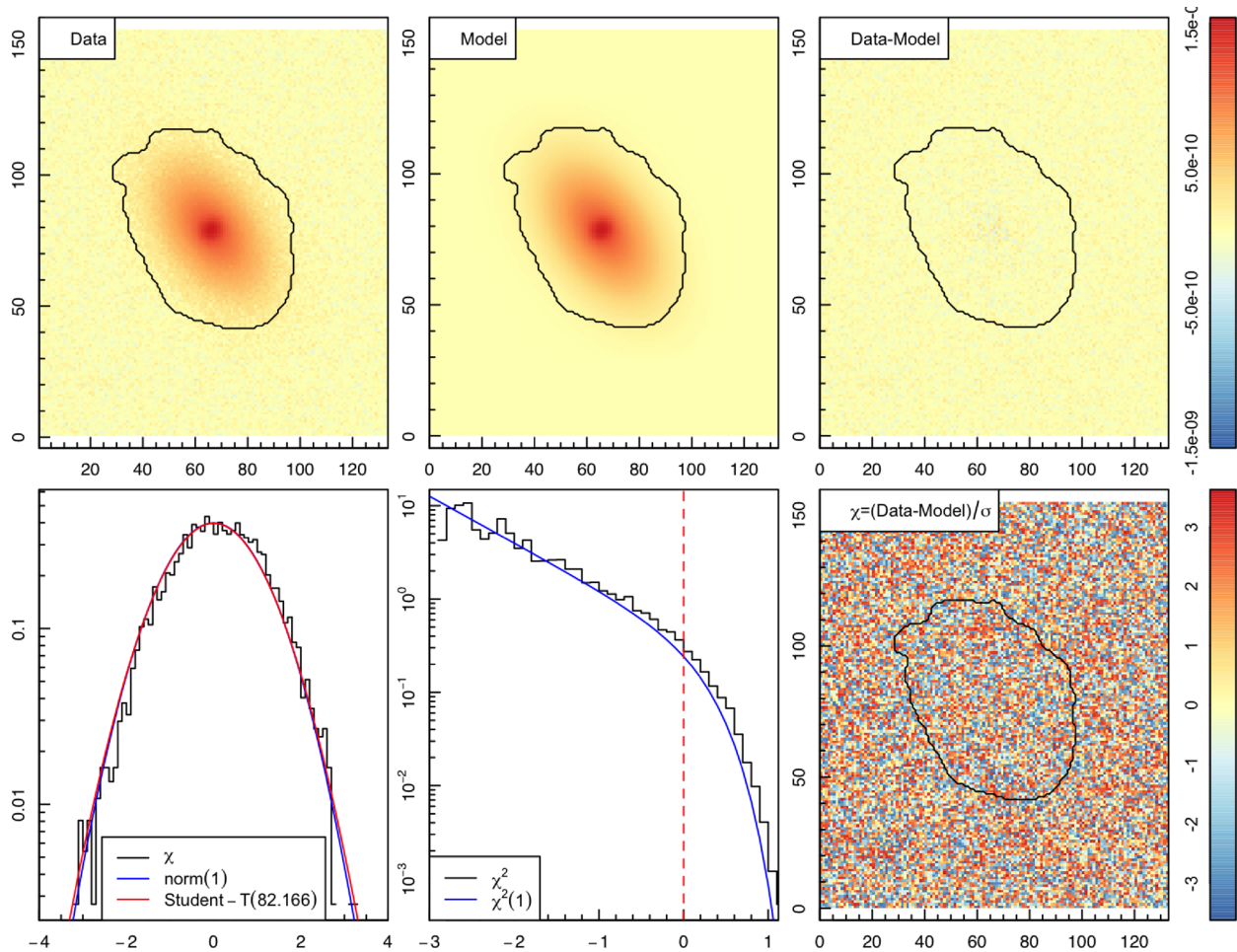
**Figure 17.** As per Fig. 15, but with the residuals removed. Image noise has been added back in via Poisson sampling of the convolved model image. This means an exact model can be fit to the data, hence the residuals are consistent with an idealized model fit.

evaluation, or optimization method. The core `libprofit` library is designed to be fast and accurate at integrating and convolving target model images, where the likelihood function to compute, and optimization engine to use, are largely choices for the user via higher level interfaces. The R package version of PROFIT contains a large number of examples using simple downhill gradient schemes, more complex genetic algorithms and more computationally expensive MCMC techniques. Users are encouraged to build from these examples to use more or less sophisticated engines as appropriate.

The various libraries and higher level interfaces are all available on GitHub, and future support and functionality will be added through these repositories. This paper necessarily refers to a static v1.0 of PROFIT, as such specifics regarding operation and options should be derived from the public repositories and documentation found there, rather than this text. The most reliable long-term location for the base `libprofit` library is https://github.com/ICRAR/libprofit. The higher level R implementation of PROFIT is available at https://github.com/ICRAR/ProFit and the PYTHON variant is maintained at https://github.com/ICRAR/pyprofit. This paper was written using the R implementation exclusively.

