ZTF Pipelines Status and Deliverables

Frank Masci & the IPAC-Caltech ZTF Team

ZTF Working Group Meeting, November 2016

The ZTF Science Data System (ZSDS)

- The ZSDS is housed at the Infrared Processing and Analysis Center (IPAC), Caltech
- IPAC is a multi-mission science center (IRAS, ISO, *Spitzer*, WISE, Herschel, Planck, 2MASS …)
- Responsibility for ZTF (like PTF):
  - data transfer from P48 to IPAC;
  - data processing pipelines;
  - long-term data archiving, curation, user-interfaces, and APIs to retrieve data;
  - generation of transient alerts and metadata to support near real-time discovery;
  - maintenance of operations, databases, fileservers, and archive infrastructure.

The ZTF Science Data System (ZSDS)

- A complex, “big-data” / system-engineering problem
- Developed from scratch to handle expected data-rates and volumes
- PTF processing architecture could not be scaled in a *faithful* manner for ZTF
- Learned many lessons from PTF: how not to do things
- Some heritage pipeline software was borrowed from other projects; also publicly available
- **Design goals:**

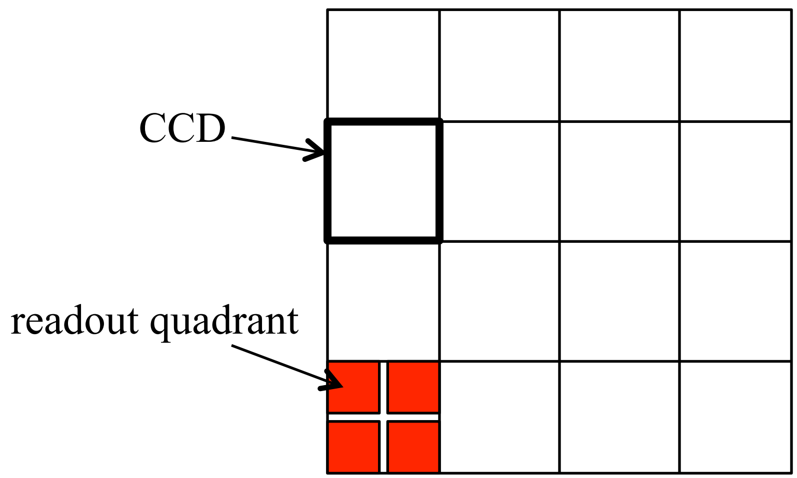
  reliability, modularity, simplicity, efficient product delivery, data tractability and recovery, resiliency to hardware outages

The ZSDS Staff

- **Ben Rusholme:** data transfer from P48 to IPAC; pipeline job scheduling/executive; optimization; source-matching infrastructure; software/config. management; pipeline modules
- **David Shupe:** astrometric calibration; source-matching and relative photometry; other analysis
- **Frank Masci (ZSDS Lead):** instrumental and photometric calibration; reference-image generation; image-subtraction and extraction; realtime pipeline; moving-objects; algorithms
- **Russ Laher:** pipeline infrastructure; integration; end-to-end testing; ingest; pipeline executive; database schemas and stored procedures; flat and bias-generation pipelines;
- **Steven Groom (and staff; IRSA lead):** pipeline operations and archive design; system engineering
- **Lee Bennett (and staff; ISG Lead):** system-engineering and hardware
- **Ed Jackson:** database management
- **Jason Surace:** image-simulation; data analysis
- **Ron Beck:** pipeline operations (iPTF for now)
- **David Imel (IPAC manager):** budgeting and personnel
- **George Helou (IPAC director)**

ZTF Raw Camera Image Data

- One camera exposure: 16 CCDs; each ~ 6k x 6k pixels
- Image data packet transmitted is one CCD (four readout-quadrant images)
- 16 CCD-based image files are transmitted every 45 sec.
- Full camera exposure: ~ 1.3GB uncompressed
- Require *lossy* compression to accommodate transfer bandwidth (~ 110 - 150 Mbits/sec, variable)

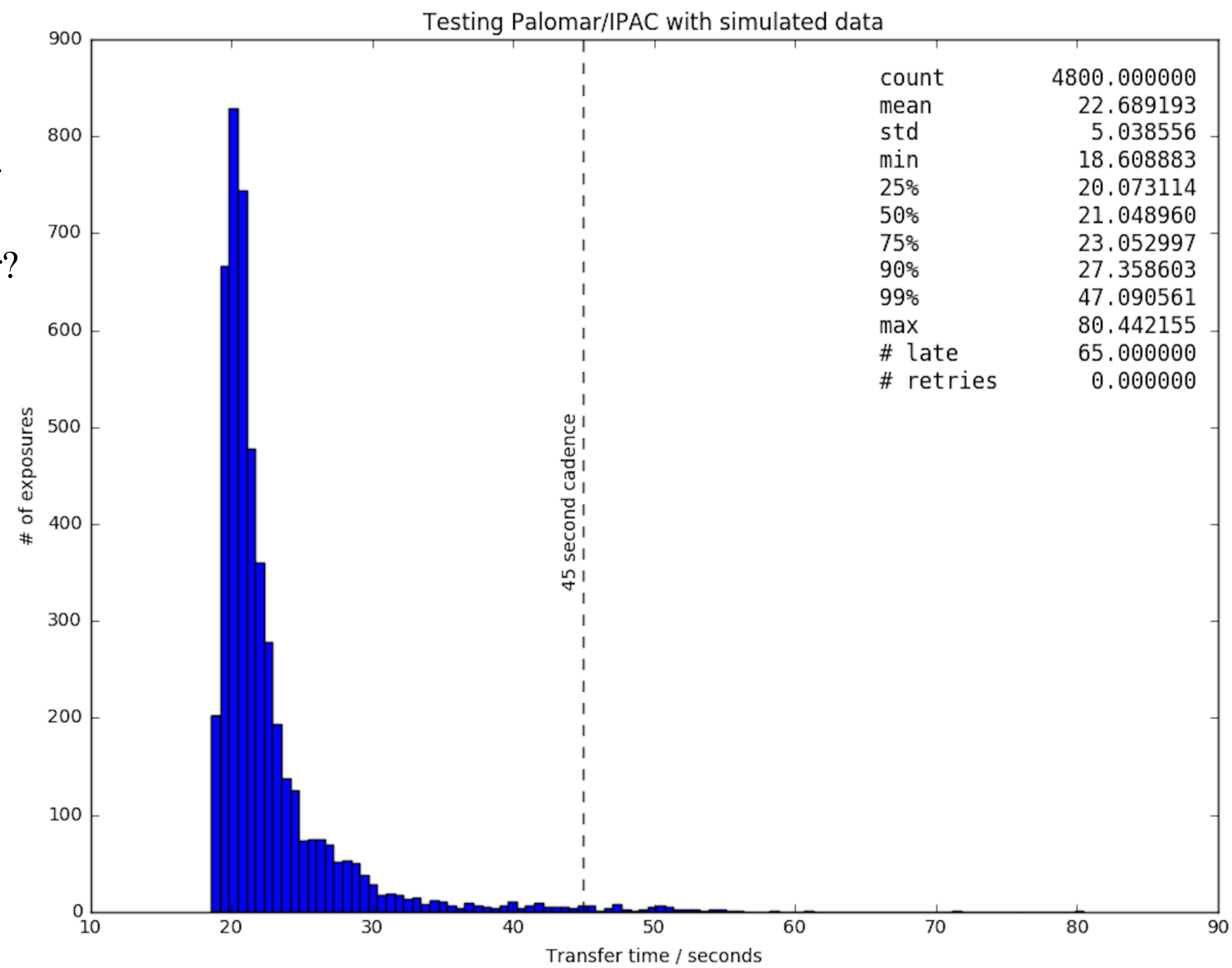


Basic image-unit for pipeline processing from which all products are derived is a ~3k x 3k readout quadrant image.

Update on P48 – IPAC data transfer

Performed by Ben Rusholme by transferring simulated camera image data (ongoing study).

- Peak corresponds to ~150 Mbps
- Depends on water vapor content below the P48
- Data compression factor?



ZTF Pipelines

Overall, there are 10 inter-dependent pipelines (one is TBD):

Raw data ingestion/processing:

1. Raw data ingest, archival of raw images and storage of metadata in database [*realtime*]
2. Raw-image uncompression, splitting into readout-quadrant images, floating bias correction, simple QA [*realtime*]

Calibration generation:

3. Bias-image derivation from stacking calibration images acquired in afternoon [*made before on-sky operations*]
4. High-ν flat (pixel-to-pixel responsivity) from stacking calibration images [*made before on-sky operations*]
5. **TBD:** Low-ν flat from either long-term ZPVM or dithered-star observations [*every week, month or longer?*]

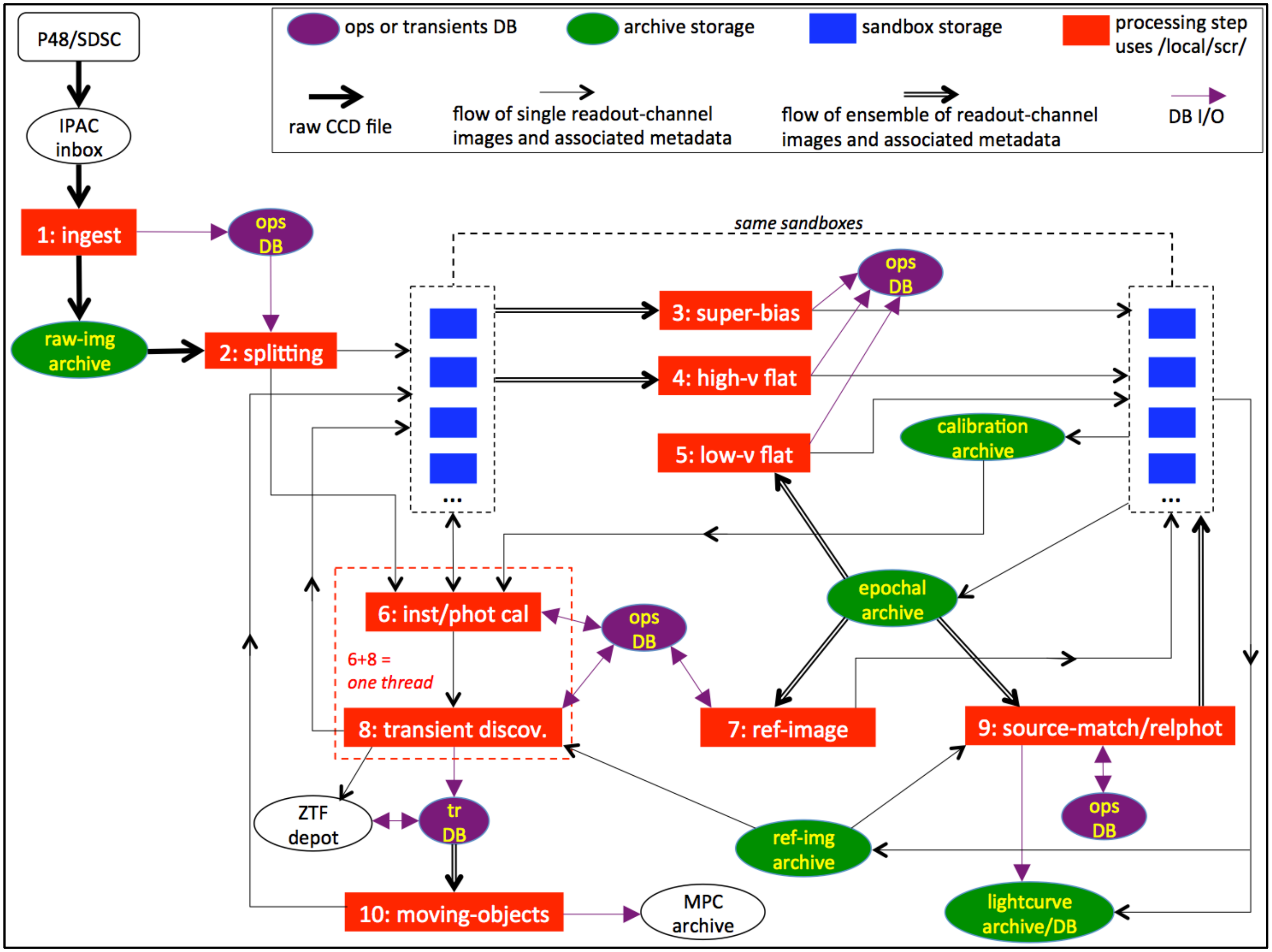
Real-time:

