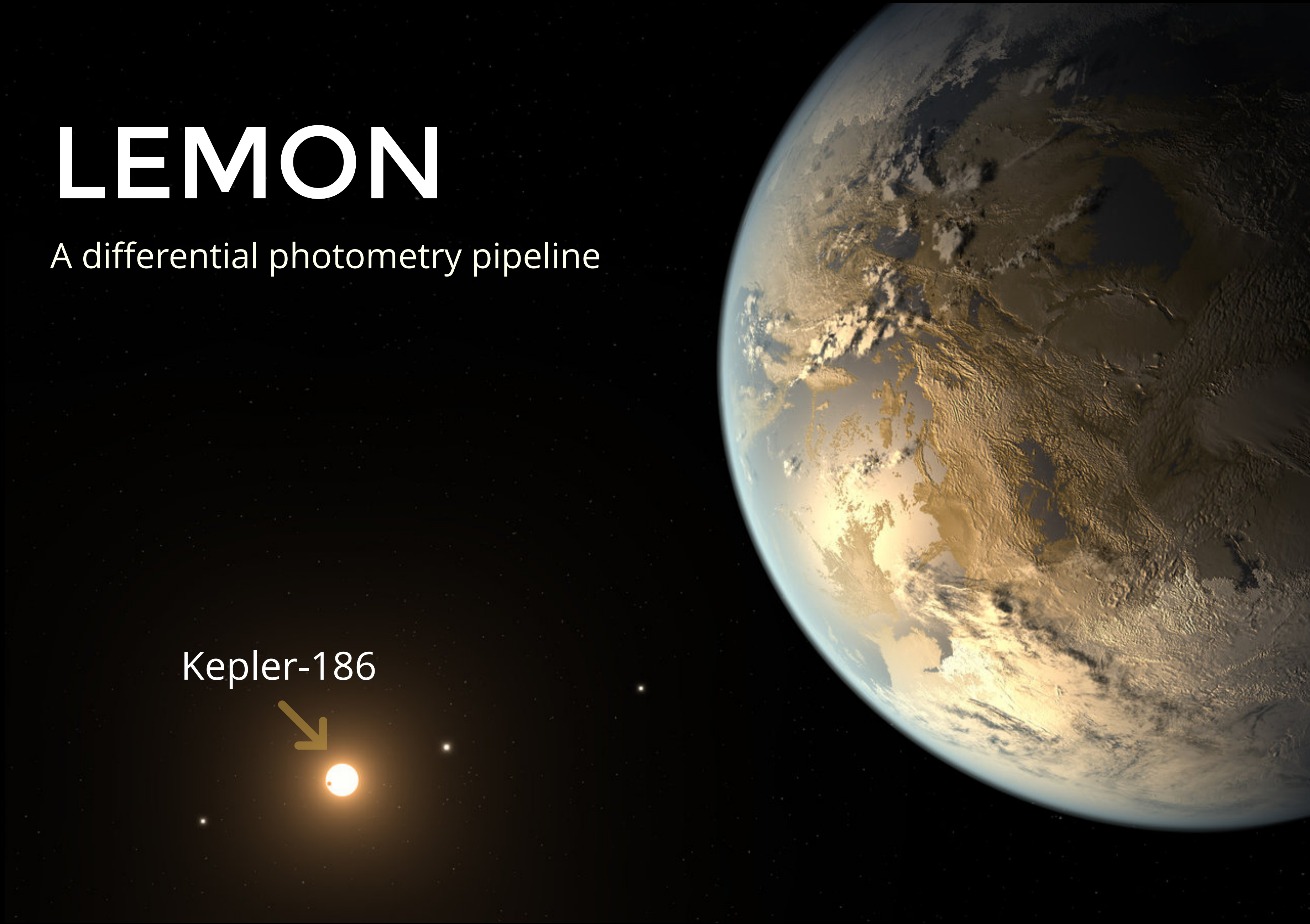
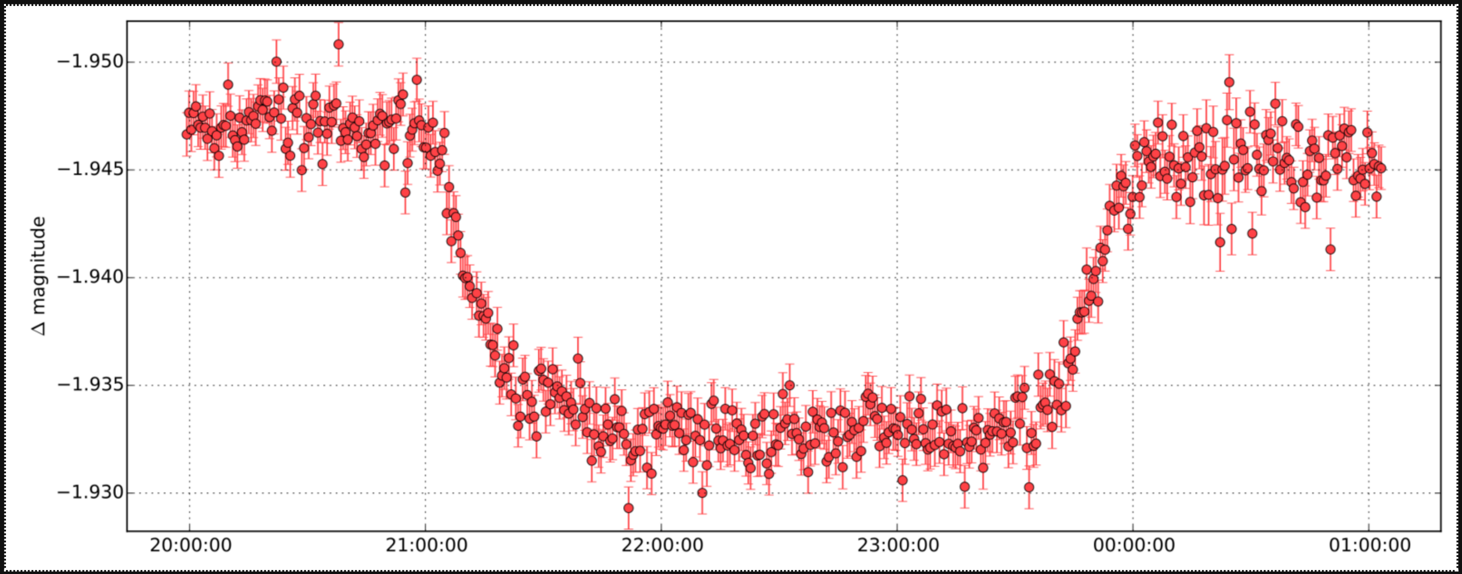
LEMON

A differential photometry pipeline



Light curves



Transit of exoplanet HAT-P-16b


My god, it's full of stars!