Community use and feedback is encouraged, especially via raising *issues* and *pull* requests through GitHub. In particular, the re-

quirements outlined in Section 1 are not an exhaustive list, so users are encouraged to submit feature proposals and requests. Currently planned extensions include built-in support for Monte Carlo image generation, simultaneous fitting of multiple images (in the same band) and/or multiple bands, zero-point calibration, covariance likelihoods, and PSF fitting. Indeed, the R package already includes a vignette with PSF fitting examples, a full description of which is omitted here for brevity.

In this paper, we have investigated the application of PROFIT to both SDSS and KiDS data, using GALFIT reference runs as a comparison. In pragmatic terms, PROFIT and GALFIT achieve consistent results given the same input data, with at least as much scatter being produced by changes between data source (in this case, SDSS versus KiDS). This is encouraging in that it validates the broad findings of recent work that made exclusive use of GALFIT, and lends credibility to our future applications of PROFIT to a number of different data sources. The main difference is a large degree of intervention was required when running GALFIT via running on a large grid of initial conditions and post-processing the results (see Lange et al. 2016), whereas PROFIT was run by AM with just a basic wrapper using one of the many available MCMC routines in R and converged to reasonable global solutions with little user intervention.
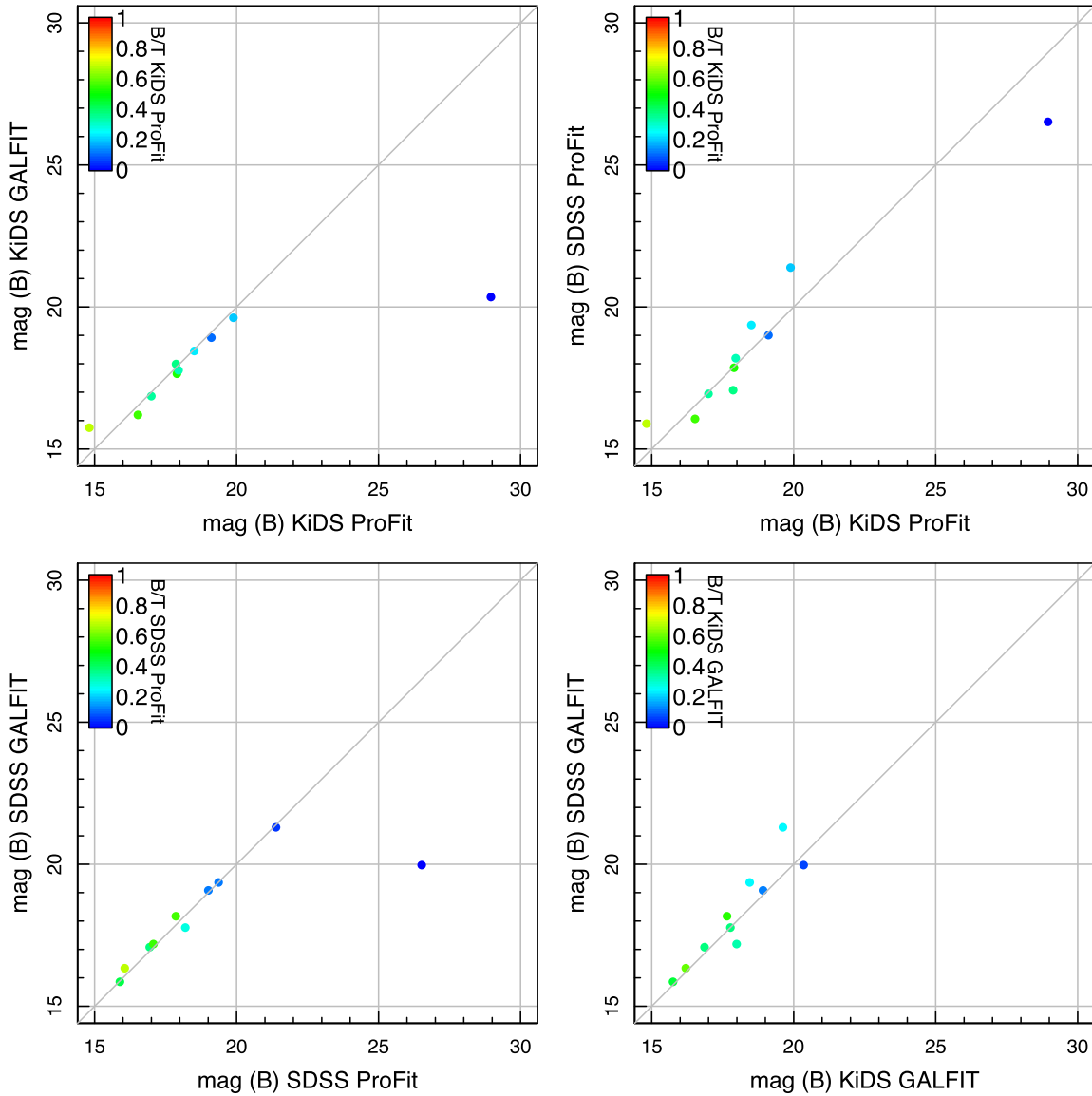
**Figure 18.** Panels comparing the measured bulge magnitude for the 10 well-resolved galaxies included with the R PROFIT package. Top-left shows the comparison of GALFIT and PROFIT using KIDS data. Bottom-left shows the comparison of GALFIT and PROFIT using the SDSS data. Top-right shows the comparison of SDSS and KiDS using PROFIT. Bottom-left shows the comparison of SDSS and KiDS using GALFIT. In all cases, the B/T that is used to colour the data points is derived from the fit used for the x-axis. Fit errors provided by both GALFIT and PROFIT are smaller than the points, so are not plotted here.
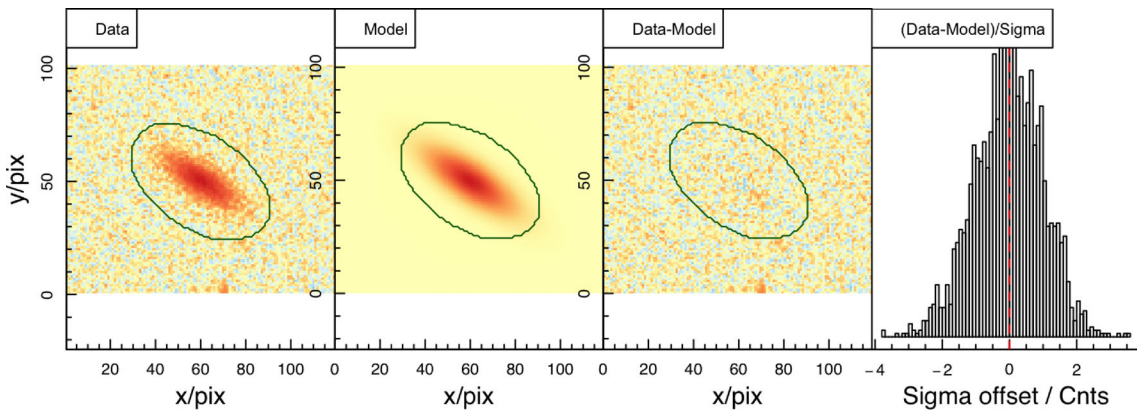


