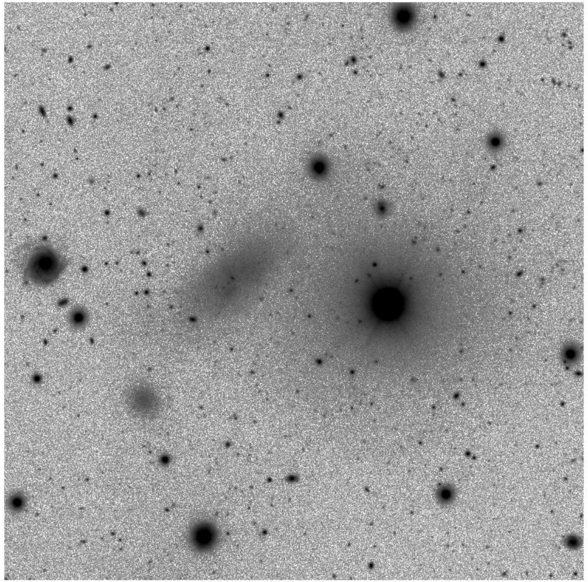
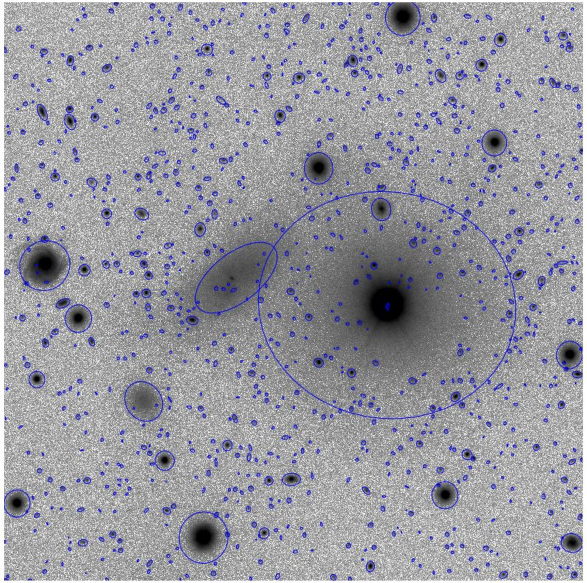
CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

Automated searches for low surface brightness (LSB) galaxies in wide area surveys



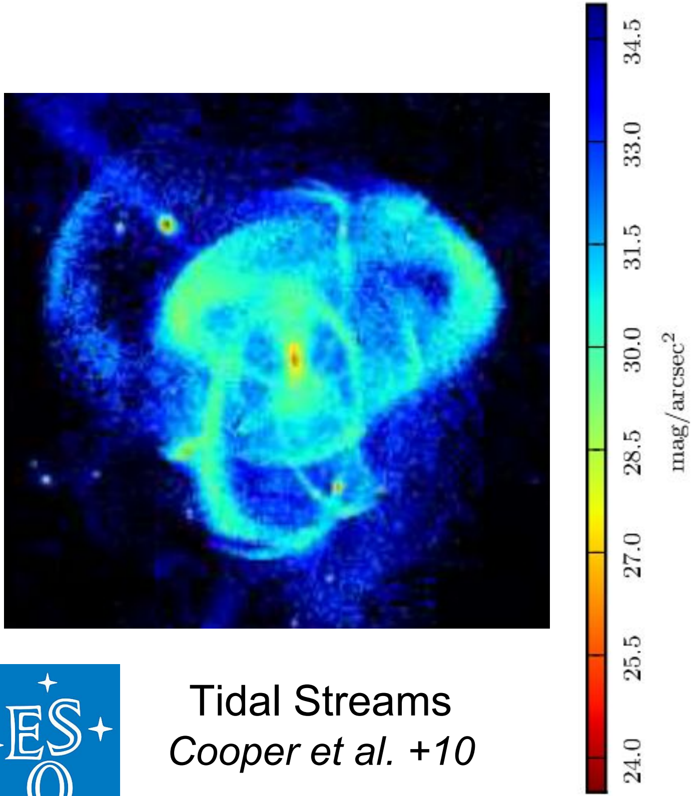


Daniel J. Prole

Supervisors: J. I. Davies (Cardiff), Michael Hilker & Remco van der Burg (ESO)

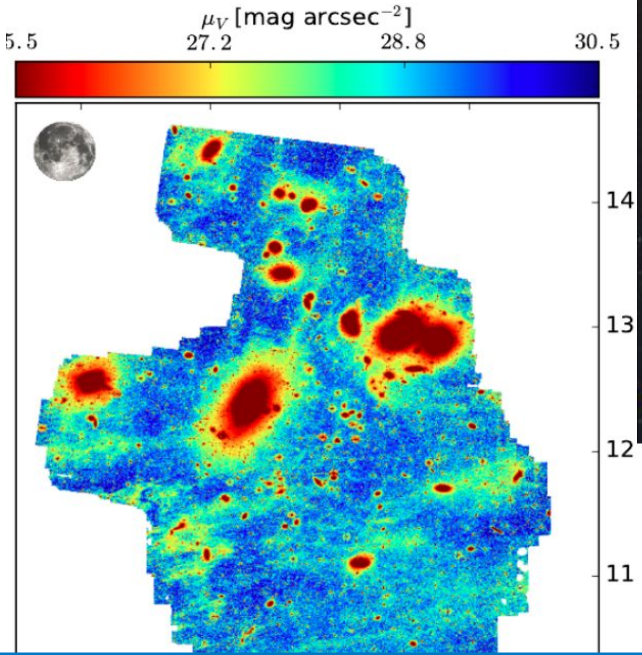
dprole@eso.org

The low surface brightness Universe



Tidal Streams
Cooper et al. +10

Intra-cluster light (ICL)
Mihos et al. 2017



LSB Galaxies
van Dokkum et al. 2018

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Ultra-diffuse galaxies (UDGs)

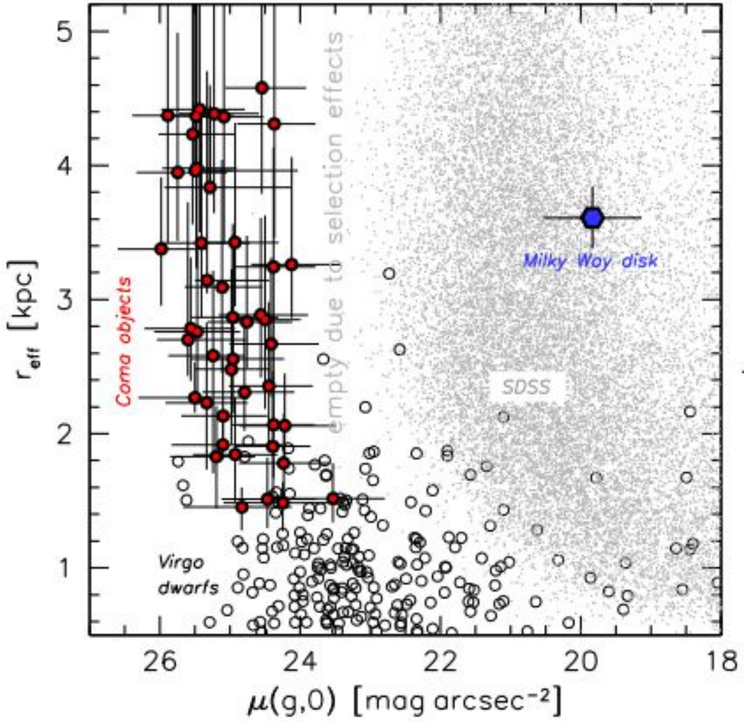
Milky way sized (effective radii > 1.5 kpc)