Introducing

LEMON

We even have a logo!

Nothing new

(admittedly)

But simple

(like — really simple)

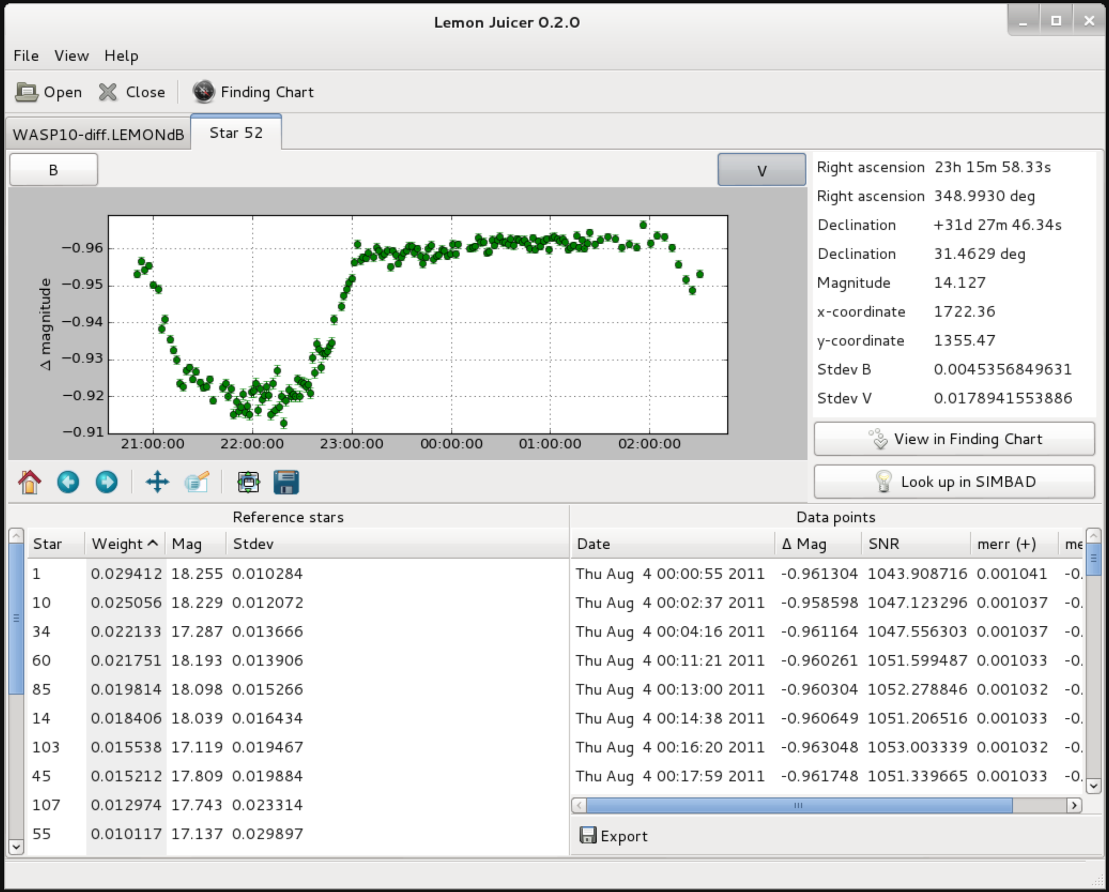
Seriously

```
$ lemon astrometry data/*.fits WASP10/
$ lemon mosaic WASP10/*.fits WASP10-mosaic.fits
$ lemon photometry WASP10-mosaic.fits WASP10/*.fits phot.LEMONdB
$ lemon diffphot phot.LEMONdB curves.LEMONdB
```

(that's it)

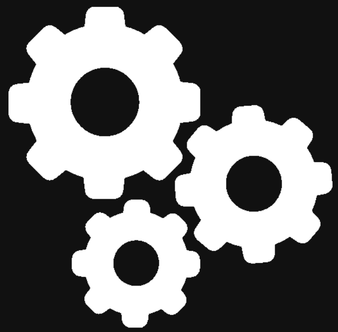
Input data, **get** curves

Analyze the results



Lots of parameters to tweak

(but the default ones just ***work***)