**Figure 19.** The PROFIT SDSS decomposition for G266035 which effectively has no bulge since $B/T < 10^{-4}$. Judging by the almost pure noise residuals, it seems reasonable to conclude that the data have little requirement for a bulge component.
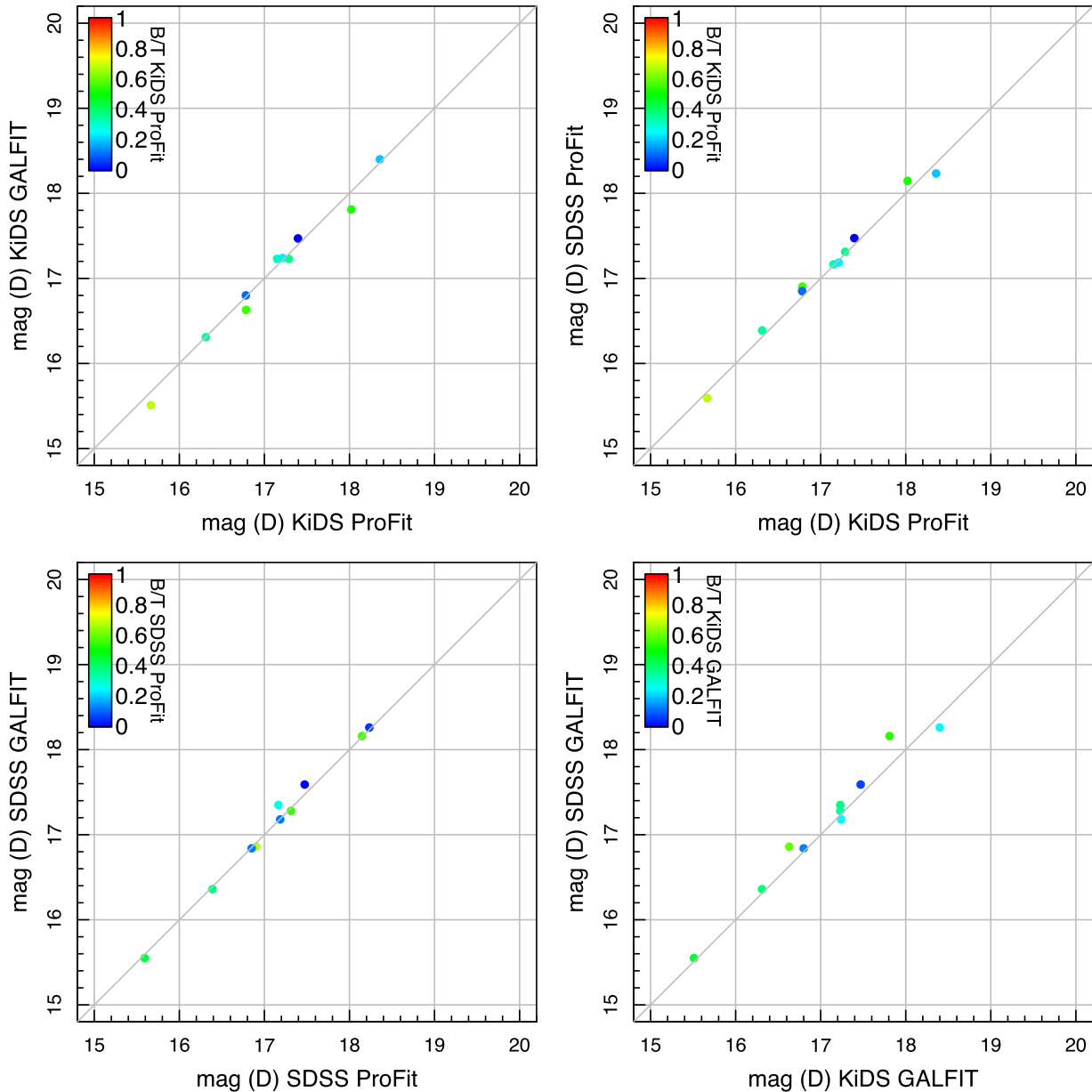
**Figure 20.** Panels comparing the measured disc magnitude for the 10 well-resolved galaxies included with the R PROFIT package. See Fig. 18 for details.