6. Instrumental calibration of readout-quadrant images: astrometry and photmetric cal [*realtime*]
7. Image subtraction and transient discovery (point sources / streaks), metadata and cutouts [*realtime*]

Ensemble-based processing:

8. Reference-image generation (co-addition of epochal images from 6) [*as needed: when good quality data available*]
9. Source-matching with relative photometric refinement for lightcurves; inputs from 6 [*every two weeks or longer?*]
10. Moving object pipeline system (MOPS): tracklets from linking transients from 7 [*every 3 or 4 hours during night*]

Data & processing flow



Deliverables and Products

1. **Instrumentally calibrated, readout-quadrant based epochal image products:**
   - images with photometric zero-points derived from PTF-fit photometry
   - bit-mask images
   - two source catalogs per image: PSF-fitting and aperture photometry: only PSF-fit catalog is absolutely calibrated
   - difference images with QA metadata

2. **Reference images (co-adds), coverage, unc maps, and two source catalogs per image**: PSF-fitting and aperture

3. **Match-files from source-matching and relative photometry of epochal extractions:** based on epochal PSF-fit photometry catalogs; to support "object-based" lightcurve database

4. **Products to support near real-time discovery:** *thresholded* transient candidates (point sources and streaks) with metadata and image cutouts

5. **Historical (users) database of all transient candidates and metadata generated from real-time pipeline**

6. **To commence 12 months after survey start**: transient alert stream extracted from real-time pipeline

7. **Products to support Solar System/NEO discovery and characterization:**
   - moving object tracks from linking point-source transients; known objects are tagged.
   - delivered to the IAU's Minor Planet Center following human vetting.

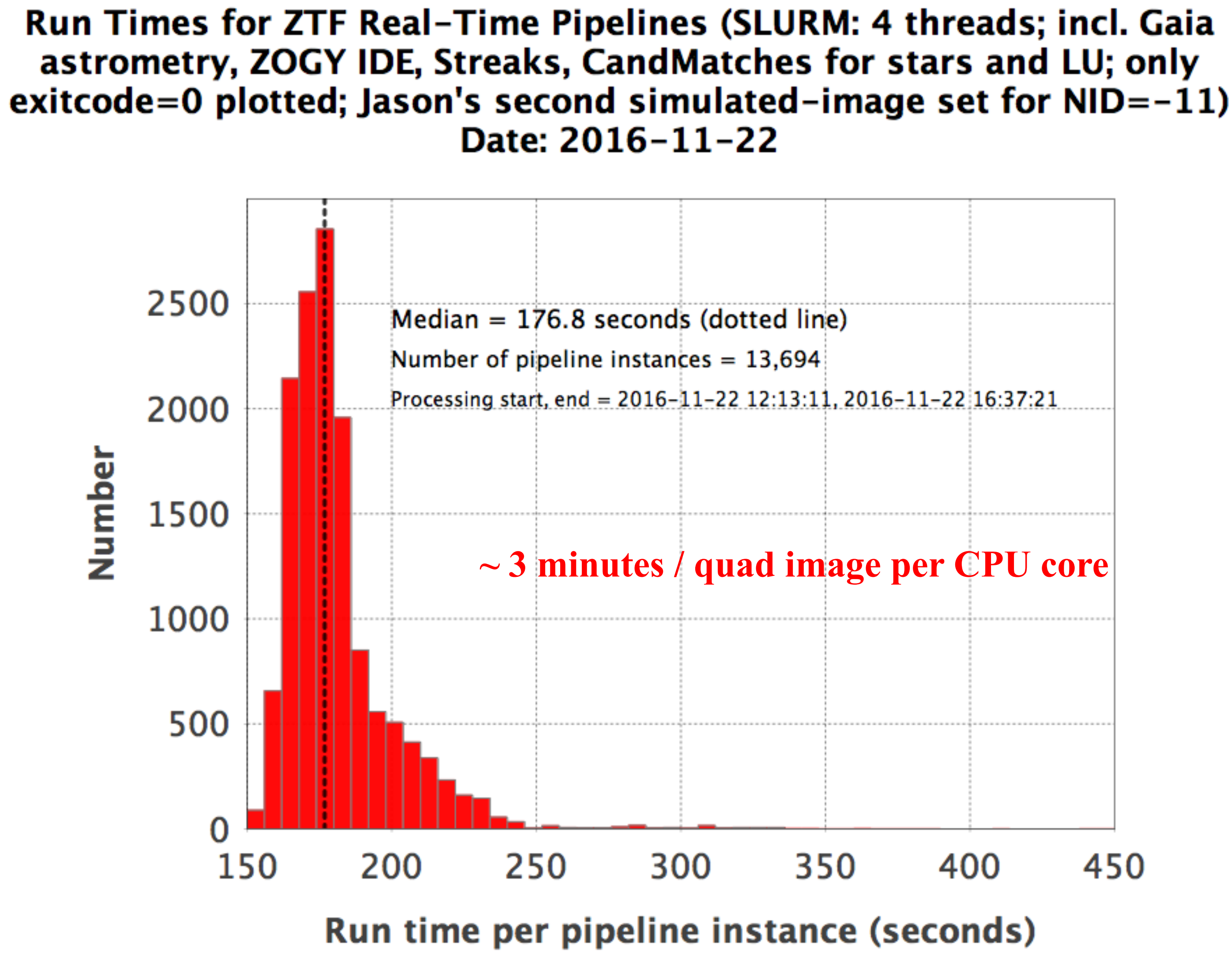
ZTF Real-time pipeline

- Does most of the heavy-lifting in real-time.
- Time-critical: to support near real-time discovery
- **Requirement:** 95% of the images received at IPAC must be processed with transient candidates presented to marshals in < 10 minutes (goal is 5 minutes)

- Real-time pipeline consists of two phases:
  1. Instrumental calibration (bias-corrections, flat-fielding, astrometry, photometric calibration, pixel masks …): generates single-epoch image and catalog products for archive.
  2. Uses outputs from 1 to perform image subtraction, extraction of transient candidates, metadata, cutouts …

- Currently tested using a camera-image simulator, written by Jason Surace:
  - ➢ Takes as input a "schedule" of camera pointings from Eric's survey simulator, with multiple epochs on same region of sky, in any filter.
  - ➢ For fields/CCDs that overlap with the SDSS footprint, sources are injected with same photometric properties and positions as in the SDSS catalog; appropriate noise is also injected.
  - ➢ Point-source and streaking transients are also simulated.
  - ➢ Data files are packaged and compressed according to camera-software specifications.

ZTF real-time pipeline runtime
(processing unit = one readout-quad image)



ZTF real-time processing throughput (naïve estimate)

- Incoming data rate (set by cadence):
  - one exposure or 64 quadrant images / 45 sec.
  - *inprate* ~ 85 quad images / minute, on average

- Processing rate (median as of today):
  - *outrate* ~ 1 quad image / 3 minutes / CPU core

- If processing was purely CPU-limited, no or negligible I/O latency, *minimum* number of CPU cores needed to keep up with input data rate is:

  $N_{cores} = inprate \,/\, outrate = 255$ cores

- This estimate is naïve since it ignores I/O, network speed, other interleaved processing tasks.
  Goal is to process faster than incoming data rate. Furthermore, runtime histogram has a tail.

- Our currently "active" ZTF compute cluster has 16 physical cores × 34 nodes = 544 cores
  (or 16 × 2 × 34 = 1088 admissible simultaneous threads, contingent on shared resources)

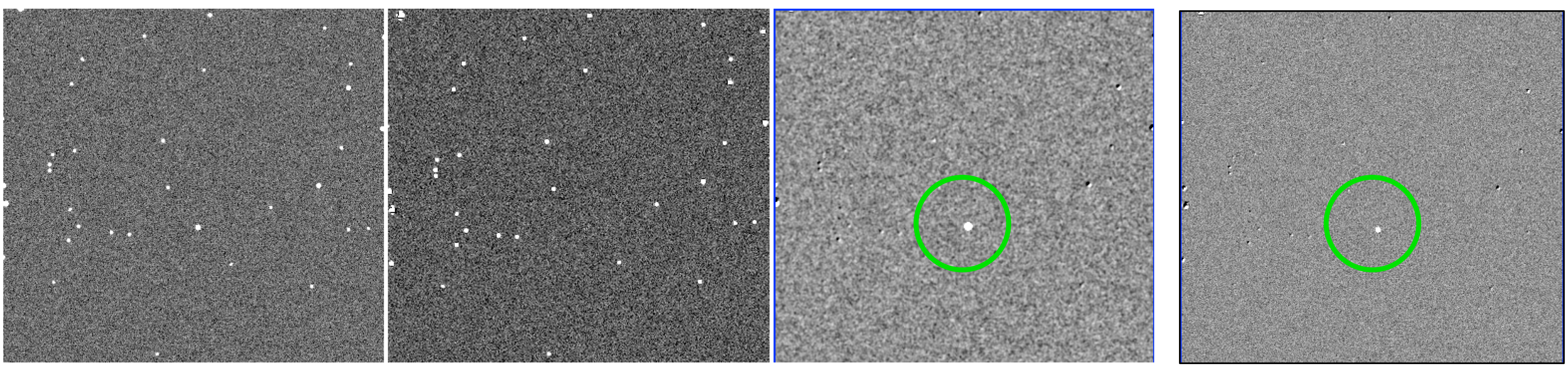
- Studies to maximize processing throughput, end-to-end, are in progress.