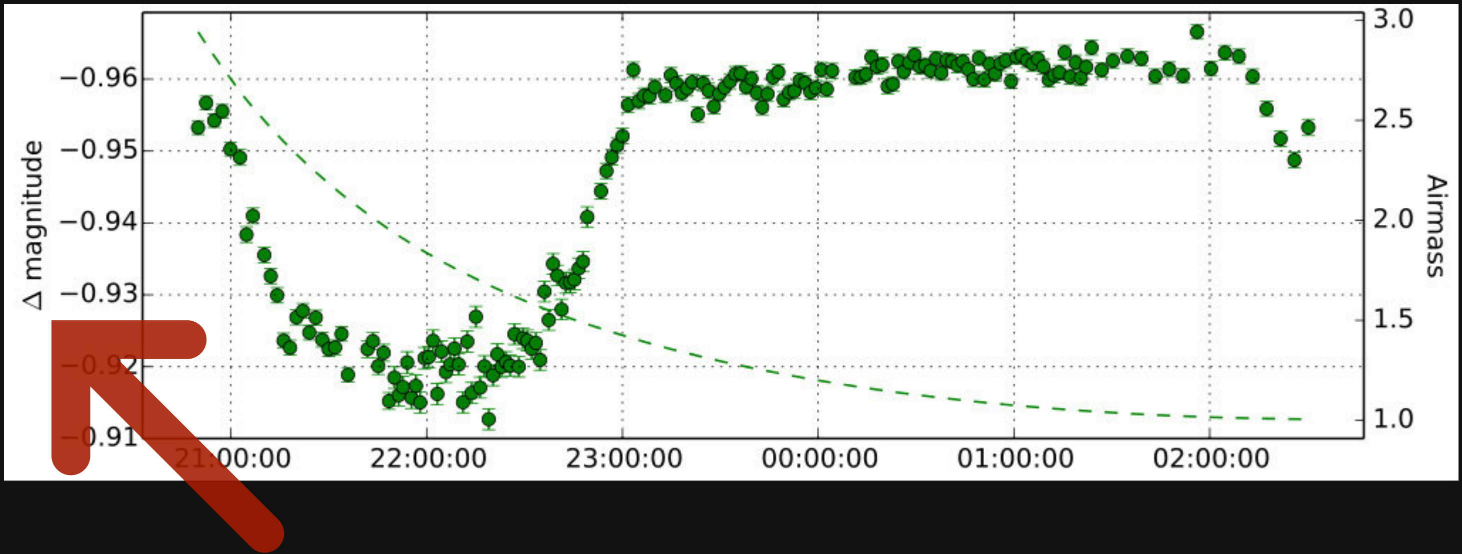
Lots of **existing** software

- Astrometry.net
- Montage
- IRAF
- SExtractor
- And **countless** Python libraries

(Shoulders of giants and all of that)

This is **differential** photometry

(*no absolute magnitudes*)



Maybe not be what you need

Multicore

(all tasks run in parallel)

This is probably an exaggeration


The Hitchhiker's Guide to LEMON
(MARS)
W A N D E R E R S
A SHORT FILM BY ERIK WERNQUIST
www.erikwernquist

Five commands

```
$ lemon
usage: lemon [--help] [--version] [--update] COMMAND [ARGS]

The essential commands are:
   astrometry   Calibrate the images astrometrically
   mosaic       Assemble the images into a mosaic
   photometry   Perform aperture photometry
   diffphot     Generate light curves
   juicer       LEMONdB browser and variability analyzer

See 'lemon COMMAND' for more information on a specific command.
```

astrometry

Find WCS solutions

FITS images need to be **astrometrically** calibrated before we can do photometry

Input FITS files

```
$ lemon astrometry data/*.fits WASP10/
```

Output directory

Find the **astrometric** solution of each image

And write new FITS files containing the WCS header

```
$ lemon astrometry data/*.fits WASP10/
>> The output directory 'WASP10' did not exist, so it had to be created.
>> Using a local build of Astrometry.net.
>> Doing astrometry on the 193 paths given as input.
>> 100%[=============================================================>]
>> You're done ^_^
```

A mere high-level interface to **Astrometry.net**

Spawn multiple **solve-field** processes

```
vterron@enzo: ~
File  Edit  View  Search  Terminal  Help
top - 18:53:01 up 151 days, 10:06, 15 users,  load average: 14.54, 11.82, 6.42
Tasks: 494 total,  19 running, 474 sleeping,   0 stopped,   1 zombie
%Cpu(s): 93.3 us,  6.7 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:  66058004 total, 65554760 used,   503244 free,   105196 buffers
KiB Swap:  2095100 total,  2095100 used,        0 free, 53482812 cached

  PID USER      PR  NI  S   %CPU  %MEM     TIME+ COMMAND
23733 vterron   20   0  R 99.865 0.203   0:35.39 solve-field
23824 vterron   20   0  R 99.865 0.195   0:37.52 solve-field
23996 vterron   20   0  R 99.865 0.173   0:03.36 solve-field
23702 vterron   20   0  R 98.506 0.219   0:39.79 solve-field
23711 vterron   20   0  R 93.751 0.218   0:36.62 solve-field
24007 vterron   20   0  R 86.278 0.084   0:01.28 solve-field
23967 vterron   20   0  R 80.843 0.223   0:07.28 solve-field
23697 vterron   20   0  R 77.446 0.198   0:37.32 solve-field
23678 vterron   20   0  S  0.679 0.036   0:00.32 python
23683 vterron   20   0  S  0.679 0.065   0:00.53 python
23691 vterron   20   0  S  0.679 0.065   0:00.47 python
23693 vterron   20   0  S  0.679 0.065   0:00.72 python
14615 vterron   20   0  S  0.000 0.003   0:00.18 sshd
15684 vterron   20   0  S  0.000 0.020   0:00.29 screen
16559 vterron   20   0  S  0.000 0.005   0:00.08 bash
17865 vterron   20   0  S  0.000 0.005   0:00.05 bash
18305 vterron   20   0  S  0.000 0.011   0:00.15 screen
19222 vterron   20   0  S  0.000 0.003   0:00.03 sshd
[ panic1 ][              0*$bash                              ][2015-04-19 18:53 ]
```

Multiple parallel processes of **solve-field**

--cores

(default → as many as there are available)

```
$ lemon astrometry data/*.fits WASP10/ --cores 4
```

Spawn four parallel processes

--radius

(default = 1)

```
$ lemon astrometry data/*.fits WASP10/ --radius 5
```

Do not search for matches in the entire sky, but only within five **degress** around the coordinates of each image

Index files

Built from an astrometric reference catalog

2MASS → ~32 GB

mosaic

Improve centroid detection

Reproject images onto a common coordinate system and **combine** them into a **mosaic**

This image is the one that we'll use to

detect astronomical objects

Maximize **signal-to-noise** ratio

This allows for a much more accurate determination of the **centroid of each star**, galaxy or any other celestial object.