**Table 1.** Assumed image quality parameters for different surveys.

| Survey | Scale (arcsec pix$^{-1}$) | PSF FWHM (arcsec) | $1\sigma$ $r$-band depth |
|--------|---------------------------|-------------------|--------------------------|
| SDSS   | 0.4                       | 1.4               | 24                       |
| KIDS   | 0.2                       | 0.6               | 26                       |
| LSST   | 0.2                       | 0.6               | 28                       |

## ACKNOWLEDGEMENTS

**Figure 21.** Comparisons of 10 000 random simulations for the SDSS, KiDS, and LSST quality data. Top panel shows the fraction of ProFit fits that converge on a solution that is not a hard limit of the fit as a function of total magnitude. Bottom panel shows the mean Euclidian logarithmic distance between the six input parameters and the ProFit fit for the galaxies that have converged fits taken from the top panel (where the distance in flux is $\log_{10}(\text{flux})$ rather than magnitude, i.e. $m/2.5$). The solid lines show the median of the results, and the dotted show the 14th and 86th percentile ranges (akin to the $1\sigma$ range if the distributions are normal). In simple terms, larger value are better in the top panel and lower values are better in the bottom panel.

# REFERENCES

Abell P. A., LSST Science Collaborations, 2009, preprint (arXiv:0912.0201)
Ahn C. P. et al., 2014, ApJS, 211, 17
Allen P. D., Driver S. P., Graham A. W., Cameron E., Liske J., de Propris R., 2006, MNRAS, 371, 2
Andredakis Y. C., Peletier R. F., Balcells M., 1995, MNRAS, 275, 874
Andrews S. K., Kelvin L. S., Driver S. P., Robotham A. S. G., 2014, PASA, 31, 4
Benson A. J., Frenk C. S., Sharples R. M., 2002, ApJ, 574, 104
Benson A. J., Džanović D., Frenk C. S., Sharples R., 2007, MNRAS, 379, 841
Brewer B. J., Pártay L. B., Csányi G., 2011, Stat. Comput., 21, 4, 649
Byun Y. I., Freeman K. C., 1995, ApJ, 448, 563
Cappellari M., 2002, MNRAS, 333, 400
Cash W., 1979, ApJ, 228, 939
de Jong R. S., 1996, A&AS, 118, 557
de Souza R. E., Gadotti D. A., dos Anjos S., 2004, ApJS, 153, 411
de Vaucouleurs G., 1948, Ann. Ástrophys., 11, 247
de Vaucouleurs G., 1958, ApJ, 128, 465
de Vaucouleurs G., 1959, ApJ, 130, 728
Dressler A., 1987, ApJ, 317, 1
Driver S. P., Liske J., Cross N. J. G., De Propris R., Allen P. D., 2005, MNRAS, 360, 81
Driver S. P., Allen P. D., Liske J., Graham A. W., 2007, ApJ, 657, 85
Driver S. P. et al., 2011, MNRAS, 413, 971