Stellar masses more like dwarfs ($M_{*} \sim 10^{7}\, M_{\odot}$)

Renewed interest after detection of high abundance in Coma cluster

Formation mechanisms:

- “Failed L*” galaxies (van Dokkum et al. +15)
- High-spin dwarfs (Amorisco & Loeb +16)
- Tidal formation (Carleton et al. +18)



van Dokkum et al. +15

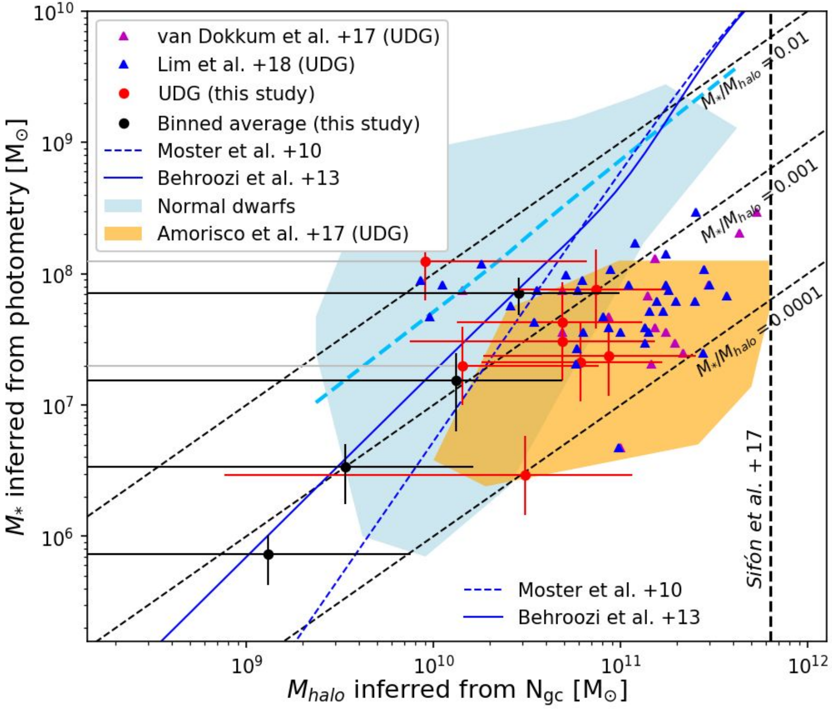
ESO

UDGs: Properties

Abundance as a function of environment tells us something about the nature of UDGs.

Key observations:

- Surprising abundance in dense environments
- Halo mass similar to dwarf galaxies
- Most cluster UDGs are red / quiescent
- Formation efficiency increases with group mass
- **Little is known about field population...**



Prole et al. *(2018, in prep.)*

Science goals & data

LSB galaxies in clusters	Fornax (FDS; Iodice et al. 2016) Hydra (VEGAS; Capaccioli et al. 2015) Virgo (NGVS; Ferrarese et al. 2012)	Size / mass distributions Nucleation fractions **Globular cluster populations***
UDGs in the field	KiDS (Kuijken et al. 2015) + GAMA (Driver et al. 2011)	Abundance & formation efficiency

Prole et al. (2018, in prep)

ESO Studentship until February 2019

dprole@eso.org

UDG field abundance with KiDS

How efficiently do UDGs form in the field?

Want to constrain UDG abundance in low-density environments...

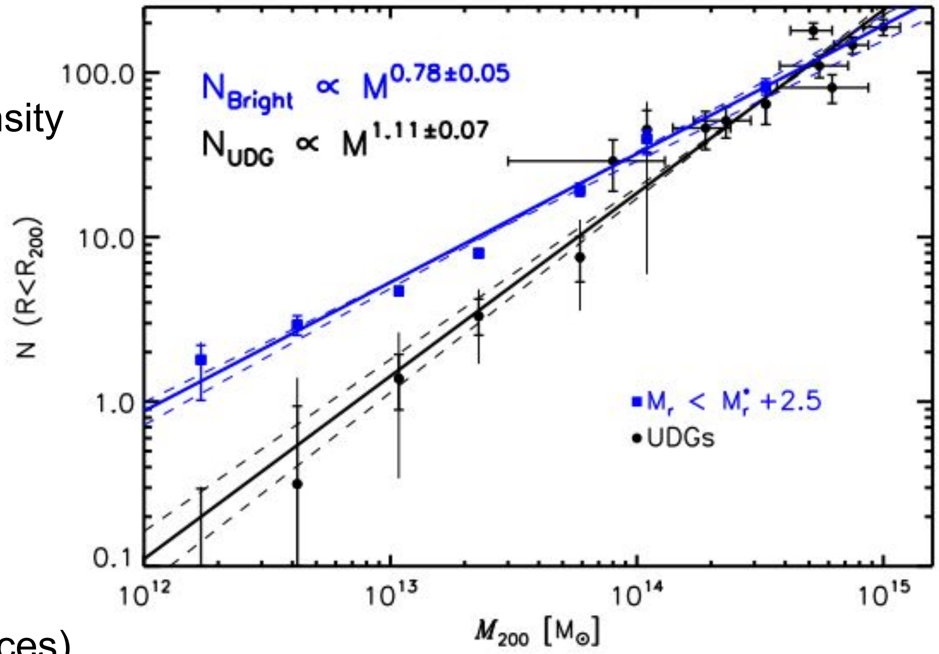
→ Need large sample of field sources

Data:

KiDS r-band, GAMA group catalogue
~250 square degrees

Measurements:

Sersic profile fits (+ nucleus)
Recovery efficiency (from synthetic sources)



van der Burg et al. (2017)

Detection & segmentation software

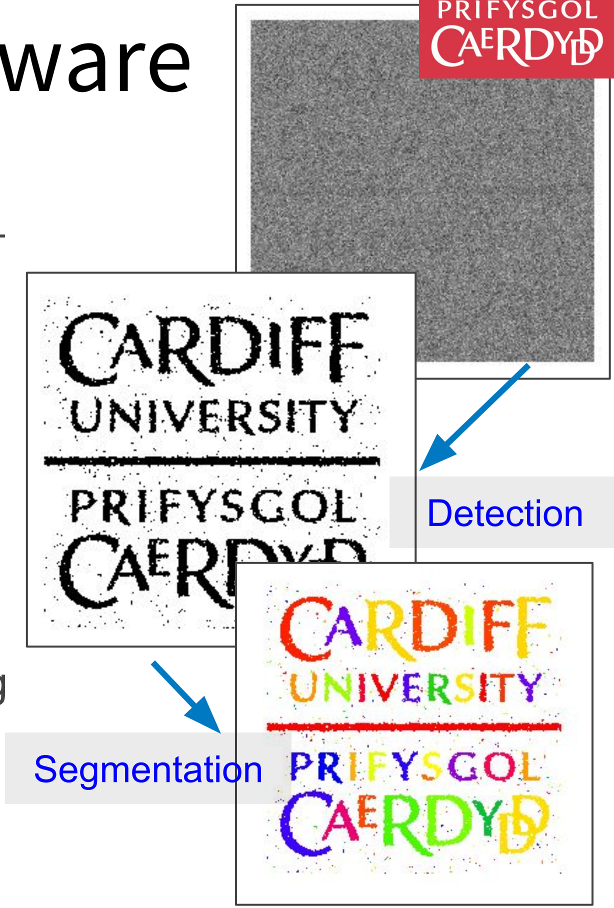
Source Extractor (Bertin & Arnouts 1996)