Implementation of ZOGY in image-subtraction pipeline

- ZOGY method: Zackay, Ofek, Gal-Yam (arXiv:1601.02655)
- First version implemented by Brad Cenko in Python. Uses pre-regularized image inputs.
- Parameter free! Optimality criterion: maximize S/N for point-source detection in sub-image.
  - Generates a "Scorr" (matched-filtered S/N) image for optimal point-source detection
  - de-correlates the pixel noise in subtraction image used for photometry;
  - also generates an estimate of the effective PSF for the sub-image.

Products from simulated images:

New image | **Reference image** | **S/N (Scorr) image** | **Sub-image**



used for detection | *used for photometry*

PTFIDE versus ZOGY on iPTF data

- Adapted ZTF image-subtraction pipeline (that executes Brad Cenko's Python implementation of ZOGY) to process PTF image data
- Experimented on 6 fields containing transients discovered from ToO program on event GW150914

THE ASTROPHYSICAL JOURNAL LETTERS, 824:L24 (9pp), 2016 June 20

iPTF SEARCH FOR AN OPTICAL COUNTERPART TO GRAVITATIONAL-WAVE TRANSIENT GW150914

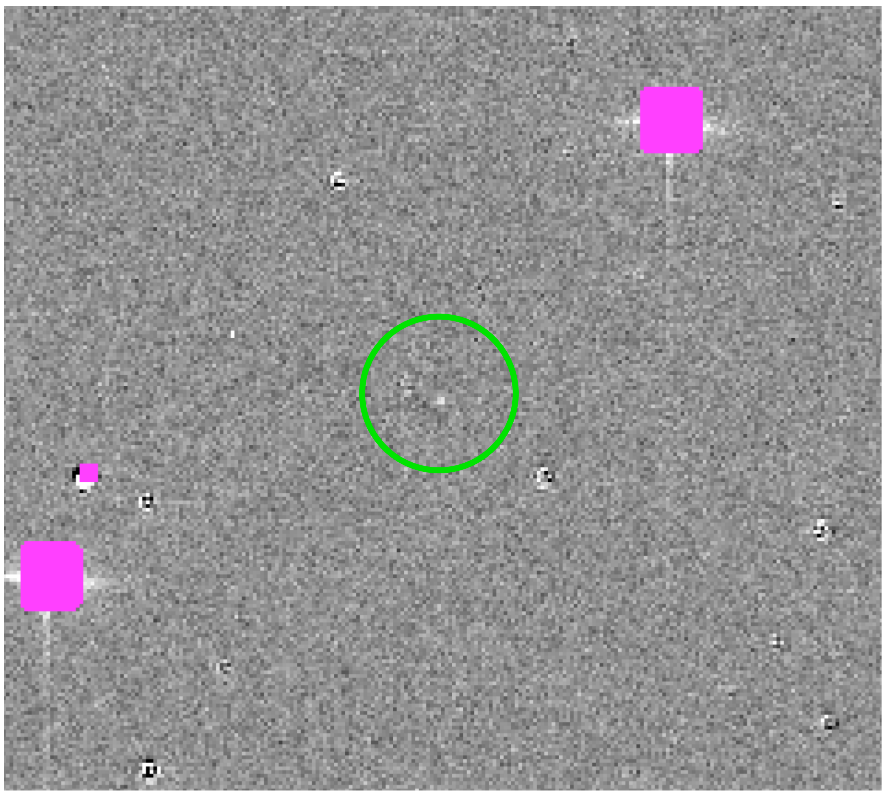
M. M. KASLIWAL[1], S. B. CENKO[2,3], L. P. SINGER[2,20], A. CORSI[4], Y. CAO[1], T. BARLOW[1], V. BHALERAO[5], E. BELLM[1], D. COOK[1], . . .

Name	RA (J2000)	DEC (J000)	Discovery Time
→ iPTF15cyo	$8^h\ 19^m\ 56^s.18$	+13 d 52′ 42″.0	2015 Sep 17 05:54:55.6
iPTF15cyp	$8^h\ 21^m\ 43^s.68$	+16 d 12′ 42″.0	2015 Sep 17 05:56:31.6
→ iPTF15cys	$8^h\ 11^m\ 55^s.59$	+16 d 43′ 10″.1	2015 Sep 17 06:05:16.6
→ iPTF15cym	$7^h\ 52^m\ 35^s.67$	+16 d 45′ 59″.6	2015 Sep 17 05:46:17.1
→ iPTF15cyq	$8^h\ 10^m\ 00^s.86$	+18 d 42′ 18″.1	2015 Sep 17 05:57:16.3
→ iPTF15cyn	$7^h\ 59^m\ 14^s.93$	+18 d 12′ 54″.9	2015 Sep 17 05:47:20.5
iPTF15cyt	$7^h\ 38^m\ 59^s.35$	+21 d 45′ 43″.2	2015 Sep 17 06:08:09.3
→ iPTF15cyk	$7^h\ 42^m\ 14^s.87$	+20 d 36′ 43″.4	2015 Sep 17 05:38:38.3