Driver S. P., Robotham A. S. G., Bland-Hawthorn J., Brown M., Hopkins A., Liske J., Phillipps S., Wilkins S., 2013, MNRAS, 430, 2622
Driver S. P., Davies L. J., Meyer M., Power C., Robotham A. S. G., Baldry I. K., Liske J., Norberg P., 2016, Astrophys. Space Sci. Proc., Vol. 42, The Universe of Digital Sky Surveys. Springer International Publishing, Switzerland, p. 205
Erwin P., 2015, ApJ, 799, 226
Erwin P., Pohlen M., Beckman J. E., 2008, AJ, 135, 20
Ferguson H. C., Binggeli B., 1994, A&AR, 6, 67
Fletcher R., 1970, Comput. J., 13, 317
Freeman K. C., 1970, ApJ, 160, 811
Gadotti D. A., 2009, MNRAS, 393, 1531
Graham A. W., 2013, in Oswalt T. D., Keel W. C., eds, A Review of Elliptical and Disc Galaxy Structure and Modern Scaling Laws. Springer Publishing, Berlin, p. 91–140
Graham A. W., Driver S. P., 2005, PASA, 22, 118
Graham A. W., Erwin P., Trujillo I., Asensio Ramos A., 2003, AJ, 125, 2951
Häußler B. et al., 2007, ApJS, 172, 615
Häußler B. et al., 2013, MNRAS, 430, 330
Hubble E. P., 1926, ApJ, 64, 321
Hyde J. B., 2009, PhD thesis
Hyde J. B., Bernardi M., 2009, MNRAS, 394, 1978
Kelvin L. S. et al., 2012, MNRAS, 421, 1007
Kelvin L. S. et al., 2014, MNRAS, 439, 1245
Kennedy R. et al., 2015, MNRAS, 454, 806
Kennedy R. et al., 2016, MNRAS, 460, 3458
Kent S. M., 1985, ApJS, 59, 115
King I., 1962, AJ, 71, 64
King I. R., 1966, AJ, 67, 471
Kormendy J., 1977, ApJ, 217, 406
Kormendy J., Kennicutt R. C., 2004, ARA&A, 42, 603
Kuijken K. et al., 2015, MNRAS, 454, 3500
Lacey C. G et al., 2016, MNRAS, 462, 3854
Lange R. et al., 2015, MNRAS, 447, 2603
Lange R. et al., 2016, MNRAS, 462, 1470
Lauer T. R. et al., 1995, AJ, 110, 2622
Laurikainen E., Salo H., Buta R., 2005, 362, 1319
Liske J. et al., 2015, MNRAS, 452, 2087
Marquardt D., 1963, J. Soc. Indust. Appl. Math., 11, 431
Meert A., Vikram V., Bernardi M., 2015, MNRAS, 446, 3943
Miyazaki S., HSC Collaboration, 2013, MNRAS, 444, 147
Moffat A. F. J., 1969, A&A, 3, 455
Moffett A. J. et al., 2016a, MNRAS, 457, 1308
Moffett A. J. et al., 2016b, MNRAS, 462, 4336
Peng C. Y., Ho L. C., Impey C. D., Rix H.-W., 2010, AJ, 139, 2097
Pignatelli E., Fasano G., Cassata P., 2006, A&A, 446, 373
Pritchet C., Kline M. I., 1981, AJ, 86, 1859
R Development Core Team, 2016, R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna
Robotham A. S. G., 2016, Astrophysics Source Code Library, record ascl:1604.004
Robotham A. S. G., Obreschkow D., 2015, PASA, 32, 33
Savorgnan G. A. D., Graham A. W., 2016, ApJS, 222, 10
Schade D., Lilly S. J., Crampton D., Hammer F., Le Fevre O., Tresse L., 1995, ApJ, 451, L1
Schaye J. et al., 2015, MNRAS, 446, 521
Sérsic J. L., 1963, Bol. Asociacion Argentina Astron. La Plata Argentina, 6, 41
Shen S., Mo H. J., White S. D. M., Blanton M. R., Kauffmann G., Voges W., Brinkmann J., Csabai I., 2003, MNRAS, 343, 978
Simard L. et al., 2002, ApJS, 142, 1
Simard L., Mendel J. T., Patton D. R., Ellison S. L., McConnachie A. W., 2011, ApJS, 196, 11
The Dark Energy Survey Collaboration, 2005, preprint (astro-ph/0510346)
Vogelsberger et al., 2014, MNRAS, 444, 1518
Wadadekar Y., Robbason B., Kembhavi A., 1999, AJ, 117, 1219
Wright A. H., 2016, MNRAS, 460, 765
Zavala J., Okamoto T., Frenk C. S., 2008, MNRAS, 387, 364

# APPENDIX A: COMPARISONS OF PROFIT EXAMPLE GALAXIES

To supplement Figs 18 and 20, below are the various other PROFIT, GALFIT, SDSS, and KiDS comparisons for the 10 galaxies that are included in the standard R PROFIT installation for demonstration and testing purposes.
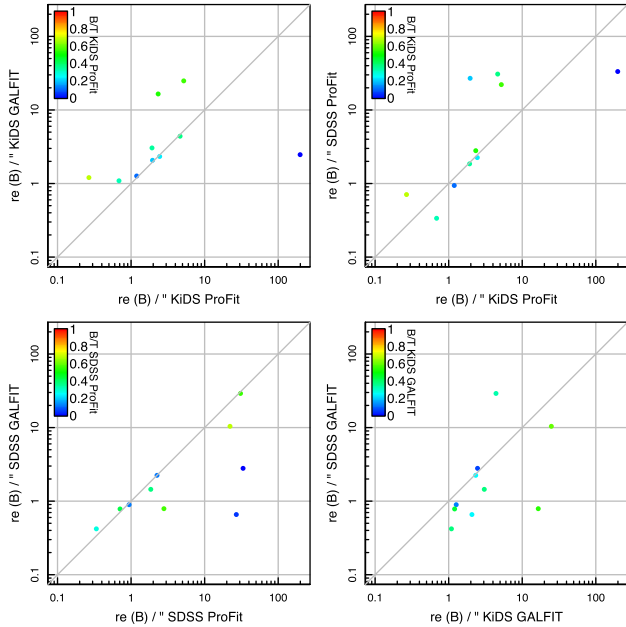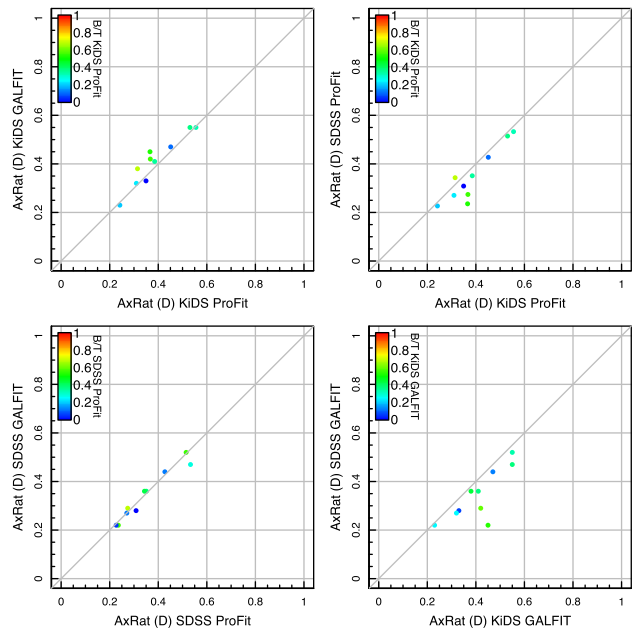