Input FITS files (with **WCS**)

```
$ lemon mosaic HAT-P-16/*.fits HAT-P-16-mosaic.fits
```

Output FITS file

Reproject and combine the data

```
$ lemon mosaic WASP10/*.fits WASP10-mosaic.fits
>> Making sure the 193 input paths are FITS images...
>> 100%[===================================================>]
INFO: Listing raw frames [montage_wrapper.wrappers]
INFO: Computing optimal header [montage_wrapper.wrappers]
INFO: Projecting raw frames [montage_wrapper.wrappers]
INFO: Mosaicking frames [montage_wrapper.wrappers]
INFO: Deleting work directory [montage_wrapper.wrappers]
>> Reproject mosaic to point North... done.
>> You're done ^_^
```

Make North be up

Just a high-level interface to **Montage**

Basically a couple of calls to montage-wrapper

--filter

By default, all input FITS files are combined

```
$ lemon mosaic HAT-P-16/*.fits HAT-P-16-mosaic.fits --filter I
```

Use only the images **taken in I**

--background-match

Remove any discrepancies in brightness or background

```
$ lemon mosaic HAT-P-16/*.fits HAT-P-16-mosaic.fits --background-match
```

Include a **background-matching** step

Über-cool, but makes it take **much longer**

photometry

Aperture photometry

Detect sources on this image

Output database

```
$ lemon photometry WASP10-mosaic.fits WASP10/*.fits WASP10-phot.LEMONd
```

Do photometry on all these images

SExtractor

Use **SExtractor** for sources detection

Photometry is done on all the **detected** objects,
in each FITS image (via their celestial coordinates)

--coordinates

Don't detect sources; use these objects instead

```
$ lemon photometry       \
  WASP10-mosaic.fits     \
  WASP10/*.fits          \
  WASP10-phot.LEMONdB    \
  --coordinates coords.txt
```

Do photometry on the objects listed here

coords.txt

List one object per line, alpha and delta

```
100.2892077  9.4940359
100.2994373  9.4420255
100.3050527  9.4362469
100.3036477  9.4375102
100.1769466  9.4277355
100.1774339  9.4239221
```

In decimal degrees

coords.txt

Proper motions in arcsec / year

```
269.456271 4.665281
269.452075 4.693391 [-0.79858] [10.32812] # Barnard's Star
269.466450 4.705625 [0.0036] [-0.0064]    # TYC 425-262-1
```

In each image, **correct the position** of the star to account for its movement over time

IRAF

Aperture photometry is done with **IRAF**

Tasks are called via **PyRAF**

Aperture photometry

The aperture and aperture are determined by the

FWHM

For example, "aperture is 3 times the FWHM"

Aperture photometry

The median

FWHM

of all the images in each filter

This gives more robust results, especially in short time series

--individual

```
$ lemon photometry        \
  WASP10-mosaic.fits      \
  WASP10/*.fits           \
  WASP10-phot.LEMONdB     \
  --individual
```

Use the FWHM **of each image**, not the median

Output databases

have the extension

.LEMONdB

A SQLite database storing **all** the information

```
$ lemon photometry WASP10-mosaic.fits WASP10/*.fits WASP10-phot.LEMONdB
>> Examining the headers of the 193 FITS files given as input...
>> 100%[===================================================================>]
>> 2 different photometric filters were detected:
>>  B: 16 files (8.29 %)
>>  V: 177 files (91.71 %)
>> Making sure there are no images with the same date and filter... done.
>> Sources image: WASP10-mosaic.fits
>> Running SExtractor on the sources image... done.
>> Calculating coordinates of field center... done.
>> α = 349.0447049  (23 16 10.73)
>> δ =  31.4843645 (+31 29 03.71)
>> Detected 155 sources on which to do photometry.
>>
>> Need to determine the instrumental magnitude of each source.
>> Doing photometry on the sources image, using the parameters:
>> FWHM (sources image) = 8.535 pixels, therefore:
>> Aperture radius = 8.535 x 3.00 = 25.605 pixels
>> Sky annulus, inner radius = 8.535 x 4.50 = 38.407 pixels
>> Sky annulus, width = 8.535 x 1.00 = 8.535 pixels
>>
>> Running IRAF's qphot... done.
>> Detecting INDEF objects... done.
>> 9 objects are INDEF in the sources image.
>> There are 146 objects left on which to do photometry.
>> Making sure INDEF objects were removed... done.
```

Detect sources

Determine instrumental magnitude of each object

Do photometry in the B filter...

```
>> Initializing output LEMONdB... done.
>>
>> Let's do photometry on the 16 images taken in the B filter.
>> Calculating the median FWHM for this filter... done.
>> FWHM (B) = 9.815 pixels, therefore:
>> Aperture radius = 9.815 x 3.00 = 29.445 pixels
>> Sky annulus, inner radius = 9.815 x 4.50 = 44.168 pixels
>> Sky annulus, width = 9.815 x 1.00 = 9.815 pixels
>> 100%[=======================================================>]
>>
>> Let's do photometry on the 177 images taken in the V filter.
>> Calculating the median FWHM for this filter... done.
>> FWHM (V) = 9.864 pixels, therefore:
>> Aperture radius = 9.864 x 3.00 = 29.592 pixels
>> Sky annulus, inner radius = 9.864 x 4.50 = 44.388 pixels
>> Sky annulus, width = 9.864 x 1.00 = 9.864 pixels
>> 100%[=======================================================>]
>> Storing photometric measurements in the database...
>> 100%[=======================================================>]
>> Gathering statistics about tables and indexes... done.
>> You're done ^_^
```

... and now in V

User scripts can be written using the LEMON **library**

```
from lemon.database import LEMONdB
from lemon.passband import Passband

path = "WASP10-phot.LEMONdB"
db = LEMONdB(path)

print(len(db))      # number of stars
print(db.pfilters) # photometric filters

# Fetch some basic information of a star
star_id = 100
ra, dec, imag = db.get_star(star_id)[-3:]
print(ra)   # right ascension
print(dec)  # declination
print(imag) # instrumental magnitude

# Loop over the photometric measurements
pfilter = Passband("Johnson I")
phot = db.get_photometry(star_id, pfilter)
for index in range(len(phot)):
    time = phot.time(index)
    mag  = phot.mag (index)
    snr  = phot.mag (index)
```

