ExEP Standard Definitions and Evaluation Team Update

Dr. Rhonda Morgan
Jet Propulsion Laboratory, California Institute of Technology
And The ExEP Standard Definitions and Evaluation Team

November 11, 2016

© 2016 California Institute of Technology. Government sponsorship acknowledged.

Outline

- A reminder of our charter: inputs, assumptions, outputs
  - Charter is signed (expected by face to face)
  - Show parts of charter
  - Charter is open to FIRS for direct imaging. Transit spectroscopy is out of scope.
- Live demo of EXOSIMS (brief)
  - Numbers flowing past
  - Plots of an ensemble
- Testing – how to have confidence in the code.
- Start of end to end cross validation with AYO.
  - We are going to get different answers. Here are some of the reasons. This highlights how different assumptions and rules and lead to different answers.
  - Det_time is not variable for EXOSIMS scheduler (currently)
- SAG13 products we will ingest as inputs.
  - Bins in = bins out
  - The idea of full distributions of populations with EXOSIMS
- What we need from STDTs
  - Mission rules and metrics are correlated
  - Characterization only vs. detection only are orthogonal missions. What does yield mean?

EXOSIMS Output

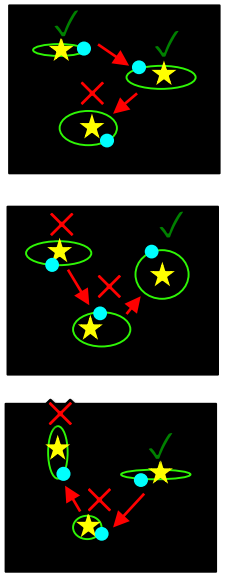
- WA < IWA

```
IWA 0.0415 arcsec
OWA 1.375 arcsec
14
det_WA [113.51881346375099]
det_status [1]
det_time 0.00556150383113
char_status [0]
char_time 0.00938436278057
28
det_WA [20.918988870726206]
det_status [-1]
det_time 0.429483809628
char_status [0]
char_time 0.0
29
det_WA [66.6611237028882]
det_status [1]
det_time 0.0294429523439
char_status [0]
char_time 0.0189686665264
54
det_WA [42.3832987149304]
det_status [0]
det_time 0.165818724898
char_status [0]
char_time 0.0
```

What is EXOSIMS?

https://github/dsavransky/EXOSIMS

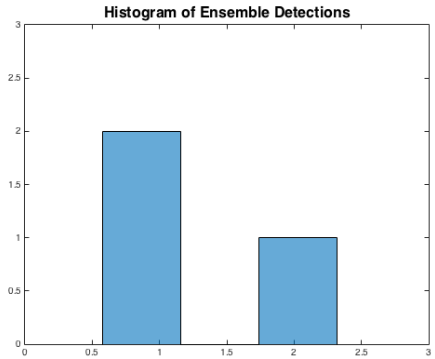
- EXOSIMS is a yield software architecture in which every model component can be independently upgraded/changed.
  - A Monte Carlo approach allows for dynamic constraints in the scheduling of the design reference mission (DRM), such as star shade fuel consumption and responses to detections.
  - Creates ensembles of DRMs which can be analyzed statistically.



2 Detections

1 Detection

1 Detection



D. Savransky, et al., JATIS 2(1) 2016

A Single DRM run

- EXOSIMS keeps track of time and lists the start time of a new observation. For this demo run the mission lasted 36 days.
  - The observing time was dominated by the 1 day optical overhead time.

```
Current mission time:  0.0 d
Current mission time:  1.00029026859 d
Current mission time:  2.00069482019 d
Current mission time:  3.00105532313 d
Current mission time:  4.00166443855 d
Current mission time:  5.00441644432 d
Current mission time:  6.00724411996 d
Current mission time:  7.01130793739 d
Current mission time:  8.01446345824 d
Current mission time:  9.0170708248 d
Current mission time:  10.0175504081 d
Current mission time:  11.019832506 d
Current mission time:  12.0199609663 d
Current mission time:  13.0217910626 d
Current mission time:  14.0218244354 d
Current mission time:  15.0245394898 d
Current mission time:  16.0248999928 d
Current mission time:  17.027371271 d
Current mission time:  18.0305267918 d
```

```
Current mission time:  19.0308170604 d
Current mission time:  20.0348808778 d
Current mission time:  21.0352854294 d
Current mission time:  22.0358945448 d
Current mission time:  23.0361938365 d
Current mission time:  24.0373381221 d
Current mission time:  25.0375798811 d
Current mission time:  26.0397738372 d
Current mission time:  27.0401783888 d
Current mission time:  28.0407068585 d
Current mission time:  29.040948612 d
Current mission time:  30.0416766025 d
Current mission time:  31.042820888 d
Current mission time:  32.0450148442 d
Current mission time:  33.0455433139 d
Current mission time:  34.046437223 d
Current mission time:  35.0472075165 d
Current mission time:  36.0482119218 d
Survey simulation: finishing OK
```