**Figure A1.** Panels comparing the measured bulge $R_e$ for the 10 well-resolved galaxies included with the R PROFIT package. See Fig. 18 for details.
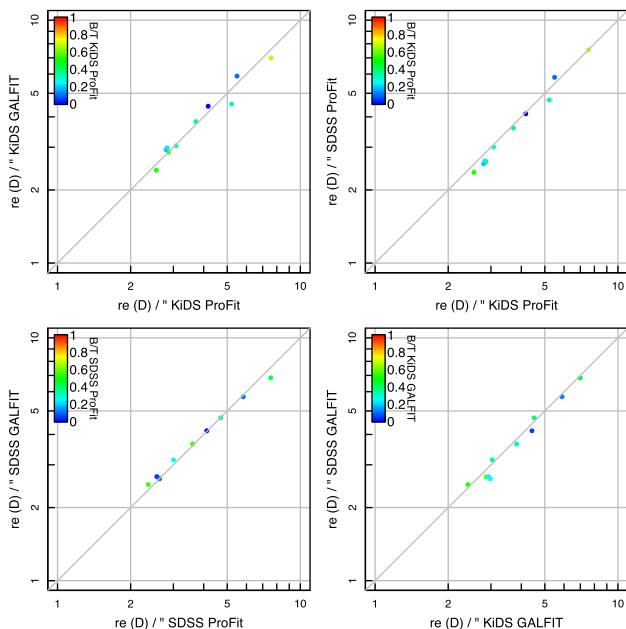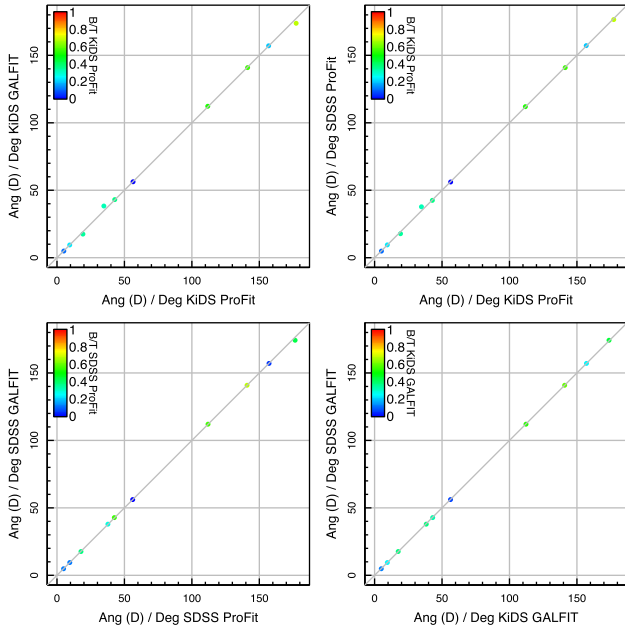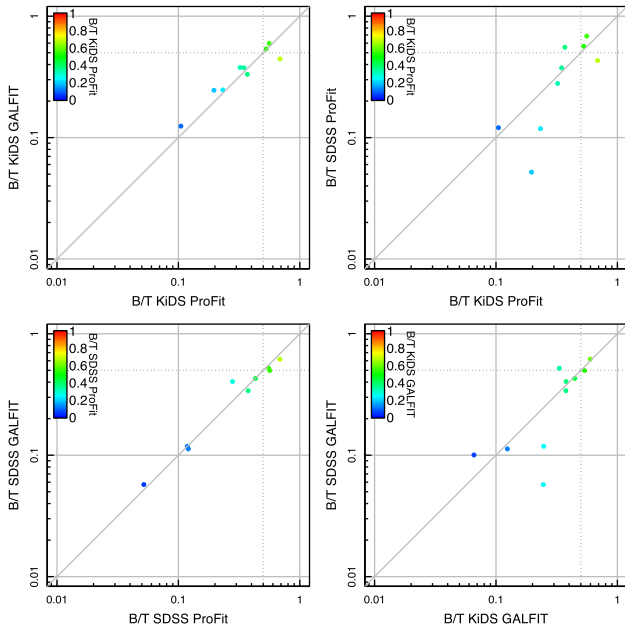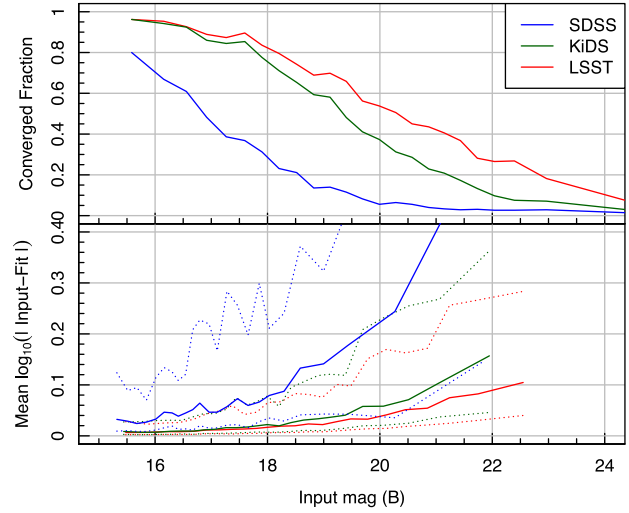