NoiseChisel (Akhlaghi & Ichikawa 2015)
-Non-parametric, noise based detection / segmentation

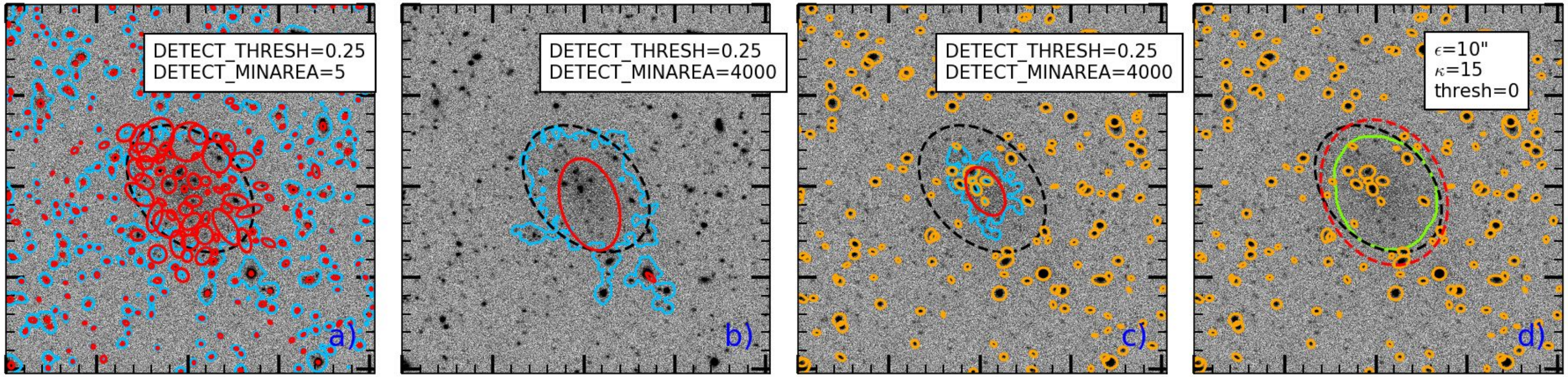
DeepScan (Prole et al. 2018)
-Density-based detection of extended LSB structure

MTObjects (Teeninga et al. 2016)
-Continuous threshold + max-tree from attribute filtering

ProFound (Robotham et al. 2018)
-Watershed deblend + iterative segment dilation



Detection software: SExtractor



Fragmentation ("shredding") | Confusion | Parameter Underestimation | DeepScan

Blue: SExtractor segmentation map
Unbroken red: SExtractor effective radius (FLUX_RADIUS)
Orange: Masked

Detection software: DeepScan

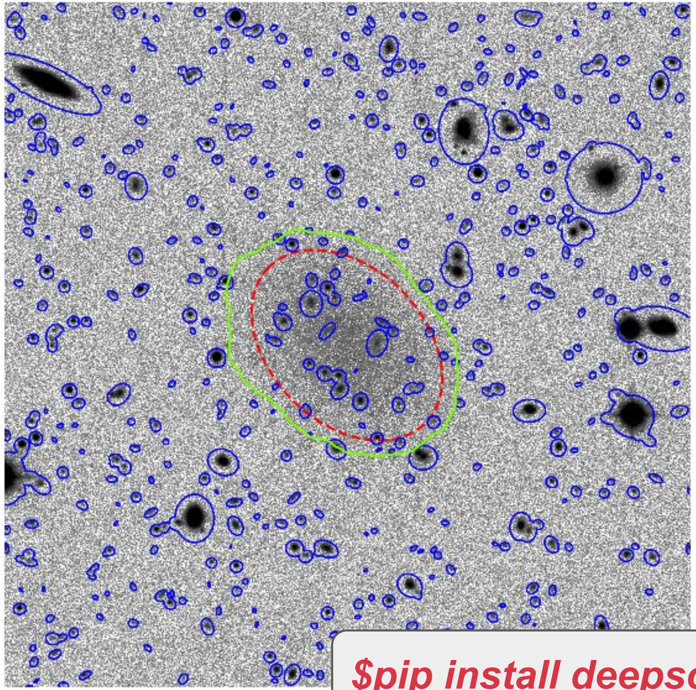
- **Sky measurement:**
  - Mesh grid + interpolation
  - Iterative pixel masking with DBSCAN
  - Custom estimators

Lower bias vs. SExtractor

- **Source masking** (optional):
  - Python interface to SExtractor

Automatic mask creation

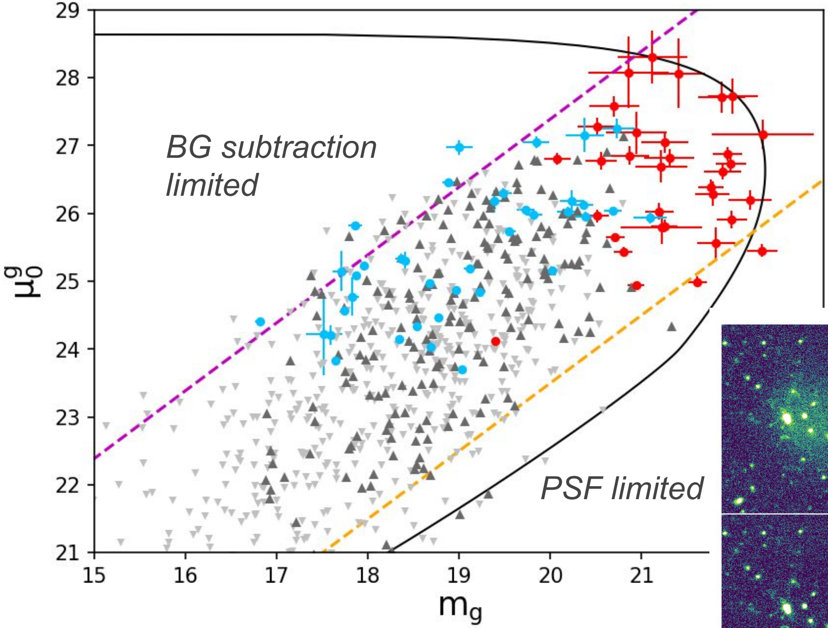
- **DBSCAN pixel clustering**
  - Identifies over-densities of thresholded pixels within radius ε