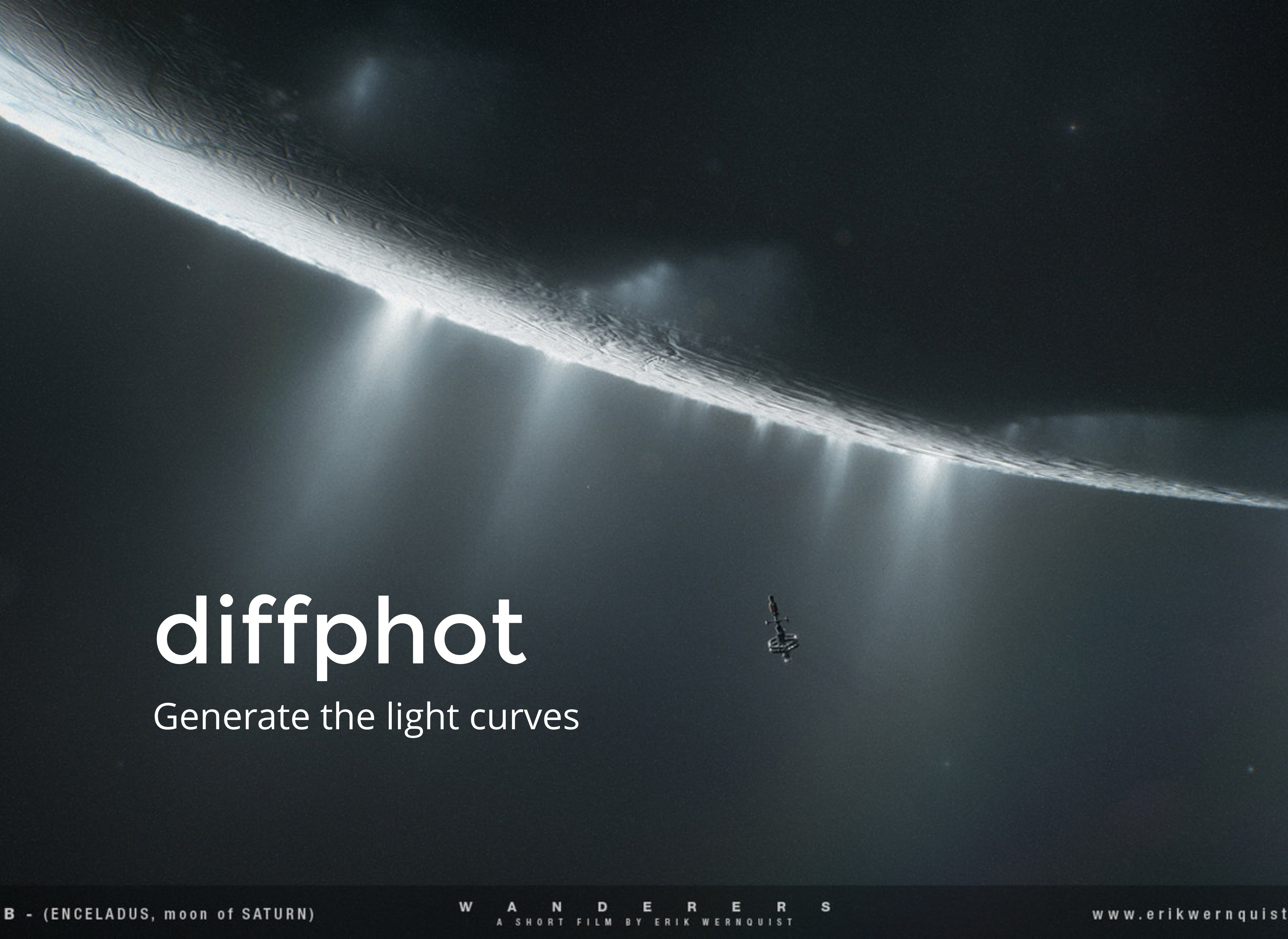
The future

photutils

For both sources detection and photometry

diffphot

Generate the light curves



Input database

```
$ lemon diffphot phot.LEMONdB curves.LEMONdB
```

Output database

Compute light curves in the B filter...

```
$ lemon diffphot WASP10-phot.LEMONdB WASP10-diff.LEMONdB
>> Making a copy of the input database... done.
>> There are 146 stars in the database
>>
>> Light curves for the B filter will now be generated.
>> Loading photometric information... done.
>> 100%[=========================================>]
>> Storing the light curves in the database...
>> 100%[=========================================>]
>>
>> Light curves for the V filter will now be generated.
>> Loading photometric information... done.
>> 100%[=========================================>]
>> Storing the light curves in the database...
>> 100%[=========================================>]
>> Updating statistics about tables and indexes... done.
>> You're done ^_^
```

... and now in V

How does this work?

I'm so glad you asked

The algorithm

A new algorithm for differential photometry: computing an optimum artificial comparison star

(C. Broeg, 2005)

2005AN....326..134B

Nobody actually ever reads the papers, but here's the link anyway

The algorithm

For each star, for each photometric filter:

1

Identify the **most constant** stars in the field

That is, those with the most stable light curve

The algorithm

For each star, for each photometric filter:

2

Combine them into an **artificial** comparison star

With weights inversely proportional to their statistical dispersion

The algorithm

For each star, for each photometric filter:

3

Compare the instrumental magnitude to the artificial one

That's the Δ magnitude!

But, wait a second...

To identify the comparison stars we need... their light curves

That's a **recursive** problem!

How Broeg **solves** this

1

Asume that **all the stars** in the field are constant

With weights inversely proportional to their brightness

How Broeg **solves** this

For each star

2

Use all the stars in the field (except **itself**) as comparison star

How Broeg **solves** this

For each star

3

Generate the light curve and compute its statistical **dispersion**

How Broeg **solves this**

After doing this for all the stars...