**Figure A2.** Panels comparing the measured disc $R_e$ for the 10 well-resolved galaxies included with the R PROFIT package. See Fig. 18 for details.
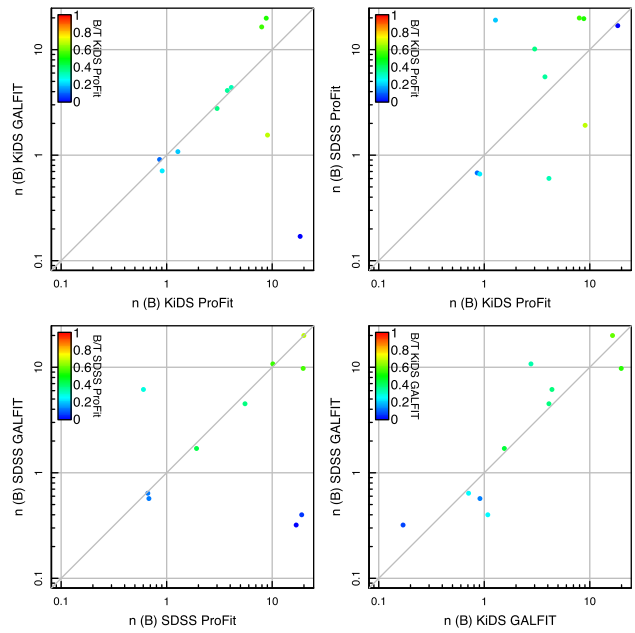


**Figure A3.** Figures comparing the measured bulge $n$ for the 10 well-resolved galaxies included with the R PROFIT package. See Fig. 18 for details.



**Figure A4.** Figures comparing the measured disc $A/B$ for the 10 well-resolved galaxies included with the R PROFIT package. See Fig. 18 for details.

**Figure A5.** Figures comparing the measured disc $\theta$ for the 10 well-resolved galaxies included with the R PROFIT package. See Fig. 18 for details.
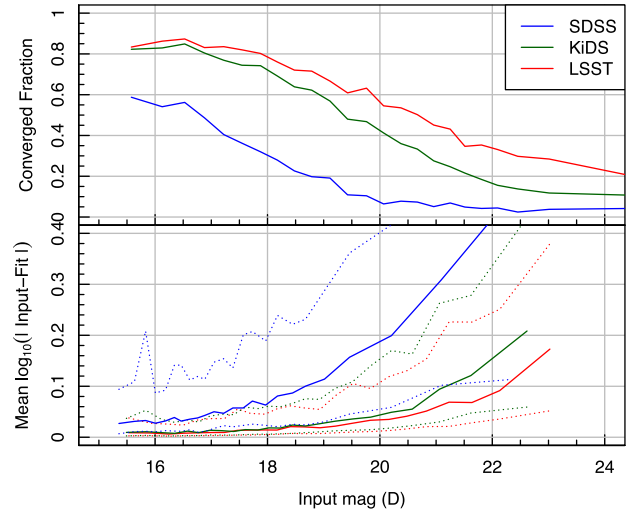
# APPENDIX B: COMPARISONS OF PROFIT SIMULATIONS FOR SDSS, KIDS, AND LSST TYPE DATA



**Figure B1.** As per Fig. 21 but for bulge magnitude on the *x*-axis.



**Figure A6.** Figures comparing the measured $B/T$ for the 10 well-resolved galaxies included with the R PROFIT package. See Fig. 18 for details.