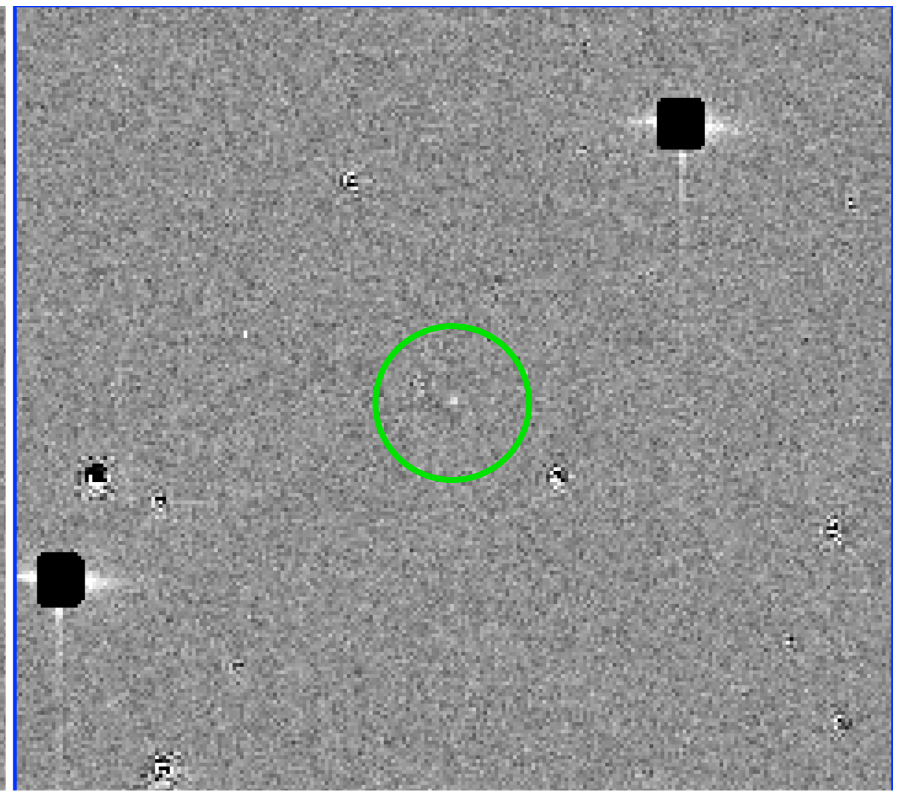
PTFIDE versus ZOGY on iPTF data

iPTF15cyk

ZOGY



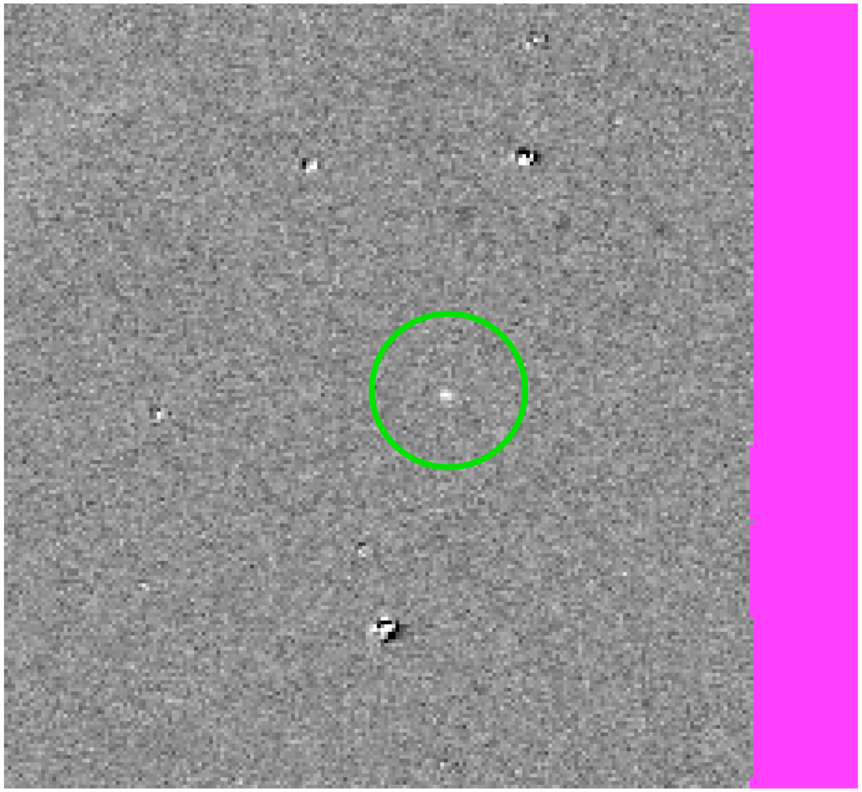
PTFIDE



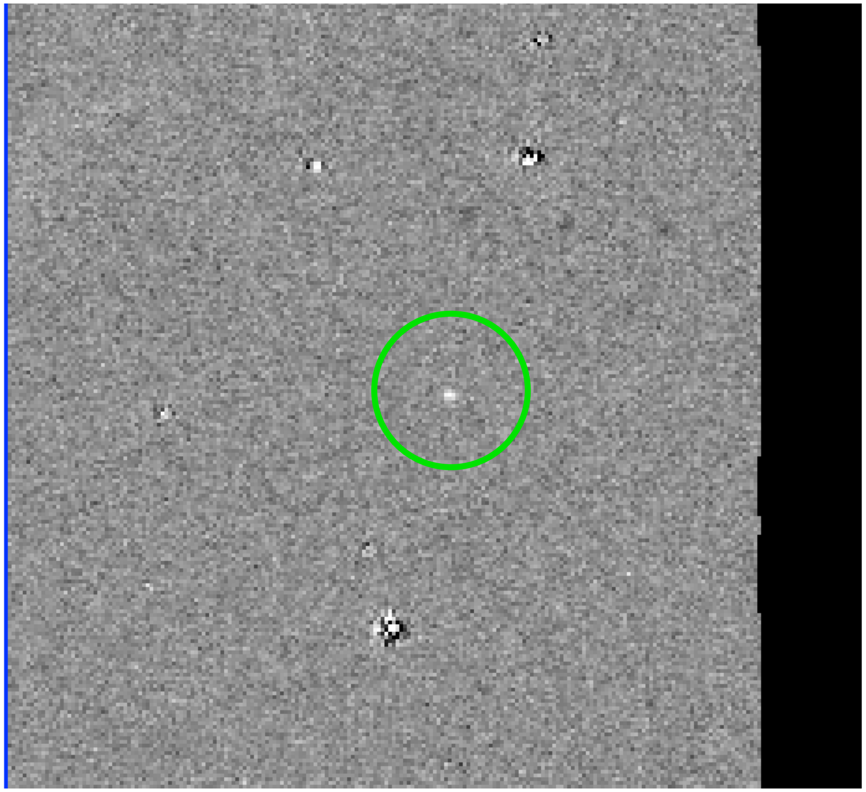
PTFIDE versus ZOGY on iPTF data

iPTF15cym

ZOGY

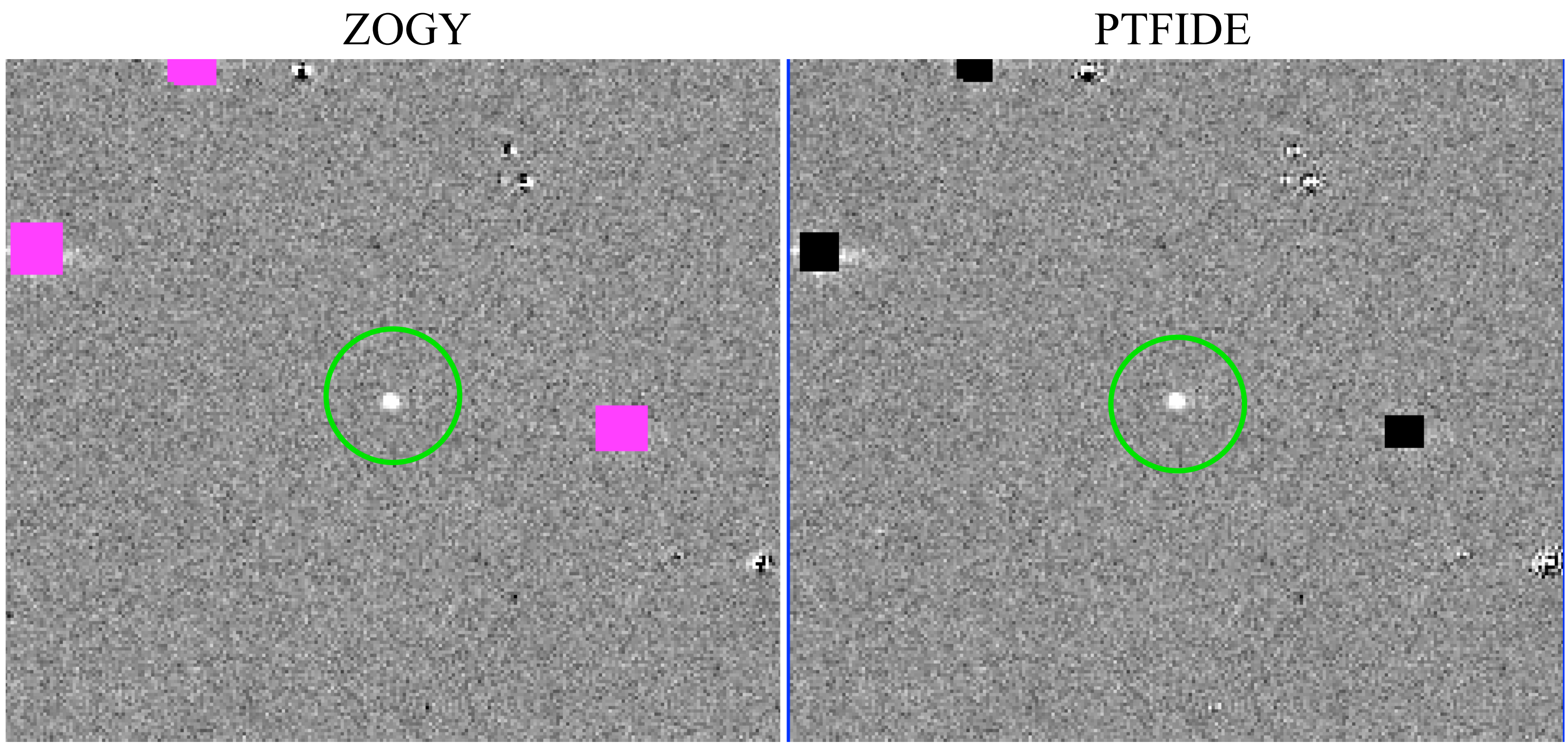


PTFIDE



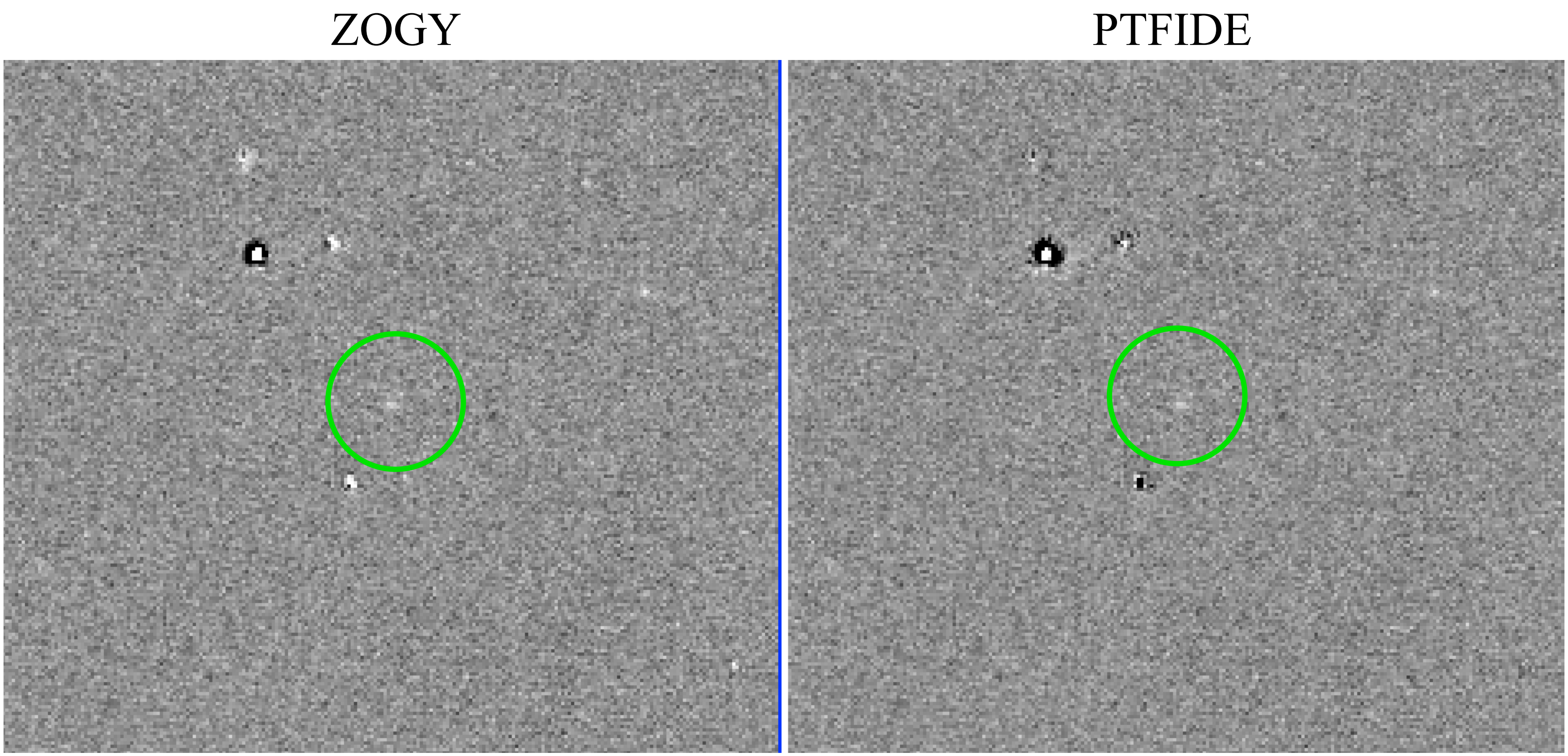
PTFIDE versus ZOGY on iPTF data

iPTF15cyo



PTFIDE versus ZOGY on iPTF data

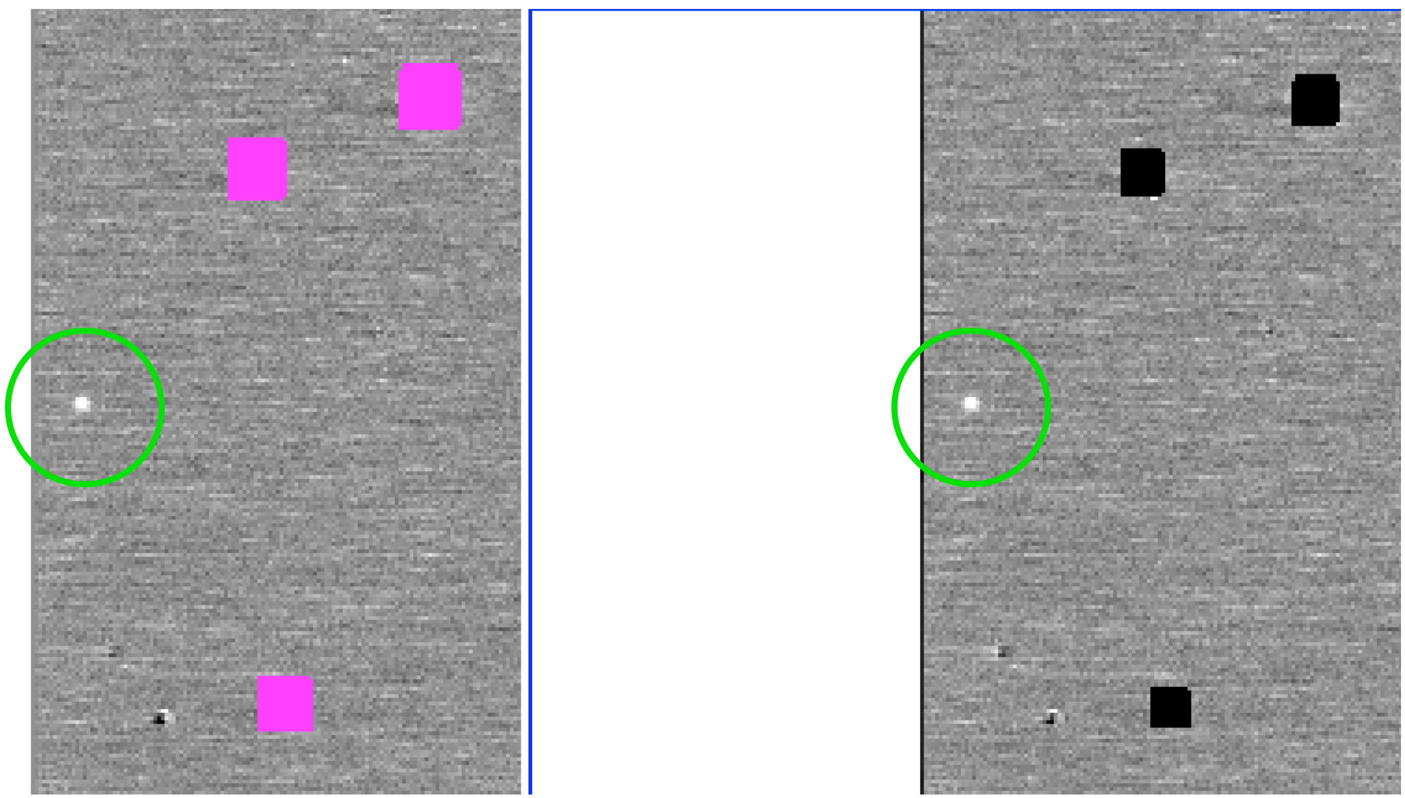
iPTF15cyq



PTFIDE versus ZOGY on iPTF data

iPTF15cys

ZOGY PTFIDE



PTFIDE vs ZOGY: summary statistics

- Number of **raw candidates** extracted to S/N = 5, using simple PSF-shape/morphology filters to remove obvious false-positives, i.e., no machine-learned RealBogus (RB) vetting.