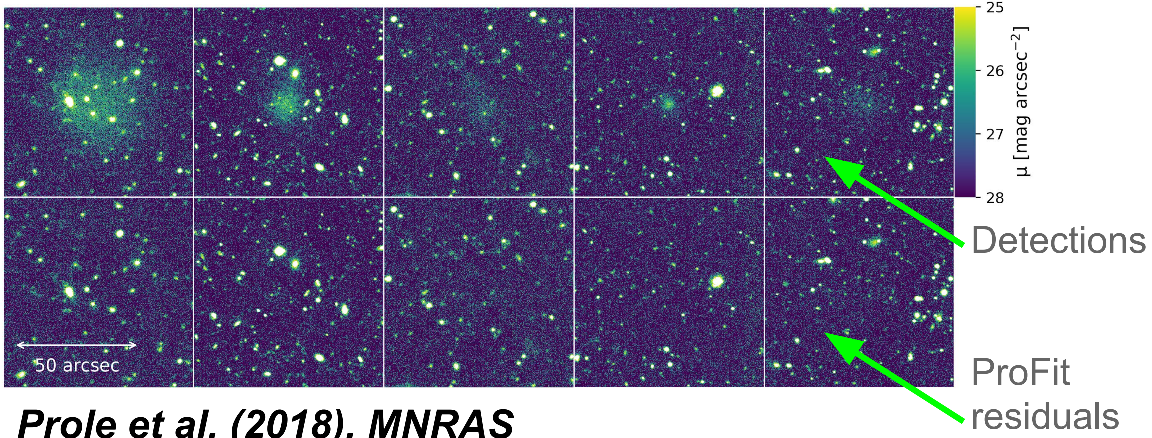
$pip install deepscan

dprole@eso.org

Detection software: DeepScan



- 30 new detections over 5 degrees^2 in public NGVS data (red points)
- Stellar masses in range 10^6-10^7 $M_\odot$
- High number of false positives



Prole et al. (2018), MNRAS

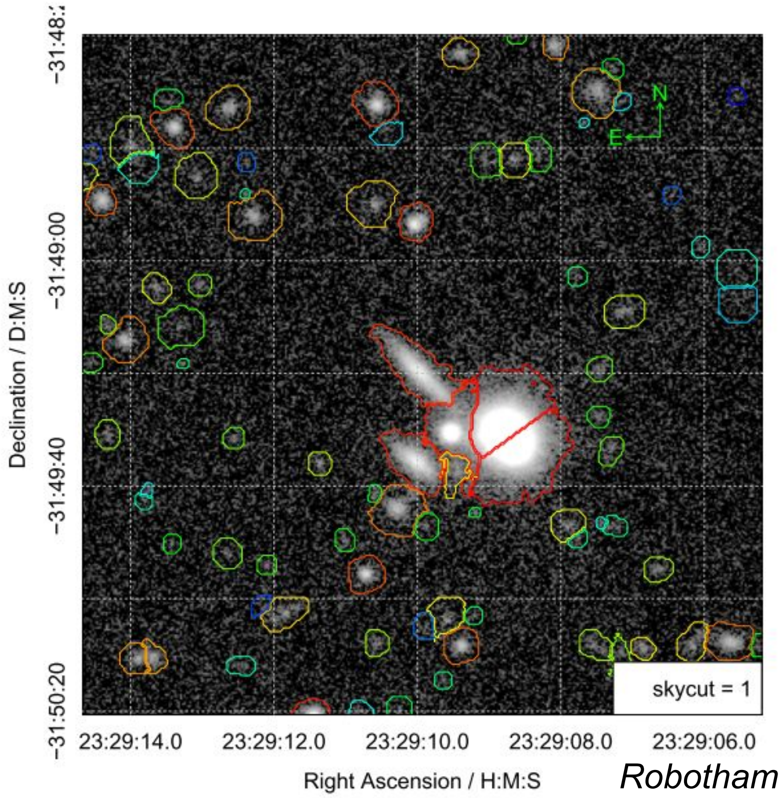
https://github.com/danjampro/DeepScan

Detection software: ProFound

- Sky estimate (mesh)
- **Watershed segment** pixels above threshold
- Measure segment stats
- **Segment dilation**
- Remeasure sky

Iterative

https://github.com/asgr/ProFound

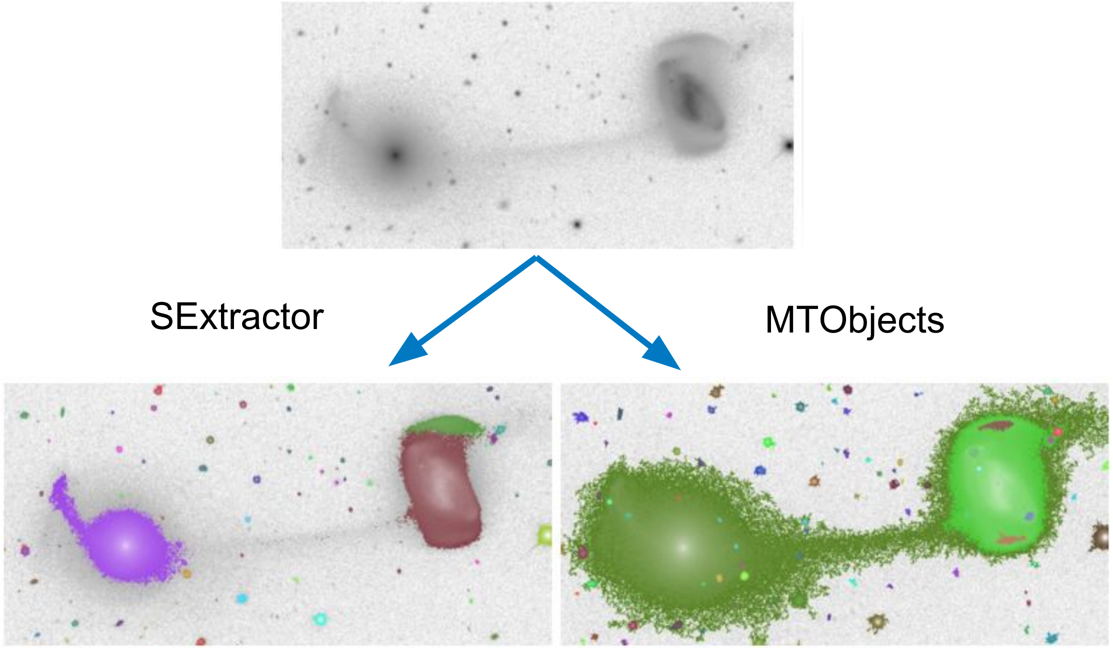


Robotham et al. (2018)

Detection software: MTObjects

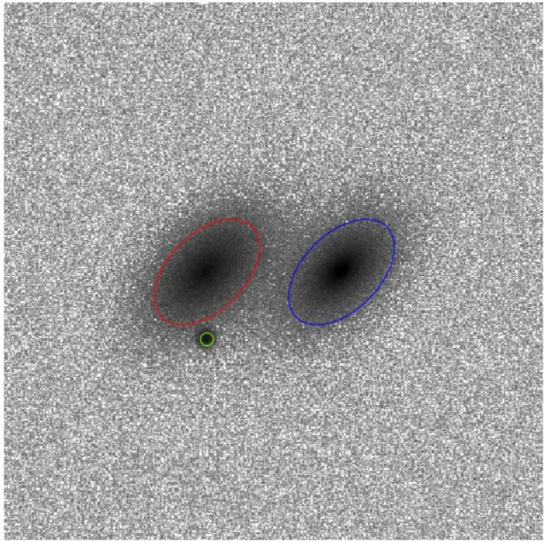
- Global sky estimate
- Hierarchical image representation
- Source identification (significant nodes)
- Segment measurement