Discard those with the **highest** disperson

How Broeg **solves** this

Rinse and **repeat**...

How Broeg **solves** this

… until only *N* stars remain

These are our comparison stars

--stars

(default = 20)

```
$ lemon diffphot phot.LEMONdB curves.LEMONdB --stars 3
```

Use the best three stars as comparison

--worst-fraction

(default = 0.1)

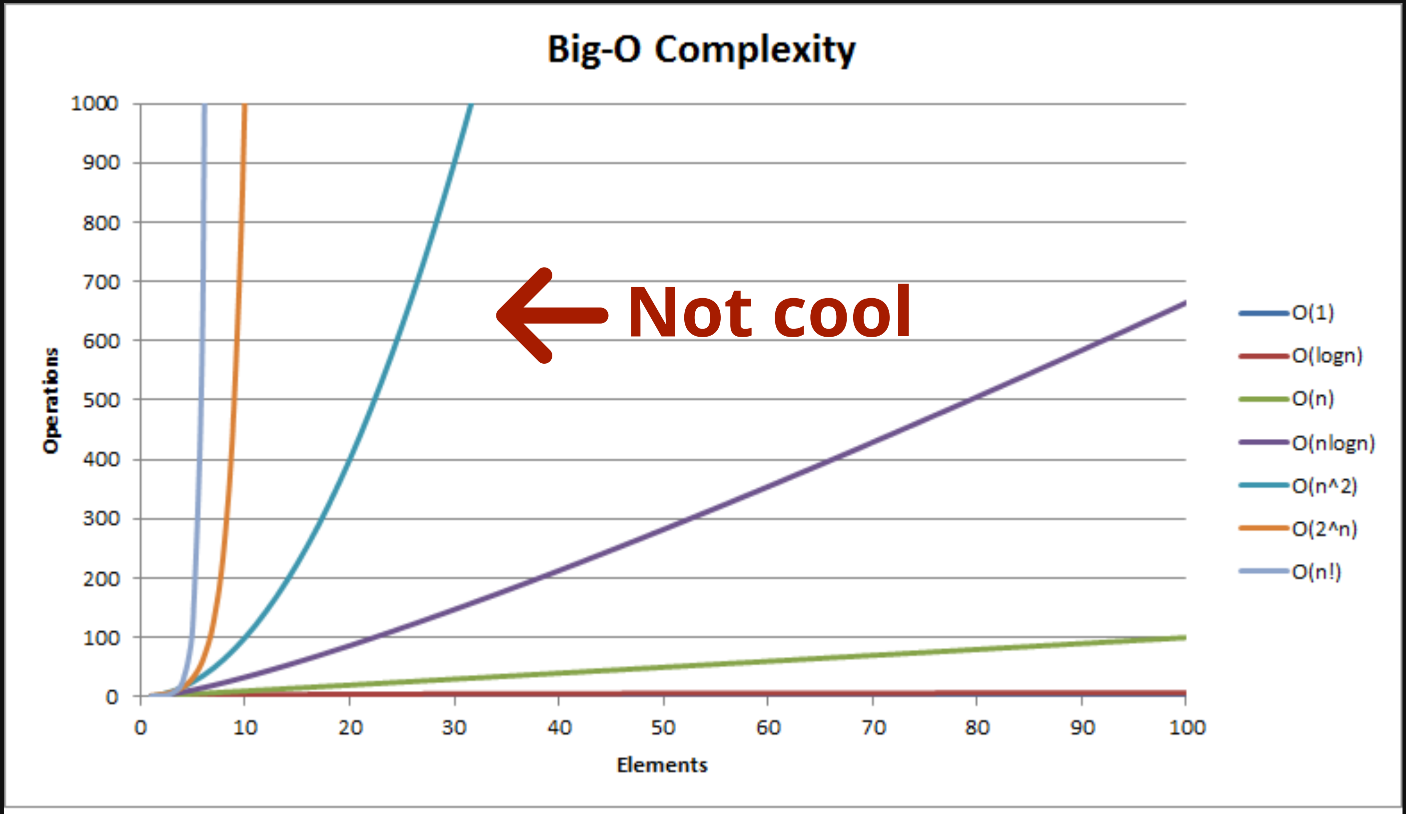
```
$ lemon diffphot phot.LEMONdB curves.LEMONdB --worst-fraction 0.05
```

At each iteration, discard the worst 5%

Oh, and since we speak programming...