real transient	Field/CCD	#candidates (PTFIDE)	#candidates (ZOGY)	#asteroids
iPTFcyk	3658 / 8	181	121	2
iPTFcym	3459 / 6	472	66	1
iPTFcyn	3560 / 7	343	436	10
iPTFcyo	3359 / 8	268	67	3
iPTFcyq	3561 / 6	210	196	4
iPTFcys	3460 / 9	350	417	4

- **NOTES:**
  - same archival PTF reference image co-adds were used in PTFIDE and ZOGY subtractions, created using an old/non-optimal method --- will be different for ZTF
  - PTF epochal images used old astrometric calibration method --- will also be different for ZTF

PTFIDE versus ZOGY

- **Conclusion:** PTFIDE and ZOGY appear to show similar performance on PTF data, at the raw level, noting the non-optimal calibrations upstream;

- ZOGY is slightly better perhaps?

- Regardless, this exercise shows that difference-image quality is primarily driven by quality of upstream calibrations (systematics): astrometry, flat-fielding, gain-matching, PSF-estimation.

- Upstream calibrations must be accurate before one starts to benefit from the *statistical*-optimality property underlying ZOGY, i.e., maximum point-source S/N in limit of background dominated noise

Number of transient candidates

- **PTF experience:**

  Raw transient stream, ~ 200 – 300 candidates per image (chip).

  ⬇

  Machine-learned RB vetting, ~ ten(s) *likely real* candidates per image; all flavors of transients;

  with ~ 250 PTF exposures/night × 11 chips × 20 candidates/chip, ~ 55,000 candidates/night.

  ⬇

  Marshal automated-vetting for specific science cases, e.g., ≥ 2 detections in night, etc.

- **Expectation for ZTF:**

  Raw transient stream, <~ 150/image ?

  ⬇

  Simple filtering on candidate metrics (or RB?), ~ ten(s) *likely real* candidates per image;

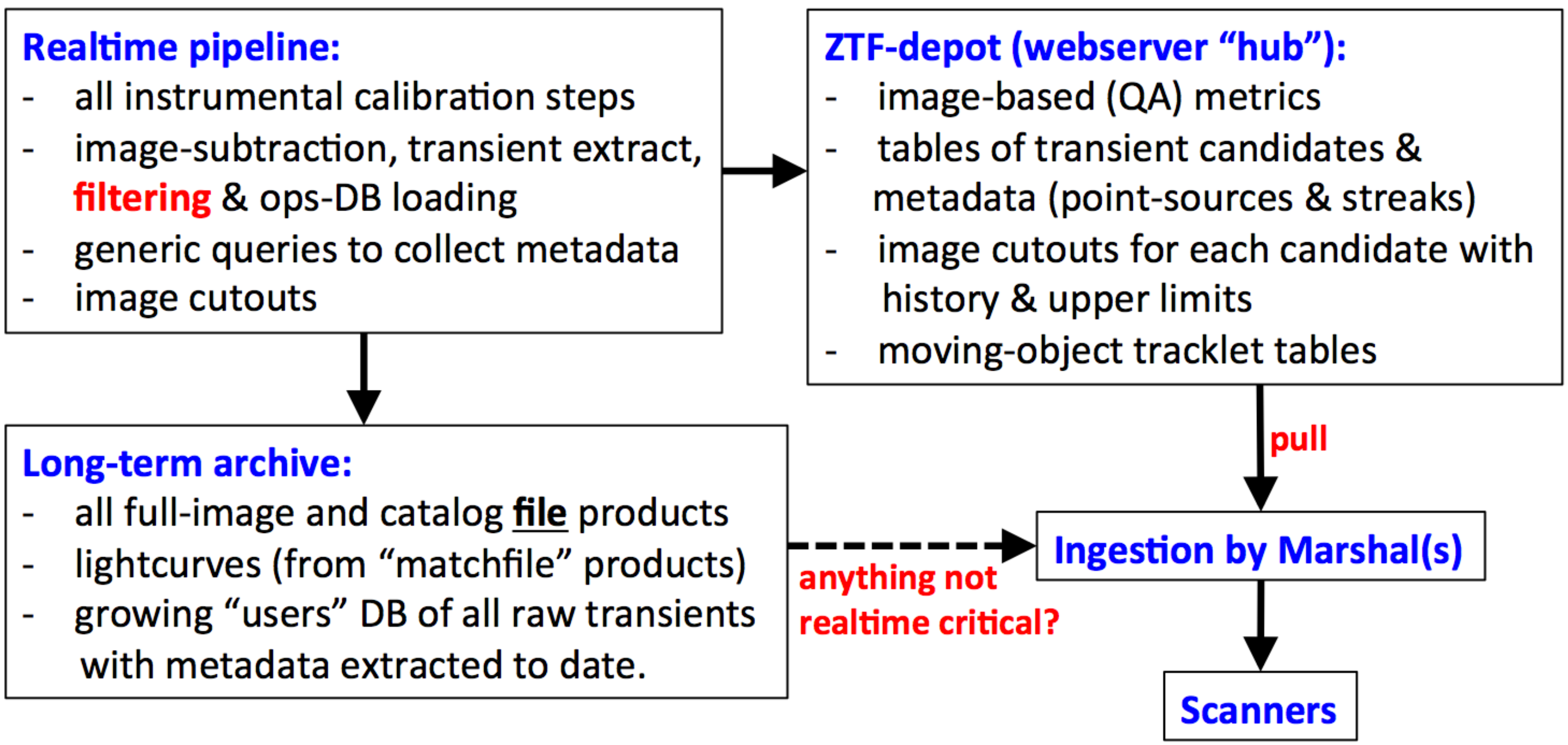
  with ~ 700 PTF exposures/night × 64 images × 20 candidates/image, ~ 890,000 candidates/night.

  ⬇

  Marshal automated-vetting for specific science use cases.

Need to maximize purity of "raw" transient stream

- Plan is to deliver a generic transient stream (following any automated RB vetting or filtering in pipeline) to a webserver for collection by all marshals.
- **Question:** do we need a machine-learned RB filter for ZOGY output? Can simple filtering work?



Development status as of today

green => in progress
red => not yet implemented or TBD

Data transfer study from P48

Pipelines:
- Ingest, quad-splitting, floating bias
- Calibration generation: biases, high-ν, **low-ν flats?**
- Instrumental calibration: **integration of PS1**
- Reference image generation
- **Source-matching/relative phot for lightcurves**
- Transient discovery: **ML/RB? streaks: fast movers**
- **Moving objects via source linking (tracklets)**

"ZTF-Depot" for realtime (marshal) products:
- transient cands/metrics
- image cutouts

(Public) long-term archive and user interfaces
- All image and catalog **file** products
- Lightcurve retrieval from "matchfiles"

Public (VO) transient alerts

Concerns and worries

- How will entire processing and archive system perform when all steps are integrated? Need to accommodate additional functionality in previous slide.

- For point-source and streak transients: whether Machine-Learned RealBogus systems are needed or simple filtering will suffice.

- Galactic plane processing performance: how will system respond? Simulations are in progress.

- Flat-fielding plan: whether low-spatial frequency responsivity maps are needed to achieve best *relative* photometric precision; currently a placeholder in ZTF pipeline.