Single DRM output

- EXOSIMS does not (yet) automatically write out the data and generate analysis graphs for the user.
- The results of the run and all parameters are stored as python objects (mix of arrays, dictionaries, lists , etc.)

1st star visited

Index of planet
In list of synthetic planets

```
In [13]: sim.SurveySimulation.DRM
Out[13]:
[{'arrival_time': 0.0,
  'det_int_time': 1.00029026858701 3,
  'det_status': 0,
  'plan_inds': array([], dtype=int64),
  'target_ind': 308},
 {'arrival_time': 1.00029026858701 3,
  'det_int_time': 1.0004045516067452,
  'det_status': 0,
  'plan_inds': array([], dtype=int64),
  'target_ind': 291},
```

Detection integration time (days).

Detection status:
0 = null detection
1 = detection
-2 = false alarm
-1 = missed detection

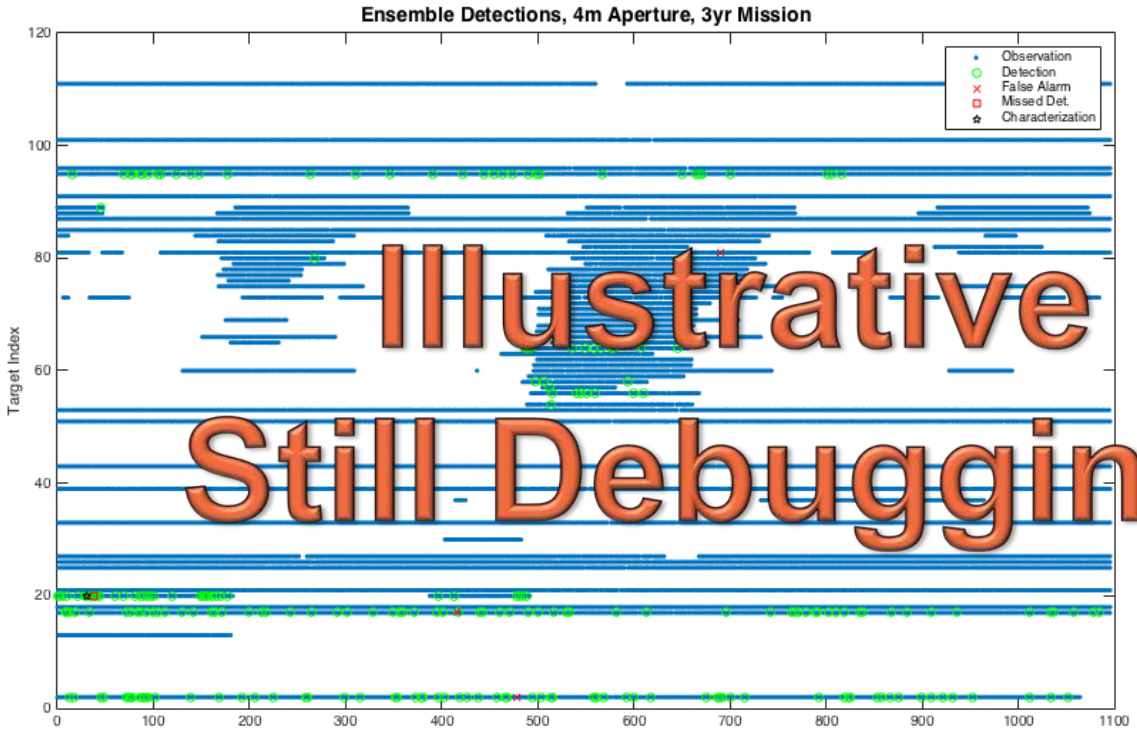
Index of star used in target list
(TargetList.Name)