The algorithm is $O(N^2)$

That's bad



Big-O Algorithm Complexity Cheat Sheet

It's

$$O(N^2)$$

because, in our implementation,

The algorithm must be run for **each star**

In order to find its optimal artificial comparison star

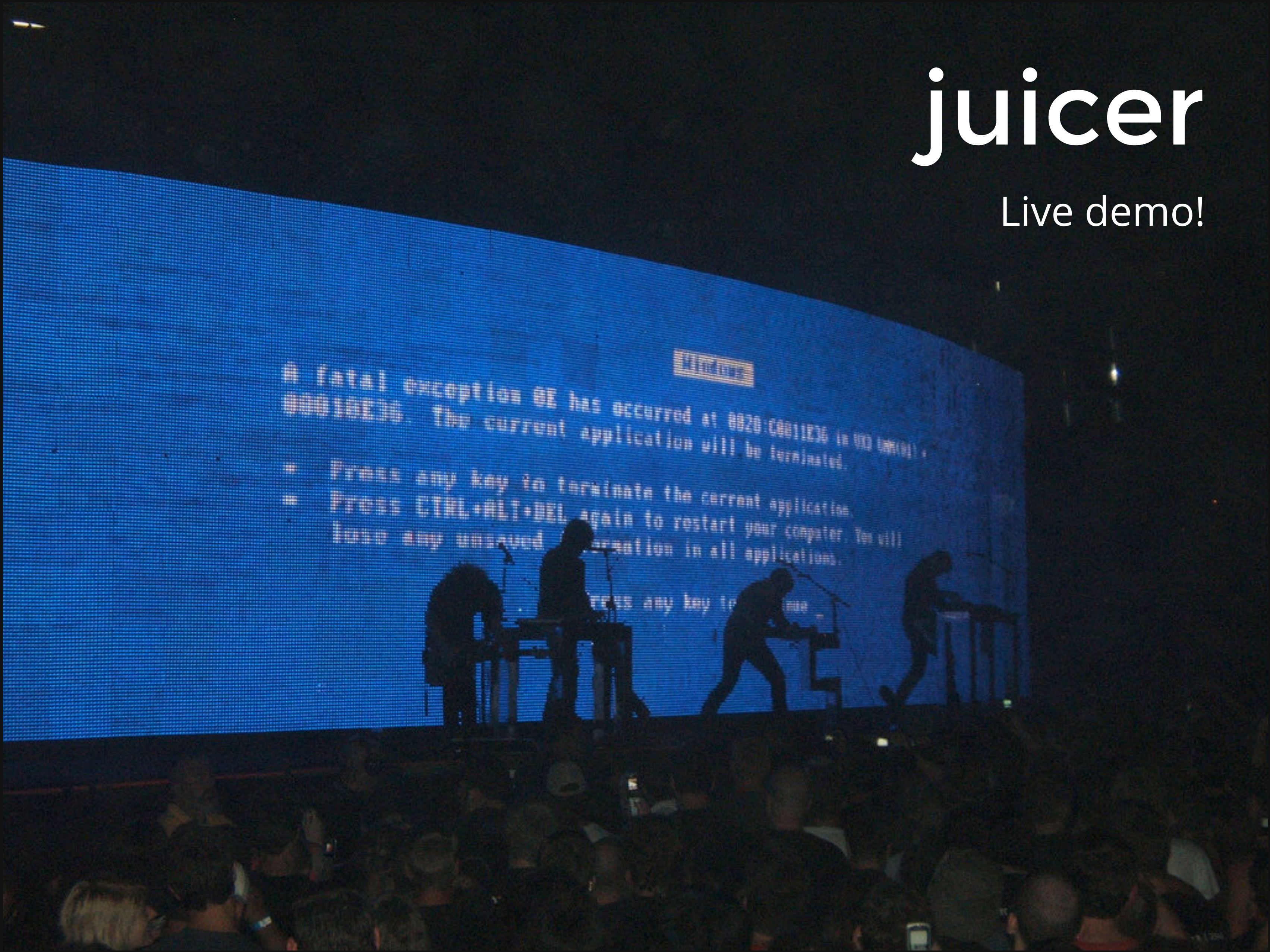
$$O(N^2)$$

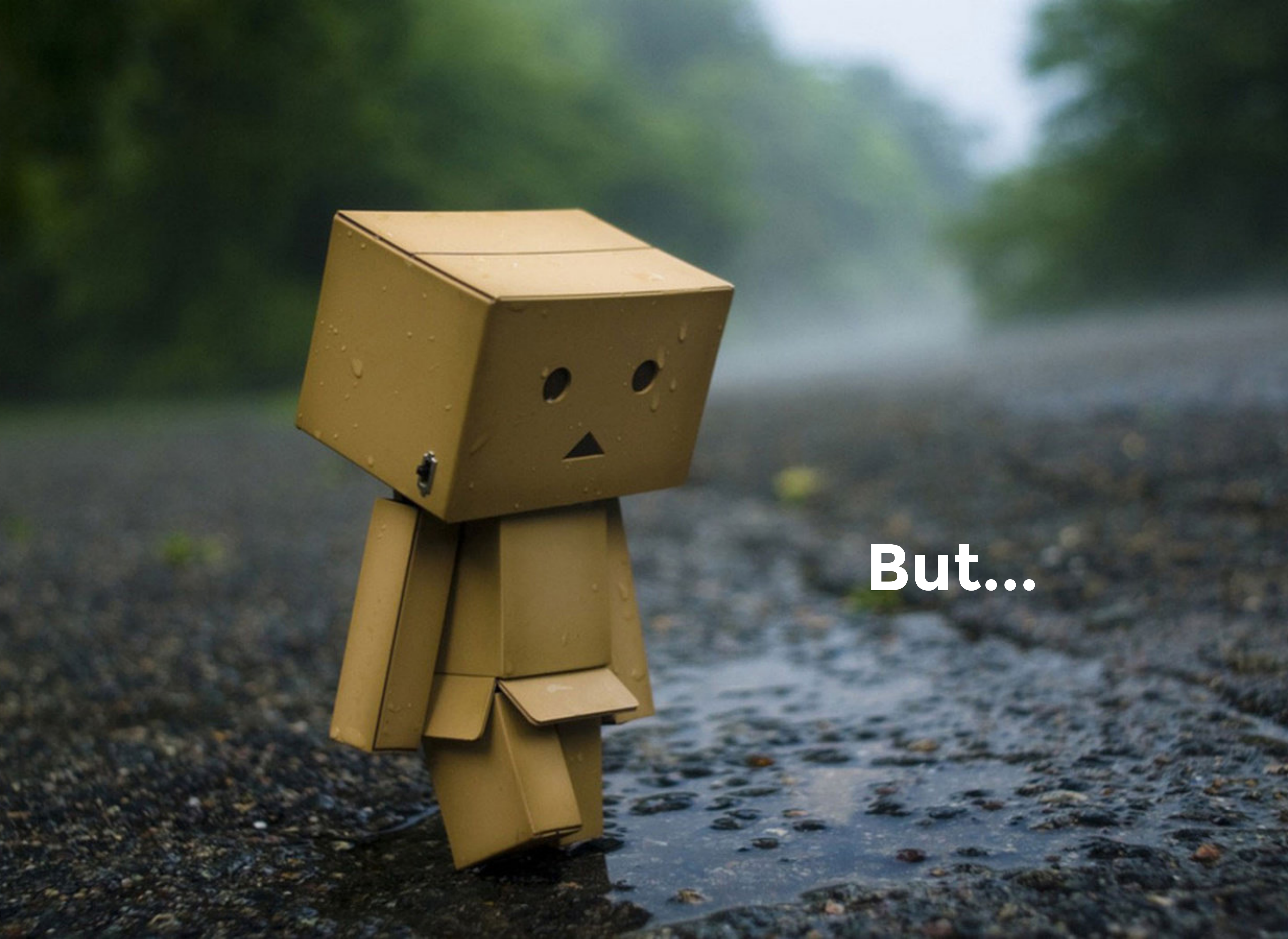
But the bottleneck is usually **telescope** time, not **CPU** time