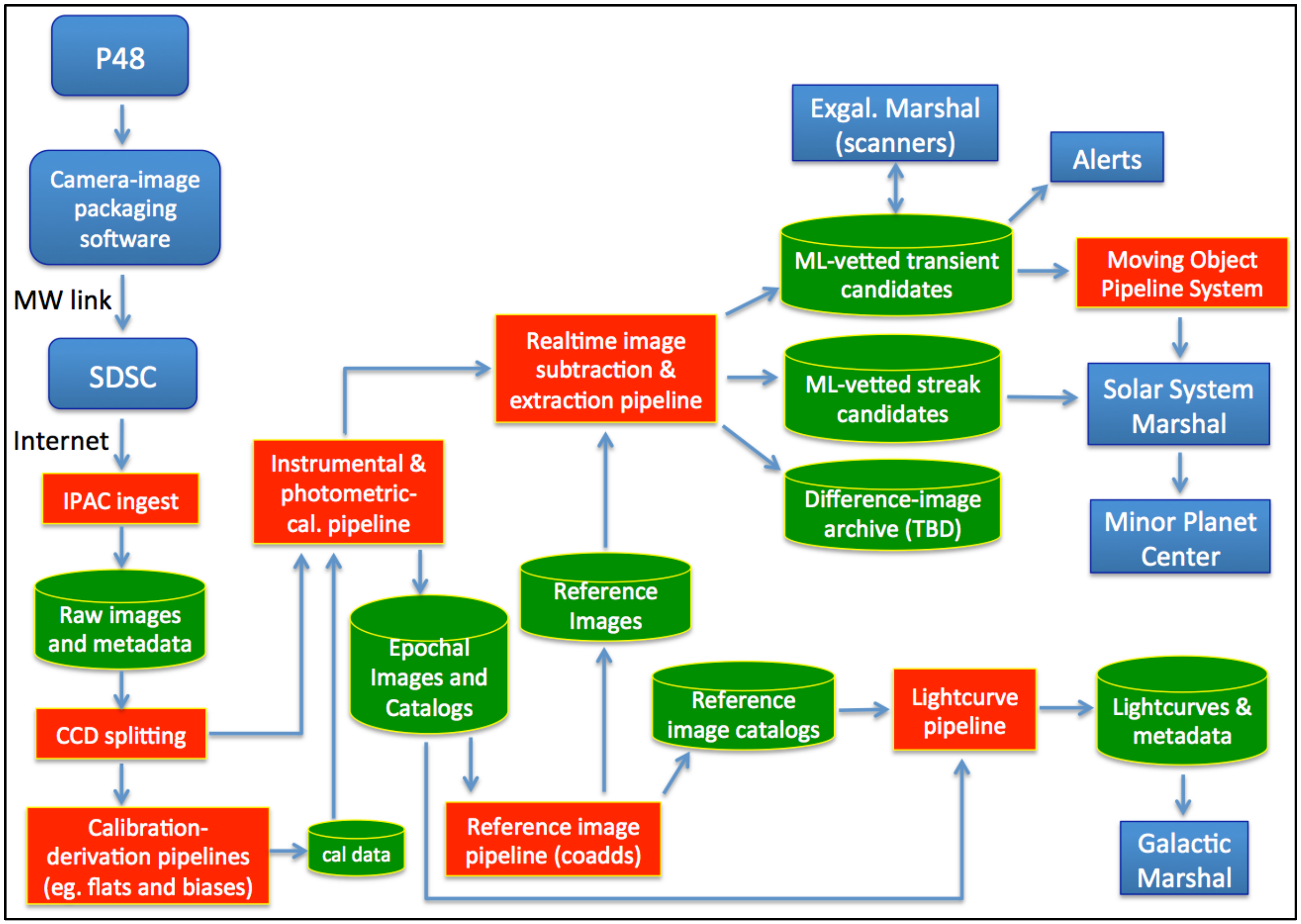
Collaboration visibility and assistance

- Within the next month: move all pipeline software and configuration/parameter files to GitHub, and link to ZTF-GitHub account. Now located in a local CVS repository
- Once archive system is in place, allow collaboration to access products (simulations for now)
- Includes “ztf-depot” portal hosting real-time products for marshal access
- Pipeline operations file-system: access limited to select individuals; contingent on compute-cluster availability
- New pipeline modules are welcome once identified; i.e., following more camera / optical system characterization in lab and from commissioning (Python, Perl, C, C++, Java, Fortran!, R, …)

Back up slides

Data Flow in the ZTF Science Data System (ZSDS)

- The ZSDS will be housed at the Infrared Processing and Analysis Center (IPAC), Caltech
- Consists of data processing pipelines (red), data archives (green), and user-interfaces



ZTF data product volumes / source counts

Per night:

Assuming average length of night at Palomar is ~ 8h:40m (summer: ~6h:20m, winter: ~ 11h), we expect ~ 700 camera exposures per night on average => 44,800 readout quadrant images.

- raw data (including calibrations): ~ 367 GB compressed (3x)
- instrumentally-calibrated epochal images, masks, and metadata: ~ 3.1 TB
- aperture photometry (epochal) catalogs: ~ 140 GB
  - ~ 310 million sources per night
- PSF-fit photometry (epochal) catalogs: ~ 44.8 GB
  - ~ 900 million sources per night
- image-subtractions and metadata ~ 2 TB **(OLD!)**

Total per night: ~ 5.65 TB

For three-year survey:

Assuming ~ 250 to 280 "good" nights per year (from PTF),

Total image/catalog file products: ~ 4.2 to 4.7 PB

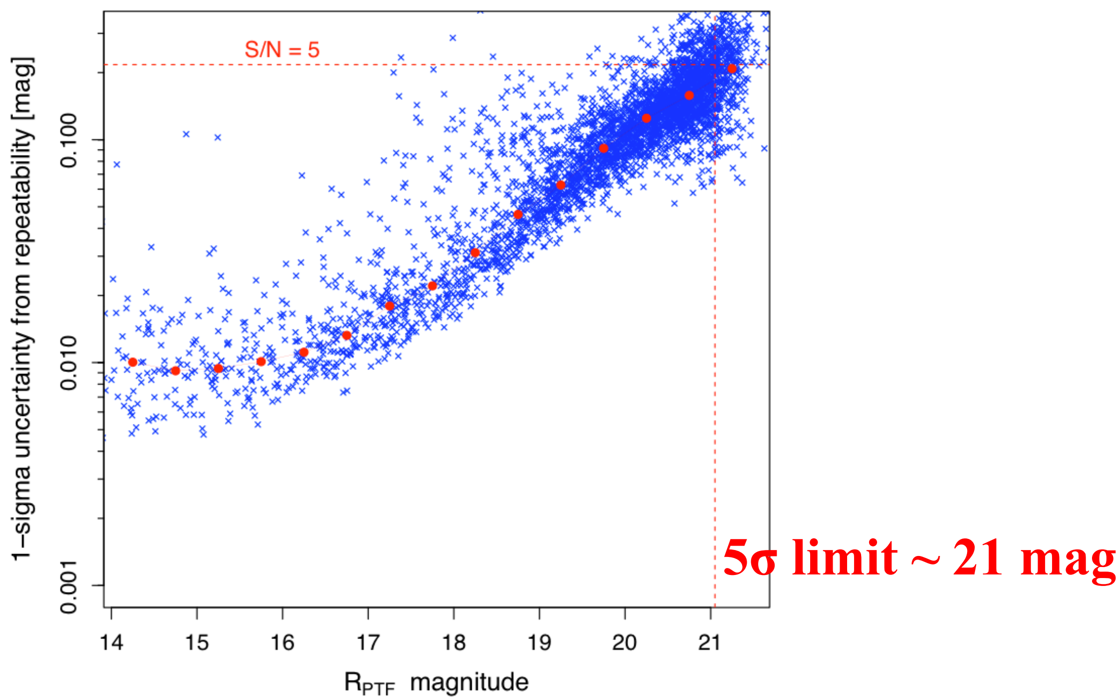
*** Includes storage of image-subtractions (not in baseline budget).
Excludes database storage for raw transients, other metadata, and epochal lightcurve database.

Basic Photometric Calibration

- Photometric calibration will be performed with respect to an external catalog (e.g., Pan-STARRS1) using PSF-fit extractions on a readout-quadrant image basis:

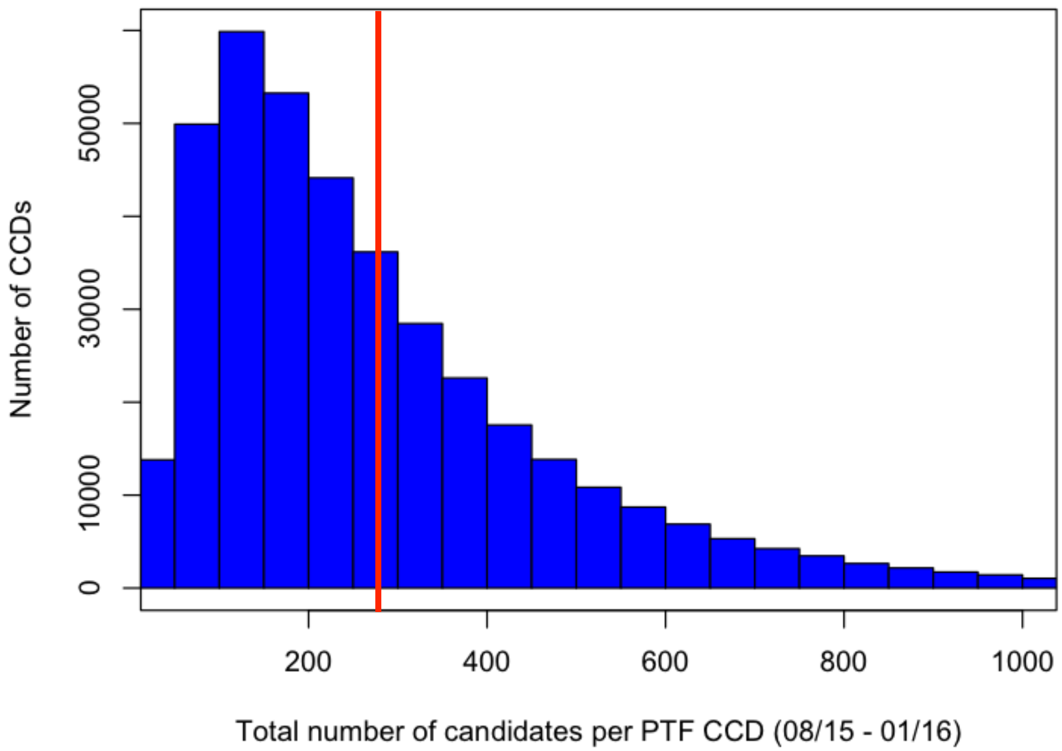
$$m_i^{PS} - m_i^{ZTF} = ZP + b\left(g_i^{PS} - R_i^{PS}\right) + \varepsilon_i \quad \Rightarrow \quad \text{solve for } ZP,\, b \text{ per image}$$

- Expect an *absolute* precision of ~ 2 - 3%.
- *Relative* photometric precision using PSF-fitting on PTF images ~ 1% (no refinement of ZPs across epochs)
  - ➢ Biggest limitation is flat-fielding!



Number of (raw) transient candidates

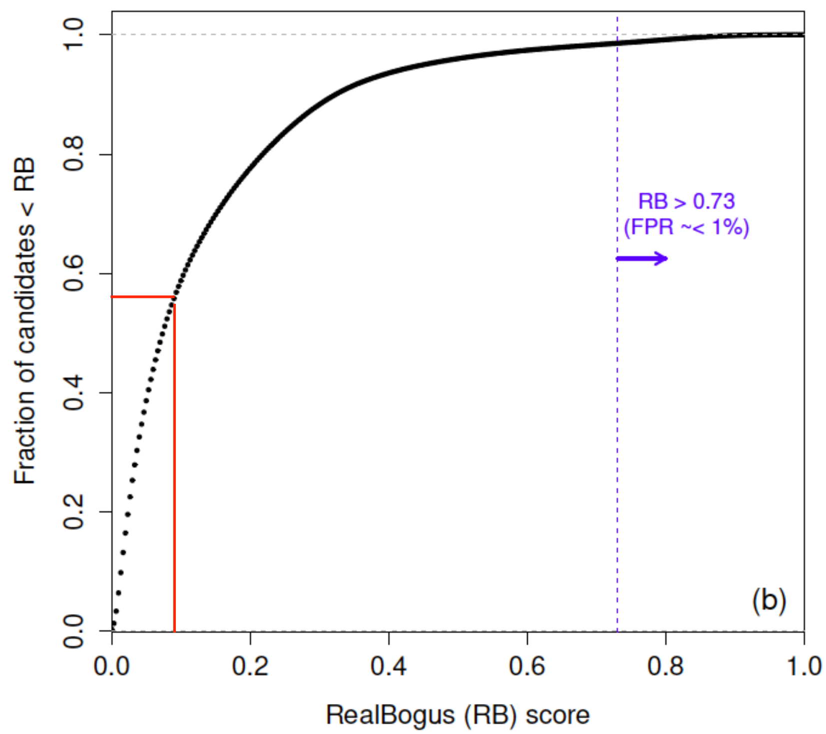
- From **PTF**, encounter ~ 260 raw, **non** machine-learned vetted candidates per CCD at > 4σ using PTFIDE.
- One ZTF CCD readout quadrant covers ~ one PTF CCD + ~ 10%. Hence we can extrapolate to ZTF.
- Have ~ 700 exposures * 64 readout quads: ~ 44,800 positive subtractions per night on average.
- Implies **~ 13 million transient raw candidates** per night for ZTF. Includes all transients (+ variables + asteroids)