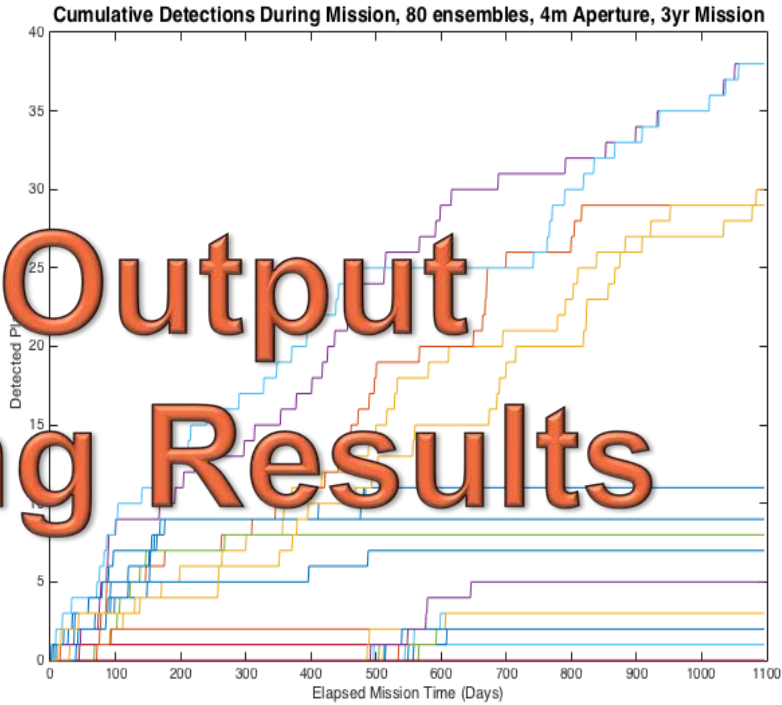
Ensemble of DRMs Output

```
In [36]: res = sim.SurveyEnsemble.run_ensemble(run_one, 80)
Submitted tasks:  80
Completed in 158805 sec
```

- The ensemble output is a list of DRMS:

  Sim.SurveySimulation.DRM[:]