Under development

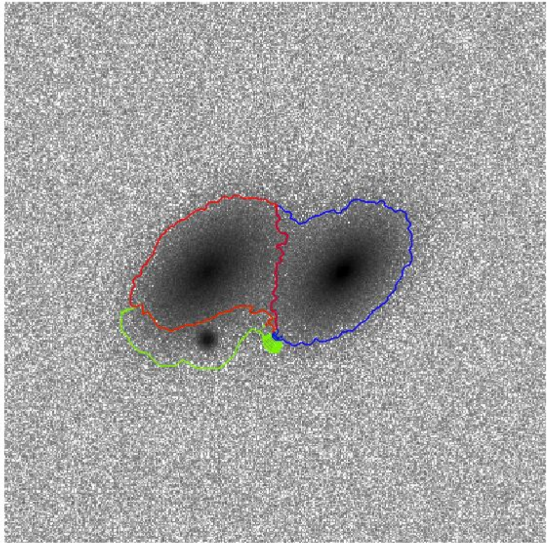


Teeninga et al. (2016)

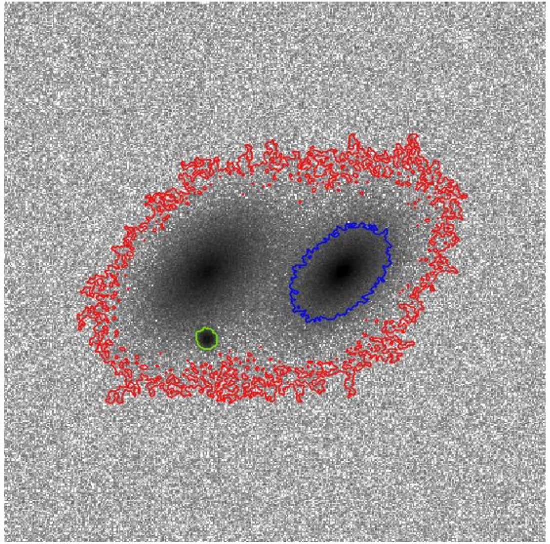
Segmentation methods: Comparison



Original



ProFound
(close to defaults)



MTObjects
(default parameters)

dprole@eso.org

Which is the best* ?

	The Good	The Bad
DeepScan	Efficient python implementation Detection of ultra-faint extended features	Confusion in crowded fields **No segmentation / source nesting**
ProFound	Well developed / documented Useful segment measurements	Long runtime / memory intensive **Halo fragmentation (+no nesting)**
MTObjects	Efficient implementation **Identification of nested LSB sources**	LSB halo confusion Measurement uncertainty in crowded fields

*Quantative analysis in prep.

Measuring LSB galaxies (KiDS)

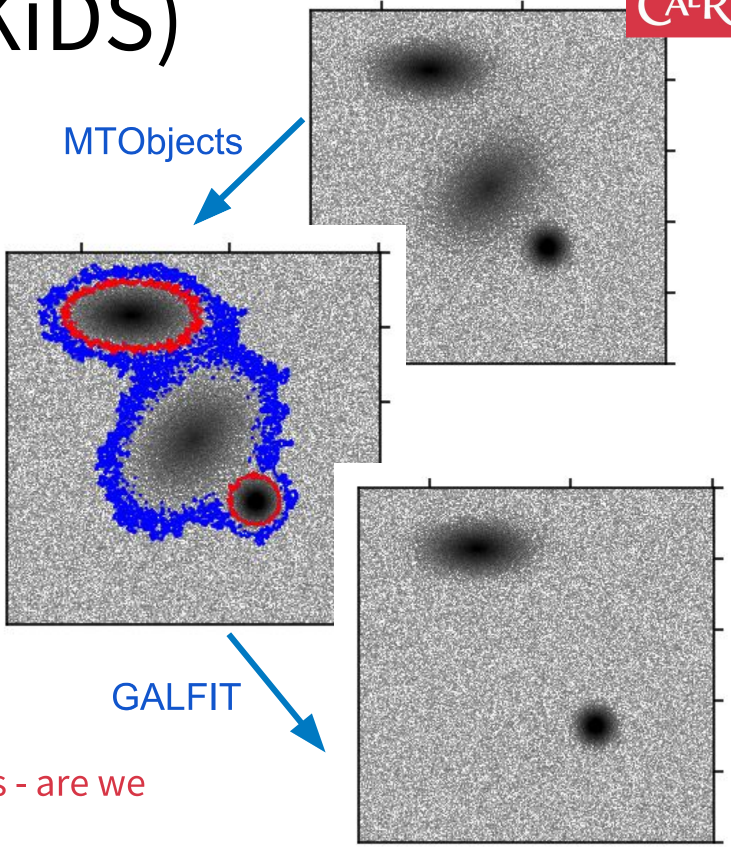
Automatic detection / measurement pipeline:

Run MTObjects

Preselect segments based on MTO size and SB

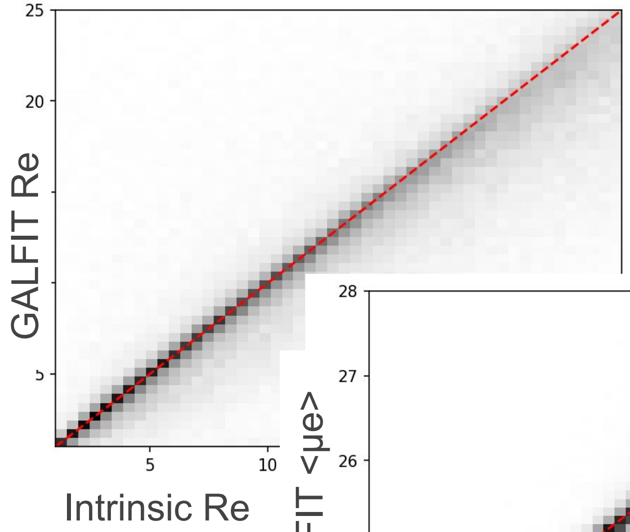
Run GALFIT on preselected sources
→ Mask all other segments
→ Fit combined Sersic + sky model
→ Fit combined Sersic + sky + nuclear PSF