or ~ per ZTF readout quadrant

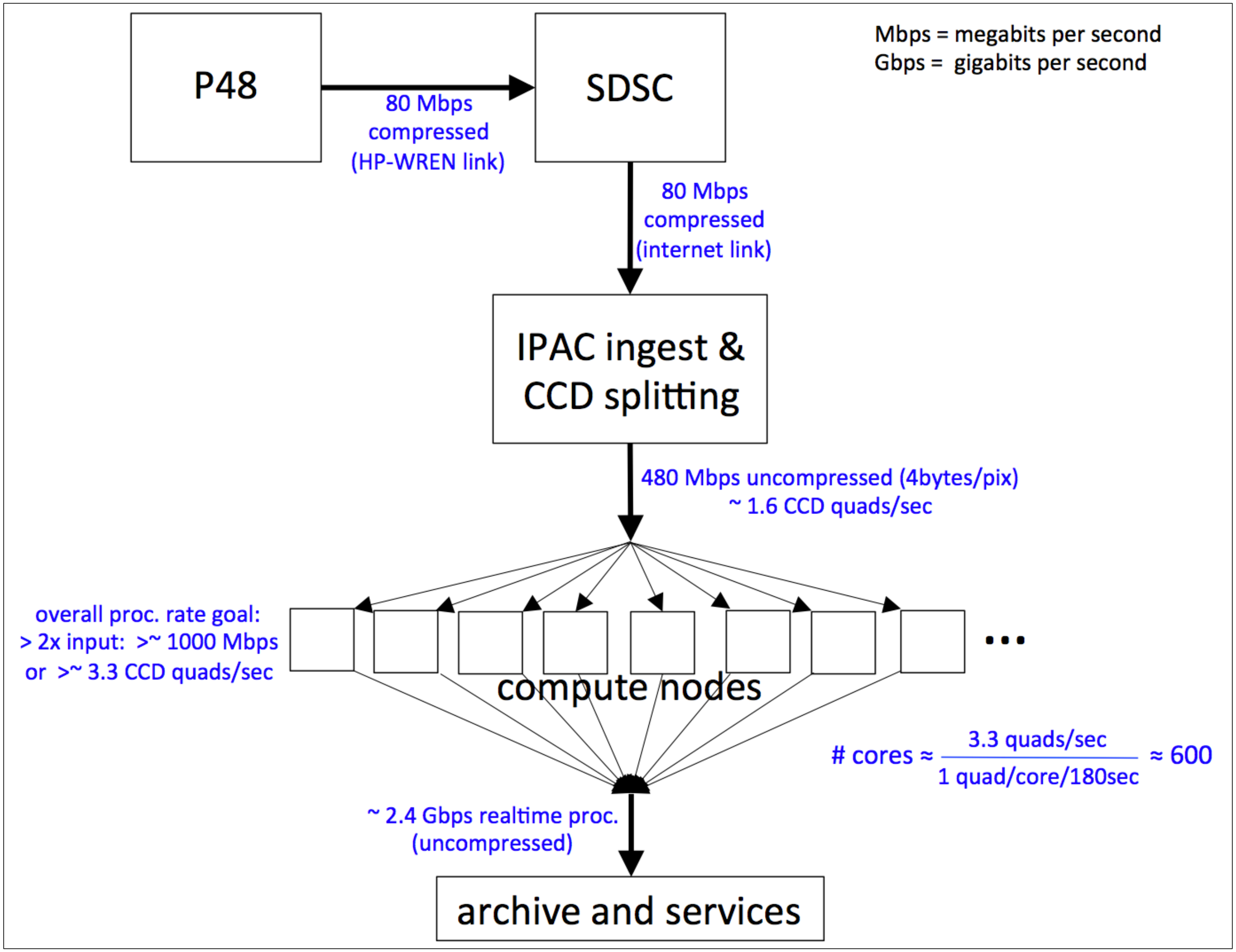
Benefit of Machine Learning

- Use the *RealBogus* (RB) quality score from a machine-learned classifier: crucial for PTF (down to 4σ).
- If avoid everything with a RB score < 0.1, only need to store ~ 6 million candidates per night in DB for ZTF.
- If use RB > 0.73 (< 1% false-positive rate) found for PTFIDE subtractions, need to scan <~ 400,000 cands/night.
- Translates to <~ 10 candidates per ZTF quadrant image or <~ 14 candidates/deg^2 on average (all transients).

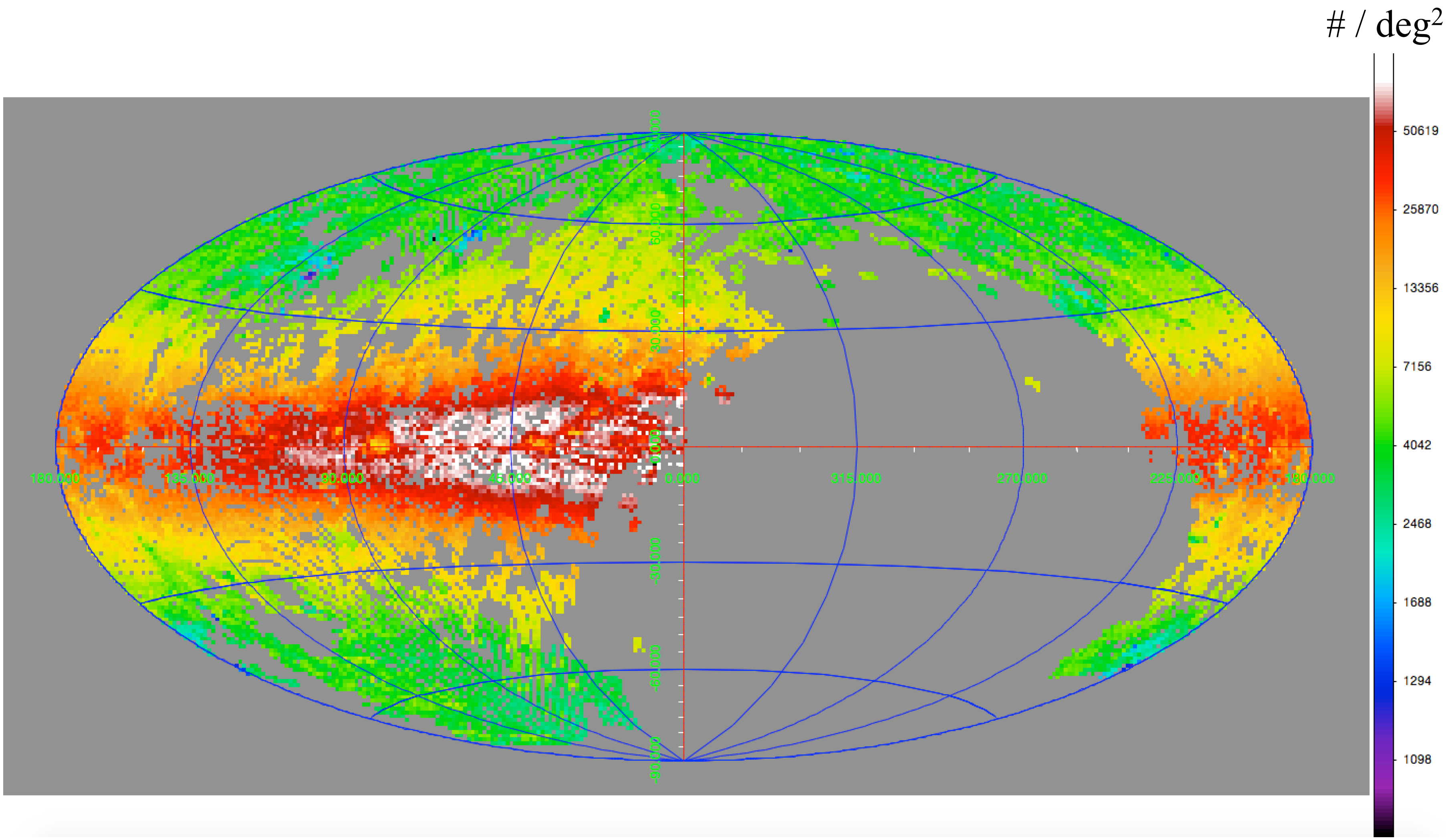


Cumulative fraction of transient candidates versus RB score from ~22,000 PTFIDE subtractions (Masci et al. 2016).