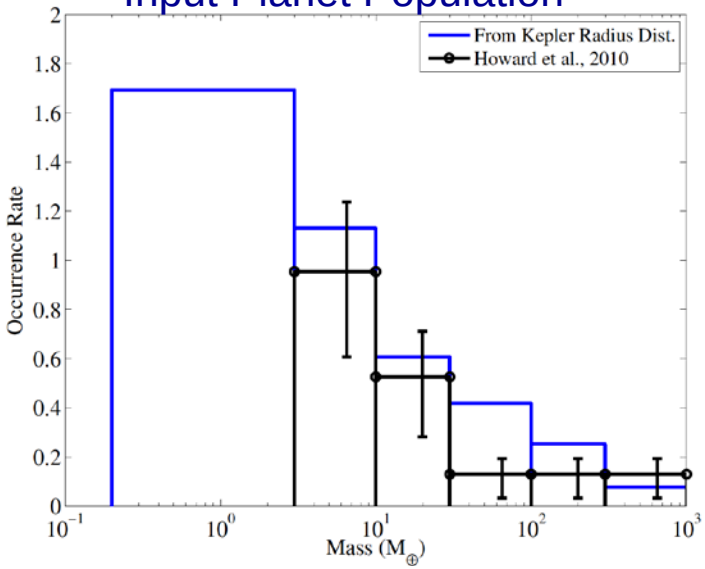


Illustrative Output

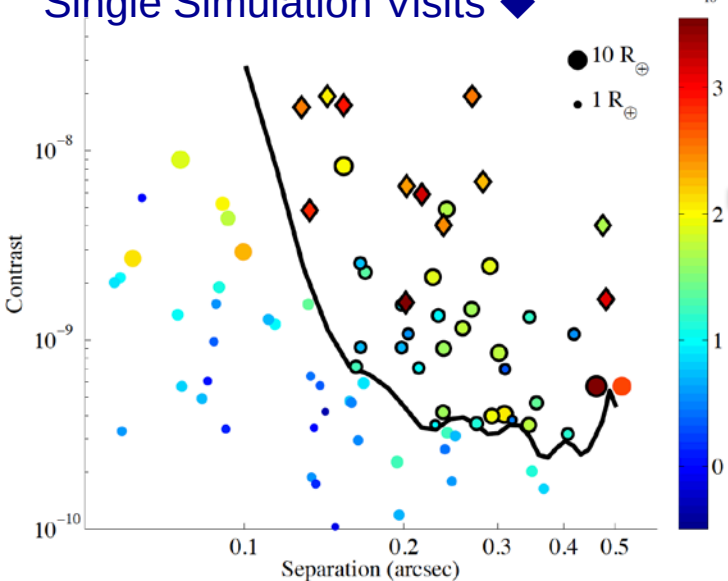
Still Debugging Results

WFIRST Yield Results by EXOSIMS

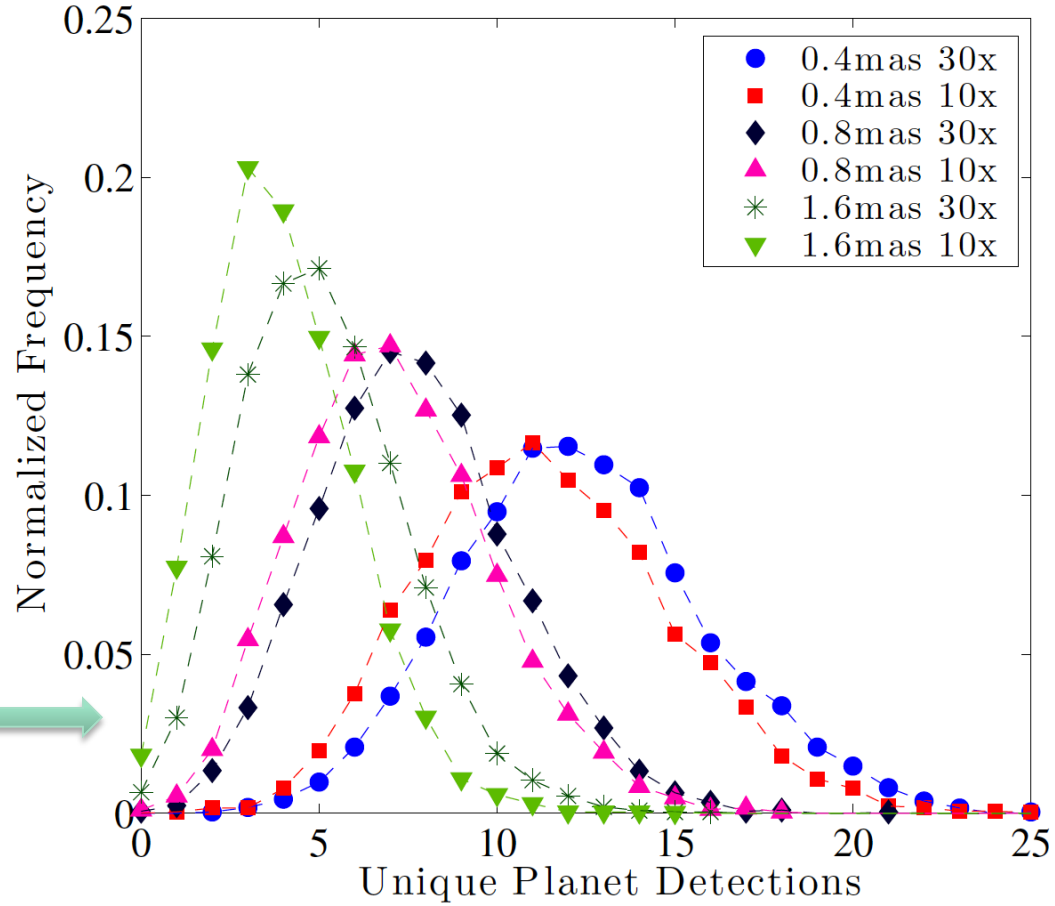