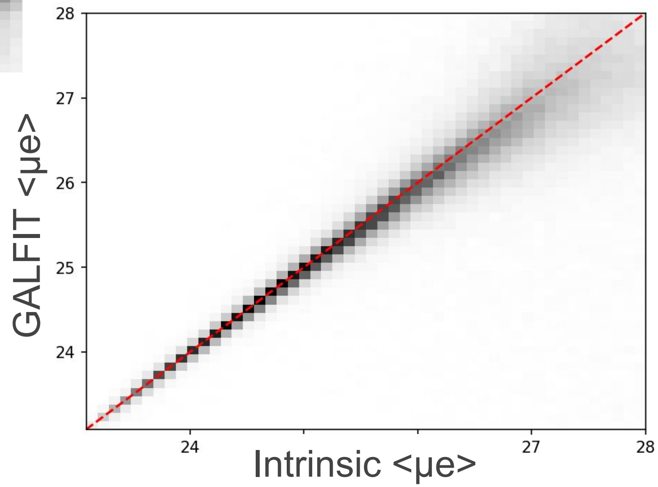
Select final sources from GALFIT models

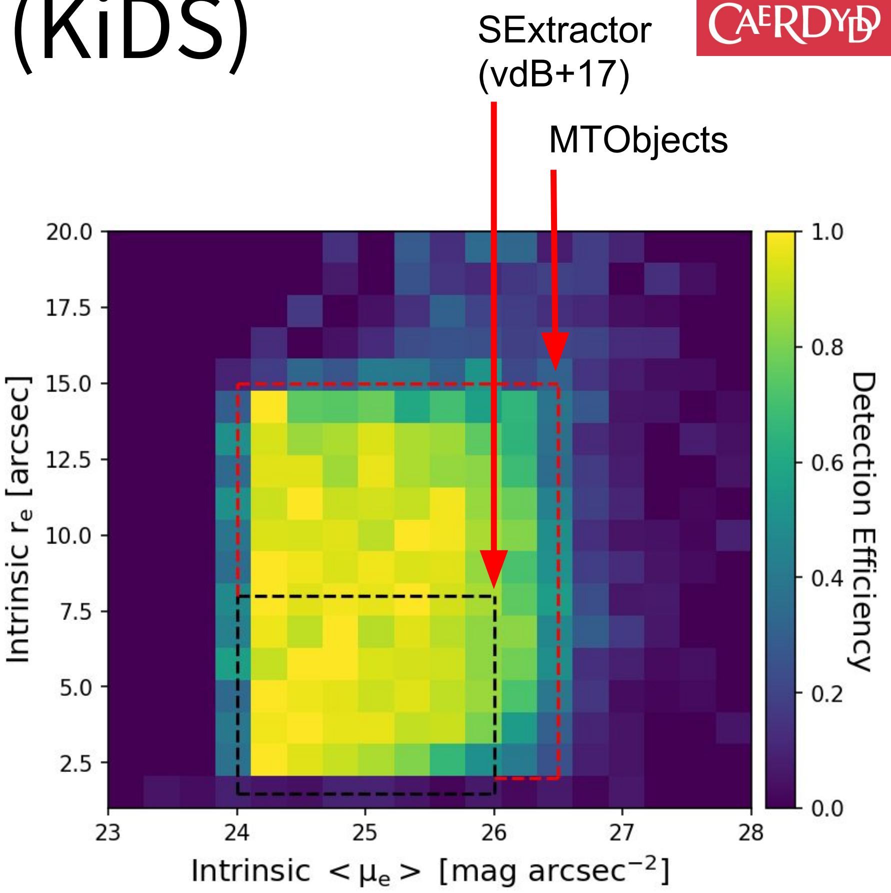


Recovery efficiency much worse for nucleated sources - are we systematically missing nucleated LSBs?

Measuring LSB galaxies (KiDS)





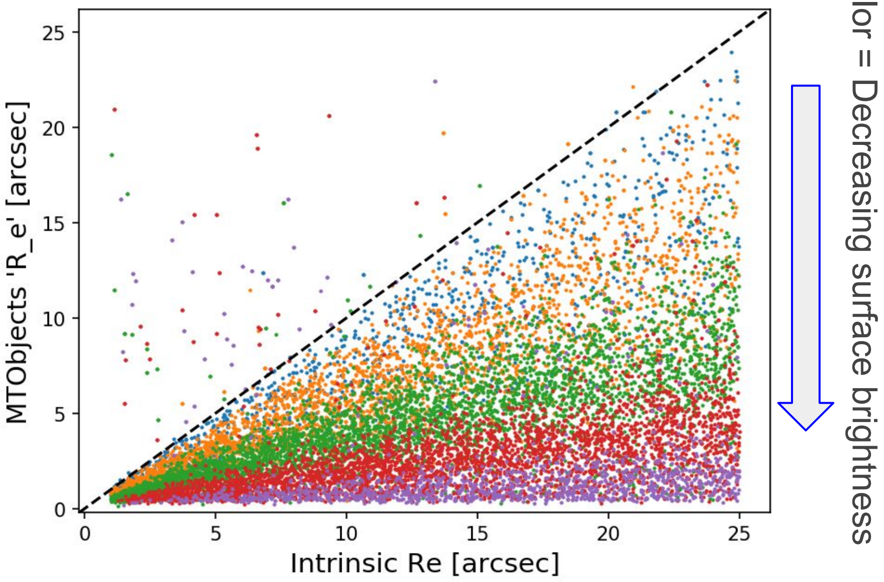


dprole@eso.org

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

Measuring synthetic LSB galaxies (KiDS)

KiDS + MTObjects



Biased parameters from segment statistics (unavoidable)

dprole@eso.org

Measuring LSB galaxies… with ML?

ML can offer several advantages over standard galaxy fitting approaches…

- Significant speed increase
- Reduction of pre-selection bias
- Automatic recognition of nuclear point sources(?)

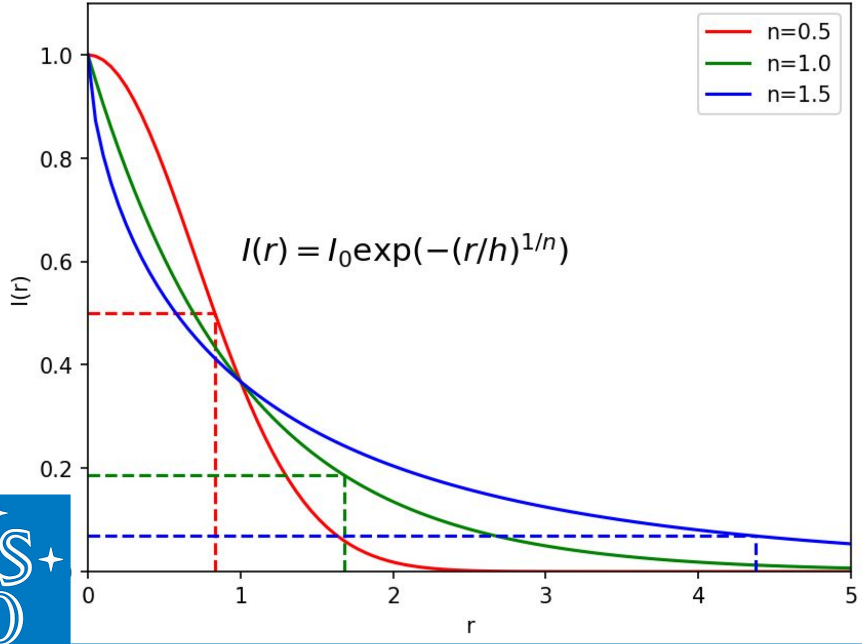
But has its own set of disadvantages…

- Training sets required (that don't exist yet)
- Need robust testing so that measurements can be trusted
- Not clear how to deal with blended sources (big issue for **LSST**)

In general ML could be used to provide initial guesses for fit parameters (at least)

Early efforts

Convolutional Neural Networks (CNNs) are well suited for image analysis...
Can we use them to measure galaxies?