**Figure B2.** As per Fig. 21 but for disc magnitude on the *x*-axis.
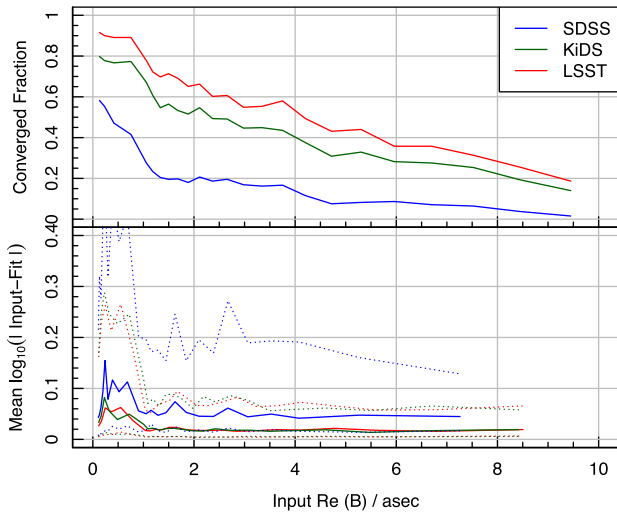
**Figure B3.** As per Fig. 21 but for bulge $R_e$ on the *x*-axis.
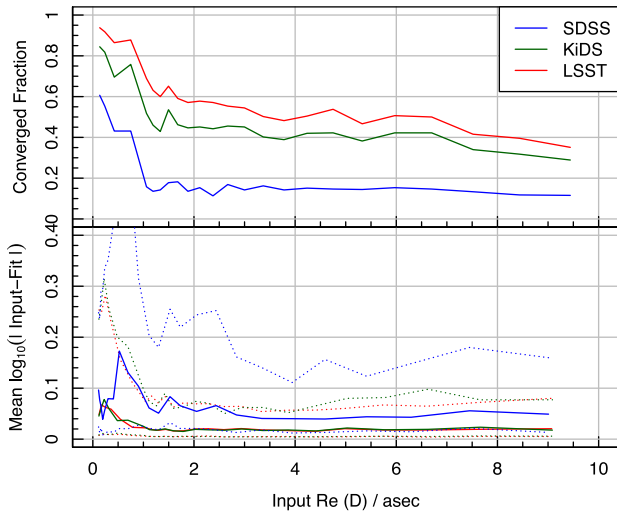


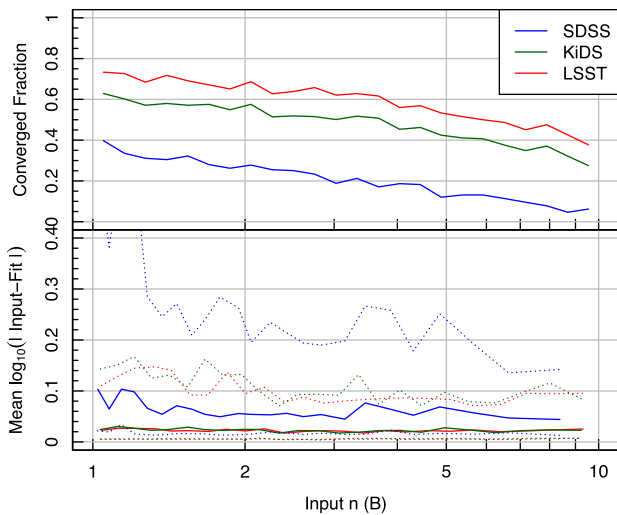**Figure B4.** As per Fig. 21 but for disc $R_e$ on the *x*-axis.



**Figure B5.** As per Fig. 21 but for bulge $n$ on the *x*-axis.
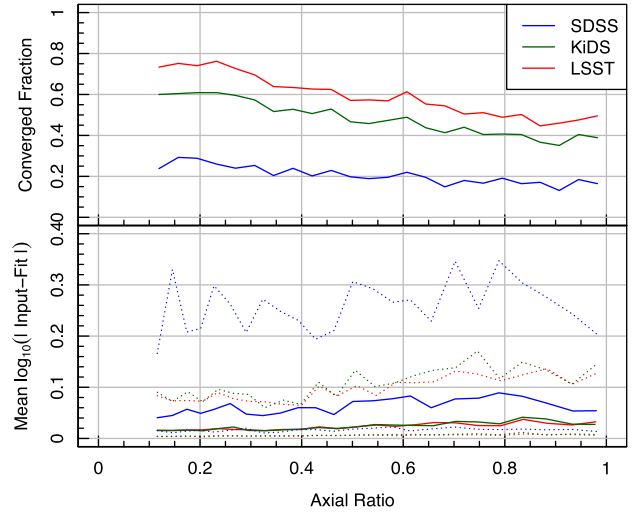


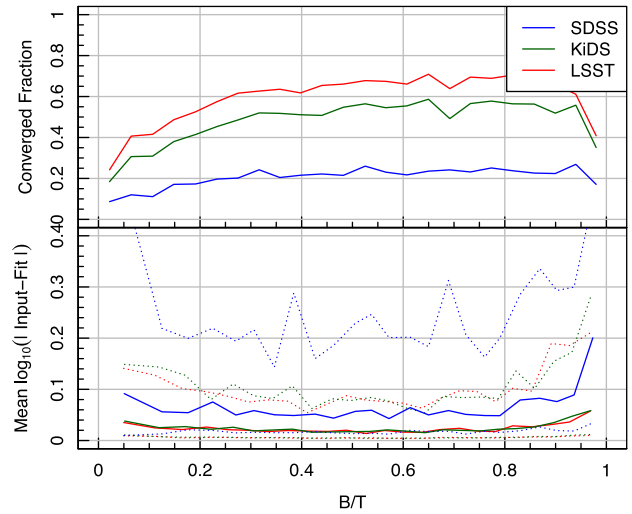**Figure B6.** As per Fig. 21 but for disc $A/B$ on the *x*-axis.



**Figure B7.** As per Fig. 21 but for $B/T$ on the *x*-axis.

This paper has been typeset from a TeX/LaTeX file prepared by the author.