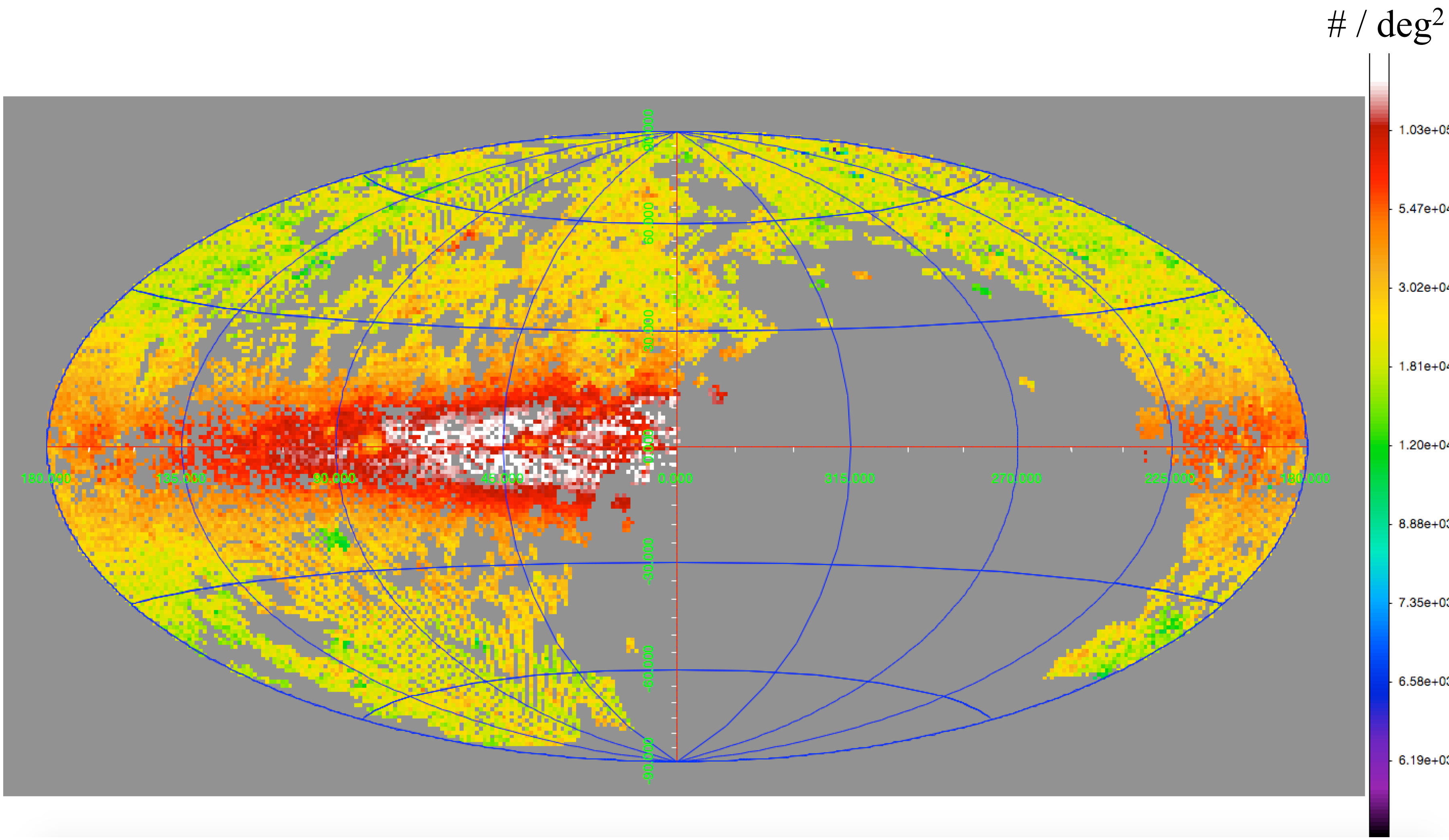
Cluster Processing Throughput



Density of aperture (SExtractor) extractions from PTF CCDs

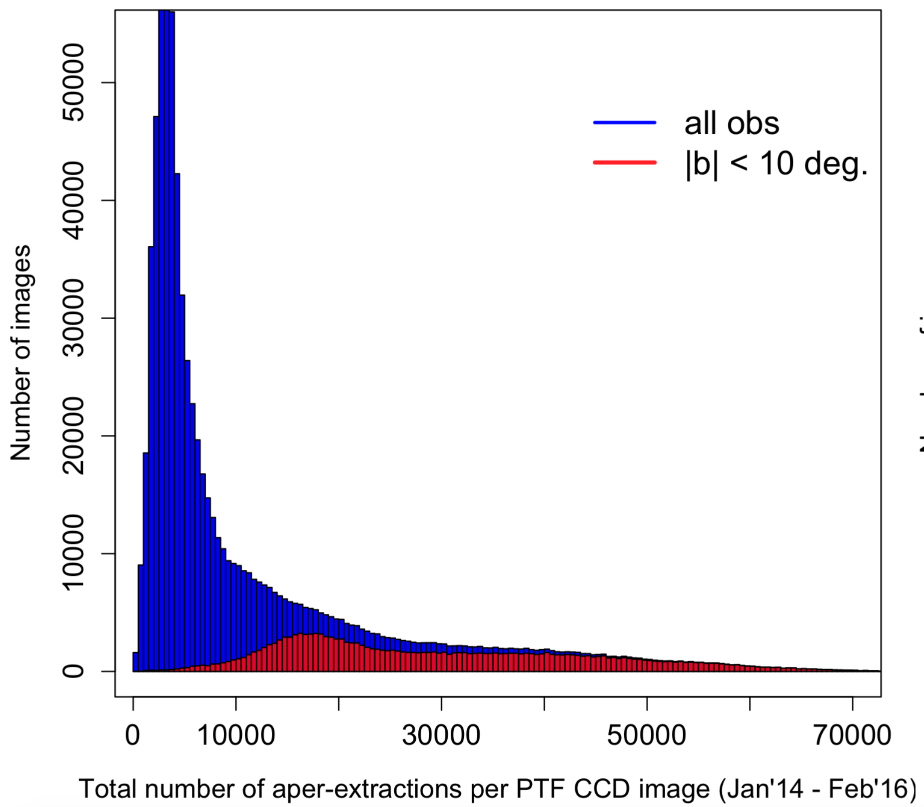


Density of PSF-fit extractions from PTF CCDs

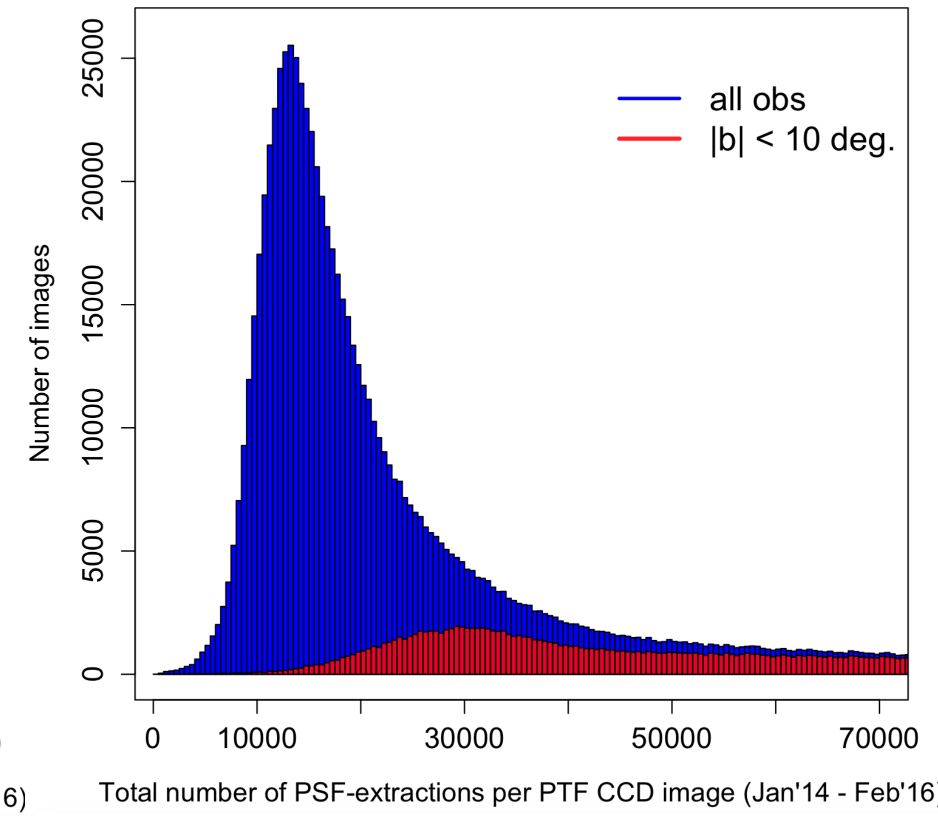


Number of sources extracted from PTF CCDs

Aperture (SExtractor)



PSF-fitting (DAOPhot)



Runtime breakdown in real-time pipeline

Instrumental Calibration:

```
elapsed time for storing and checking inputs [sec] = 0.068803
elapsed time to initialize sci-mask and apply image calibrations (bias, flats) [sec] = 0.622925
elapsed time to compute slowly-varying background, RMS images, and metrics [sec] = 17.390747
elapsed time to estimate saturation pixel value for processed science image [sec] = 0.604437
elapsed time to detect and mask aircraft/satellite streaks [sec] = 14.965530
elapsed time to compute pixel uncertainty image [sec] = 0.918172
elapsed time to execute SExtractor to generate FITS_LDAC catalog for astrometry [sec] = 2.061922
elapsed time to execute solve_astrometry.py [sec] = 25.404176
elapsed time to estimate spatially-varying PSF using DAOPhot [sec] = 4.186130
elapsed time to extract sources and perform PSF-fit photometry using DAOPhot/ALLStar [sec] = 7.149641
elapsed time to compute metrics on PSF-fit catalog [sec] = 0.003958
elapsed time to perform photometric calibration and compute associated metrics [sec] = 0.175211
elapsed time to update header of PSF-fit catalog and convert to FITS-binary table [sec] = 0.082048
elapsed time to execute SExtractor to generate final catalog [sec] = 2.454809
elapsed time to update header of SExtractor FITS-binary table [sec] = 0.048021
elapsed time to update mask image with SExtractor-detected sources [sec] = 0.158130
elapsed time to compute metrics on final SExtractor catalog [sec] = 0.477136
elapsed time to compute more QA metrics on science image [sec] = 2.273971
elapsed time to process InfoBits, assign processing status flag, update image headers [sec] = 1.98751
```

Image Subtraction and Extraction:

```
elapsed time for storing and checking inputs [sec] = 0.074510
elapsed time for computing some input image-based QA/metadata [sec] = 0.023964
elapsed time for setting up pixel mask [sec] = 0.538969
elapsed time for gain-matching sci and ref image pixels [sec] = 0.070595
elapsed time for resampling and interpolating ref-image onto sci-image grid using SWarp [sec] = 1.323505
elapsed time for setting up final effective bad-pixel mask from sci and resampled ref-image [sec] = 0.808826
elapsed time for computing slowly-varying background 'delta' image [sec] = 15.487246
elapsed time to match background-level variation in sci-image to that in resampled ref-image [sec] = 0.900749
elapsed time to prep and subtract background from sci-image and compute some metrics for PSF estimation [sec] = 1.330343
elapsed time to generate uncertainty image for sci-image [sec] = 0.903090
elapsed time to estimate PSF for science image [sec] = 3.823288
elapsed time to prep and subtract background from ref-image and compute some metrics for PSF estimation [sec] = 1.128204
elapsed time to generate uncertainty image for ref-image [sec] = 0.924011
elapsed time to estimate PSF for ref-image [sec] = 4.939668
elapsed time to execute py_zogy.py [sec] = 55.968457
elapsed time to rescale diff-image from ZOGY, apply mask and generate negative diff-images [sec] = 3.187851
elapsed time to compute QA metrics on final diff-image(s) [sec] = 1.494892
elapsed time for computing uncertainty image for final diff-image [sec] = 1.057311
elapsed time for checking input image InfoBits and diff-image quality for setting diff-image status flag [sec] = 0.000125
elapsed time to store ref-image PSF-catalog sources and mapping to x,y frame of diff-image [sec] = 0.151723
elapsed time to execute SExtractor on positive diff-img outputs [sec] = 1.526919
elapsed time to store SExtractor catalog, cross-match with ref-image catalog sources [sec] = 0.081730
elapsed time to find closest Solar System objects [sec] = 2.161086
elapsed time to generate Solar System object-only table file [sec] = 0.009866
elapsed time to execute SExtractor on negative diff-img outputs [sec] = 1.376591
elapsed time to store SExtractor catalog, cross-match with ref-image catalog sources [sec] = 0.076706
elapsed time to gather source metadata, compute PSF-fit photometry, filter, and write to text file for DB [sec] = 20.15198
elapsed time for executing findstreaks [sec] = 1.417615
elapsed time for computing more metrics/features for streak-candidates [sec] = 0.003217
```