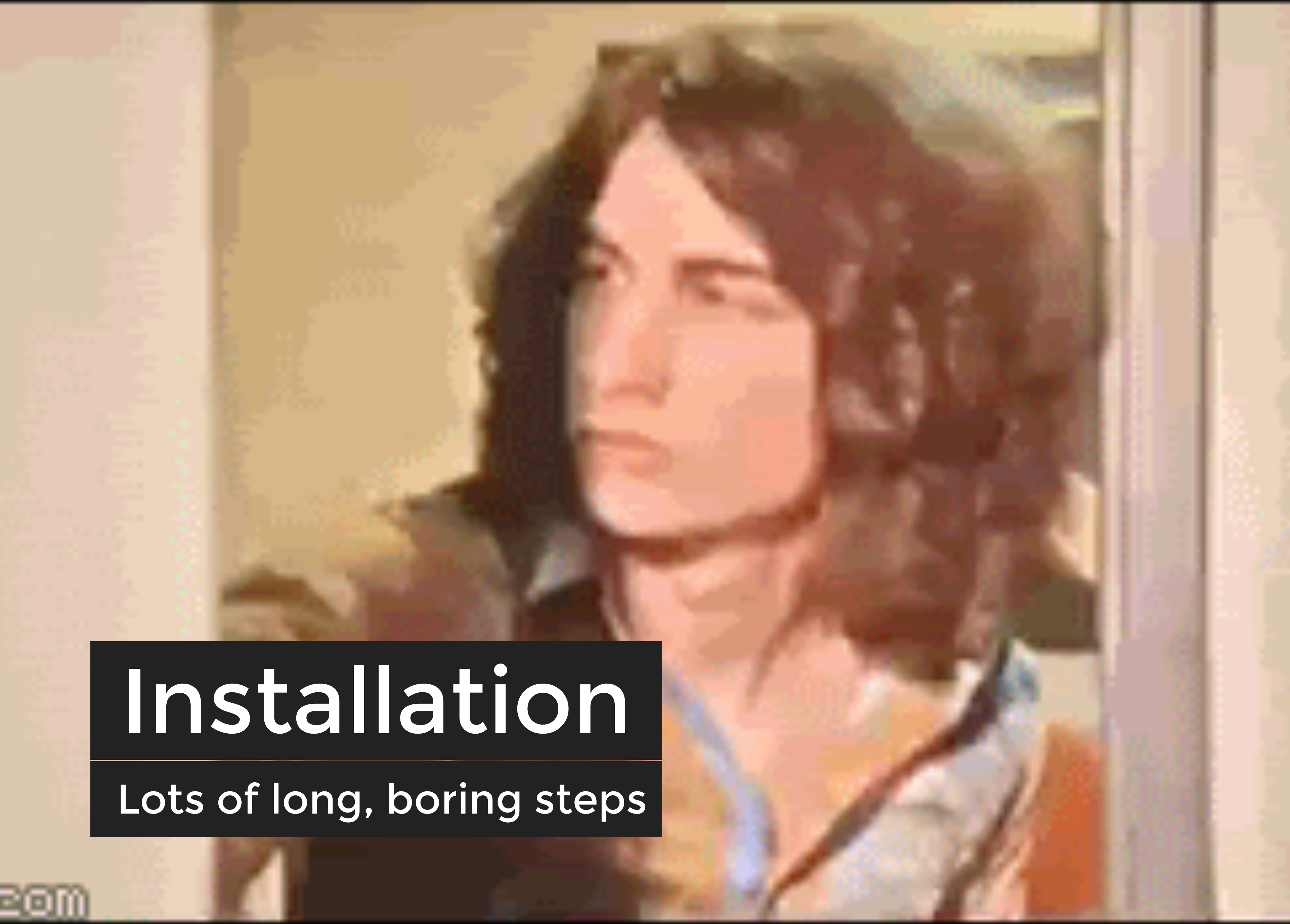
If that's not your case, buy more CPUs!

juicer

Live demo!




But...



Installation

Lots of long, boring steps

Things shouldn’t be this way.
Not in Python.

I ripped this off from somebody else.

```
pip install lemon
```

It will be a package on **PyPI** – soon

Second appareance of the logo
Shameless attempt at leaving an imprint on your minds

http://github.com/vterron
http://lemon.readthedocs.org/

Image credits

- Artist's concept of a rocky Earth-sized exoplanet in the habitable zone of its host star, possibly compatible with Kepler-186f's known data (NASA/SETI/JPL) [Wikipedia]
- Hubble Extreme Deep Field (full resolution) released by NASA on September 25th, 2012 [Wikipedia]
- Futurama Poster — A Mindless Worker Is A Happy Worker.
- Screenshot from Back to the Future (1985)
- Screenshots from Wanderers - a short film by Erik Wernquist
- Icons "Business idea", "Settings gears", "Winner jumping on podium with medal of number one", "Number 1 drawing", "Number 2", "Drawing of number 3", "School Doubt", "Multiply sign" and "Leage Winner", by Freepik. License: CC BY 3.0
- Icon "Loop", by Icomoon. License: CC BY 3.0
- Big-O Complexity Chart, by Eric Rowell.
- 30 Priceless Blue Screen of Death (BSoD) to Chuckle About, by Hongkiat.
- Sadness in Rain wallpaper.
- Fluffy Haired Overreaction GIF.