Network architecture (tflearn):

Input layer (28 x 28 pixel image, relu)

Convolution layer (11x11 pixels x 32 layers, relu)

Max pool

Convolution layer (5x5 pixels x 64 layers, relu)

Fully connected layer (128 inputs, relu)

Dropout

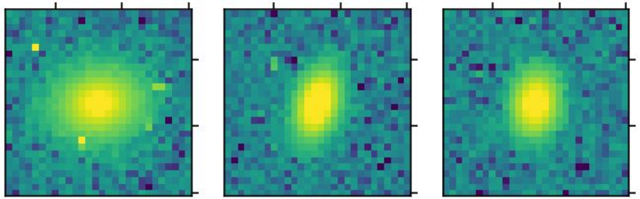
Fully connected layer (5 inputs, linear activation)

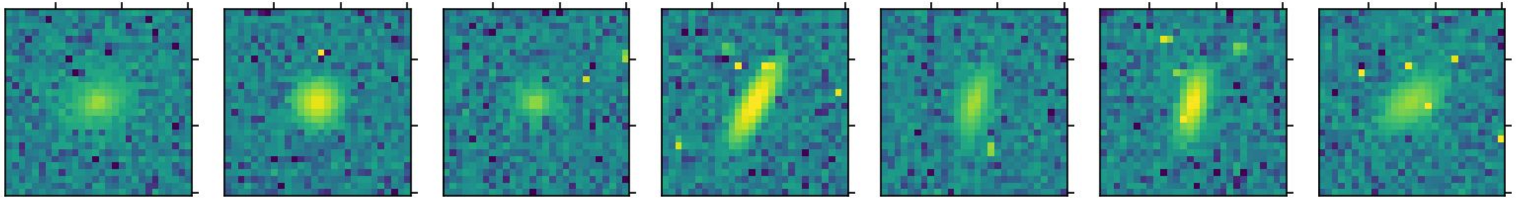
Early efforts: Training set

No sufficient training set exists for LSB galaxies (want >1E+4 samples)

For now we can use a purely synthetic dataset:

- Generate 5E+4 synthetic galaxies (28x28 pixels)
- Add noise
- Add additional components (point sources)
- Convert abs(ADU) to surface brightness and subtract RMS in SB units





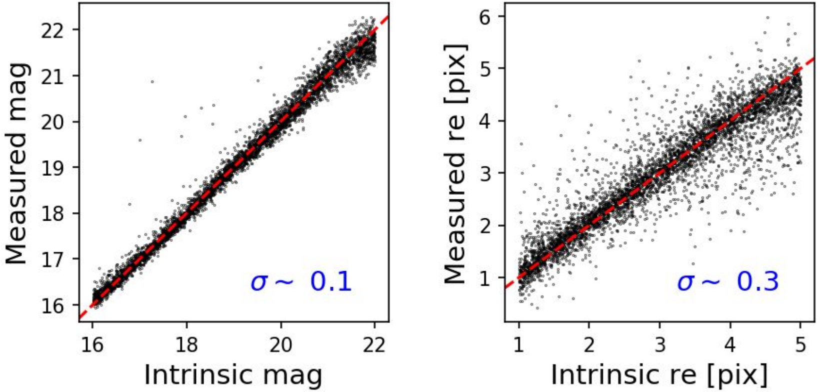
dprole@eso.org

Early efforts: Results

Trained on 18 CPUs for 100 epochs (approx ~2hrs)

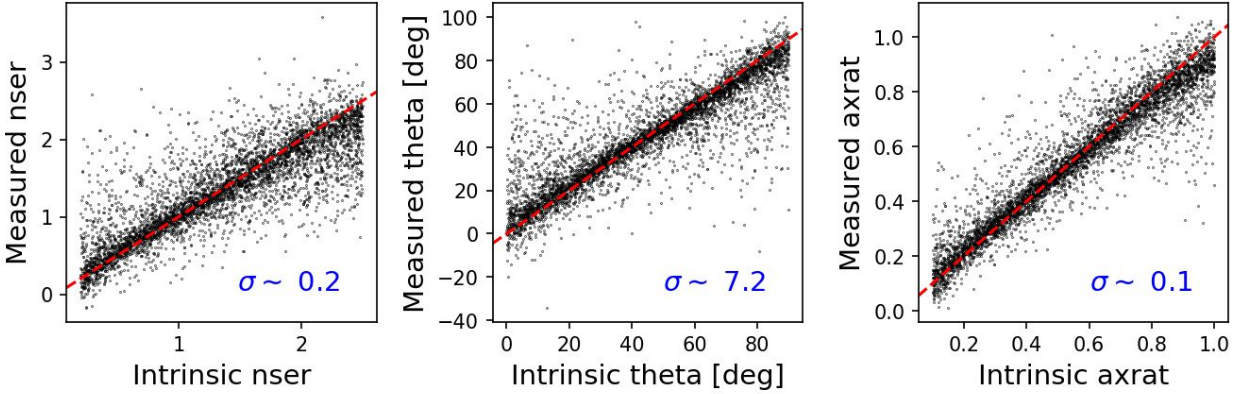
Batch size: 128

Optimiser: ADAM



~500 fits per second.

GALFIT: ~10 per second (2% failure rate)



dprole@eso.org

Early efforts: Obvious criticisms

Training sources are always centred

Sky, RMS needs to be known before

Training sources are pure Sersic profiles

Training sources are all isolated

No parameter uncertainties

No PSF

Summary & Future work

Several alternatives to SExtractor now exist...

MTObjects is favourable over SExtractor DeepScan, ProFound (NoiseChisel untested) for wide field blind surveys.

The real problem is now source measurement:

Measurements from segment statistics are biased…
Pre-selection leads to a drop in recovery efficiency at the faint end!

Early efforts using CNNs show that they might be useful for estimating Sersic parameters in the future… but more work needed!

https://github.com/danjampro/DeepScan

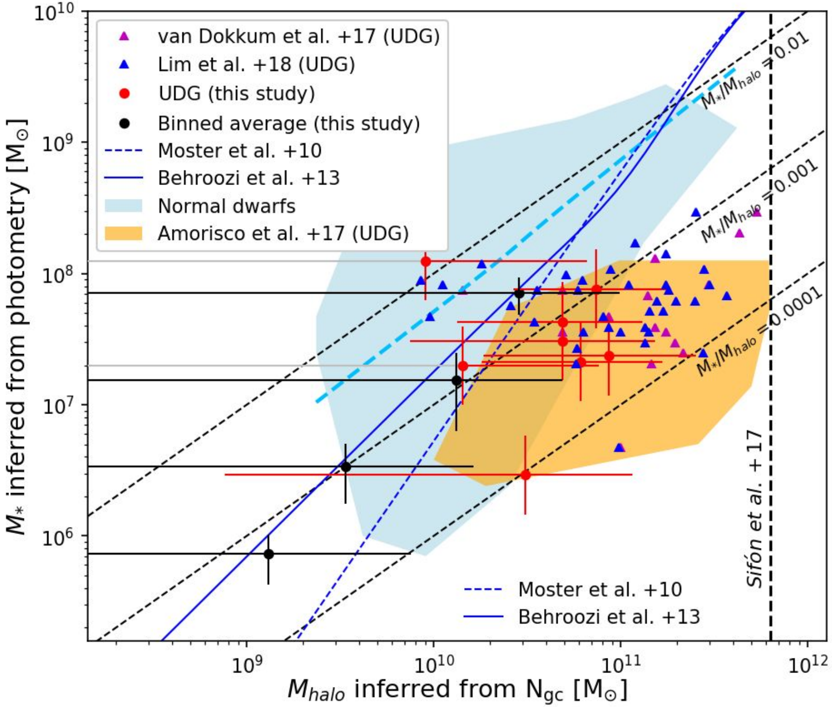
dprole@eso.org

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

UDGs: Halo mass

Several methods of measuring halo mass:

- Stellar kinematics (~30 hr integration times)
- Globular cluster kinematics
- Tidal features
- Weak lensing
- Spatial distributions
- Number of globular clusters



UDGs have dwarf sized halos that are relatively massive for their stellar mass

Prole et al. *(2018, in prep.)*

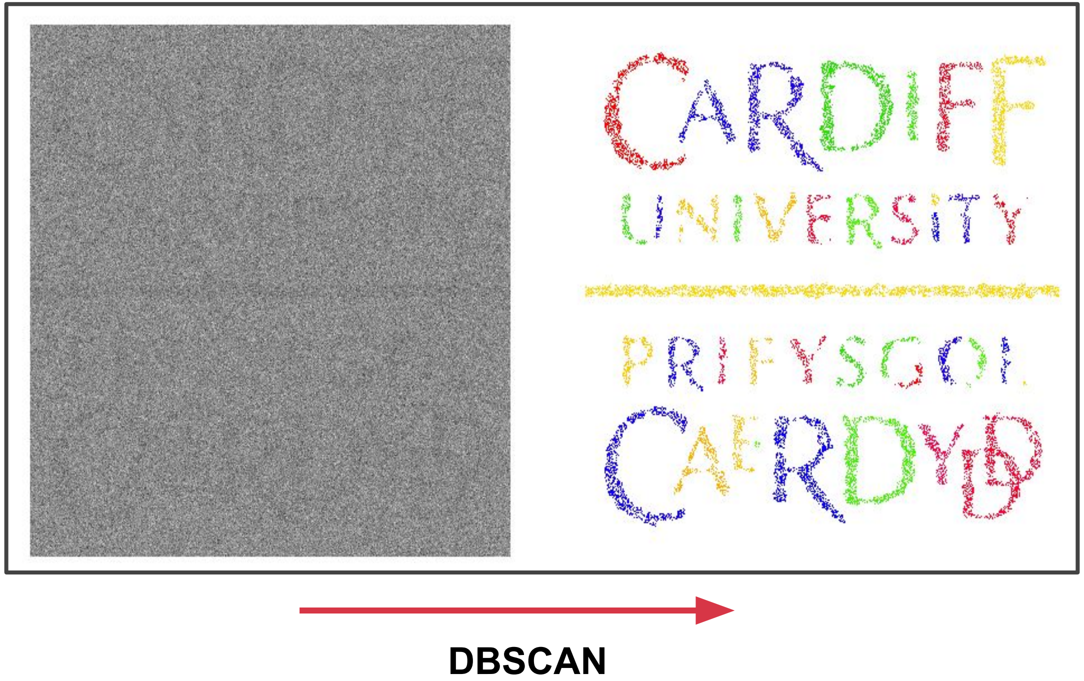
CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Detection methods: DeepScan

DeepScan is a Python package designed to identify extended LSB features (Prole et al. +18)...

At its core, using the DBSCAN algorithm (Esther et al. 1996) for detection...

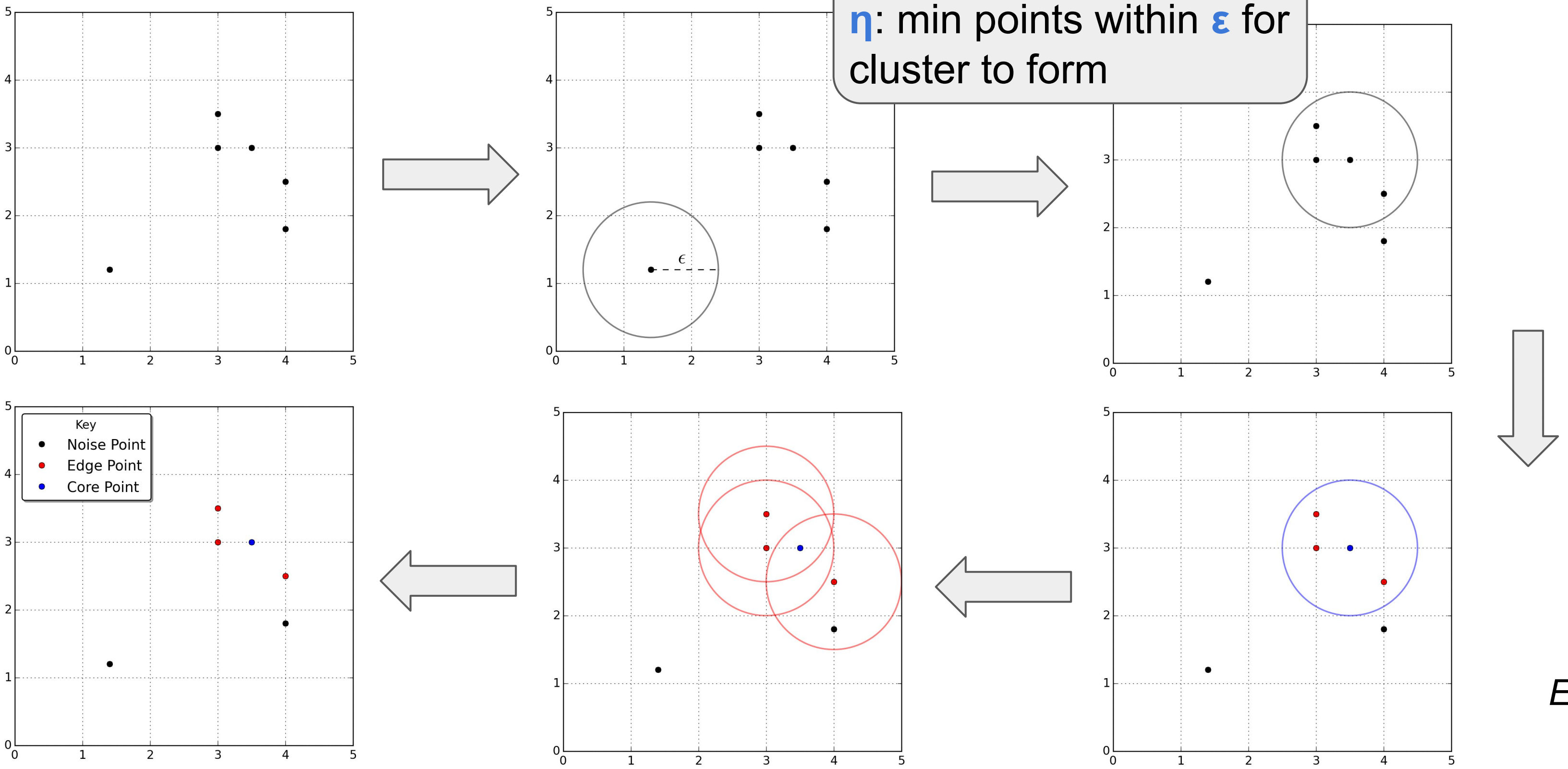
$pip install deepscan



ESO

The DBSCAN algorithm

ε: clustering scale length
η: min points within **ε** for cluster to form



ε=1
η=3

Ester et al. (1996)