Input Planet Population



Single Simulation Visits ◆

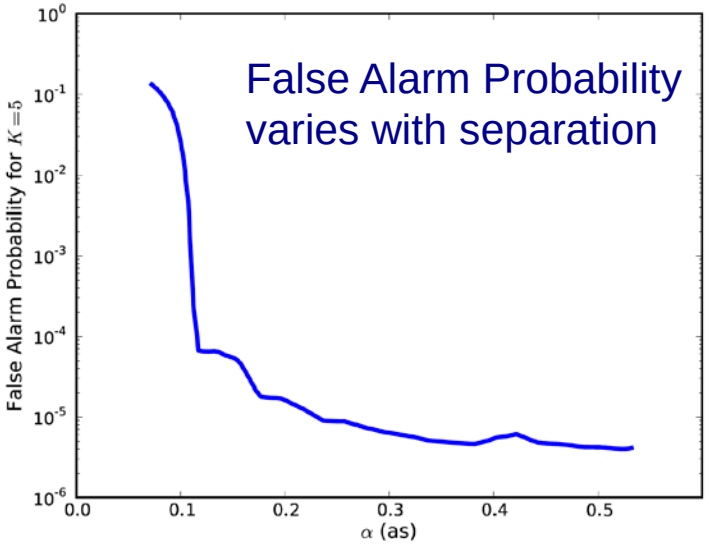


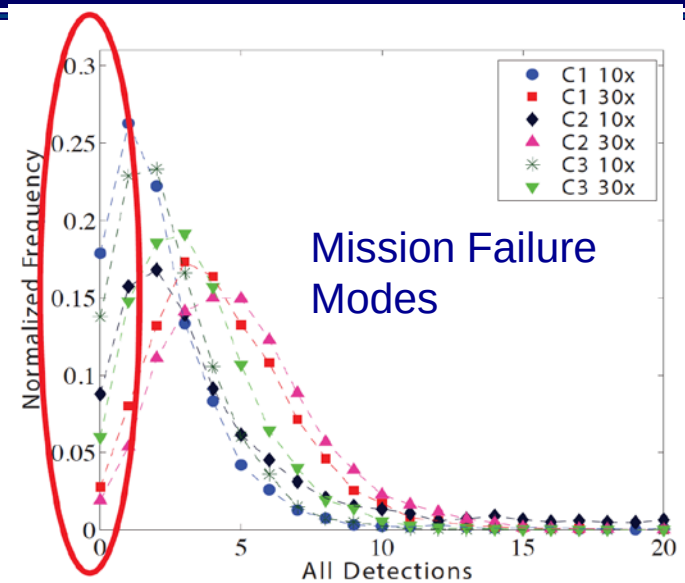
Simulation Ensemble Yield PDFs

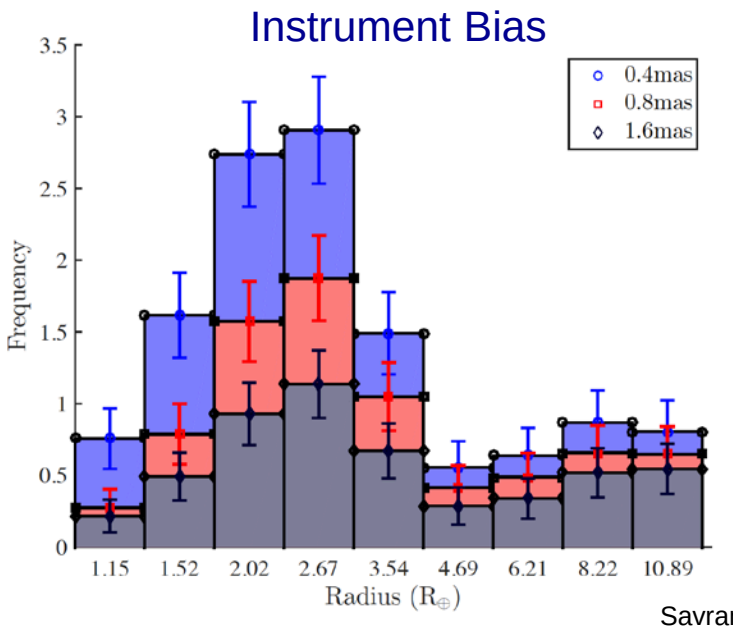


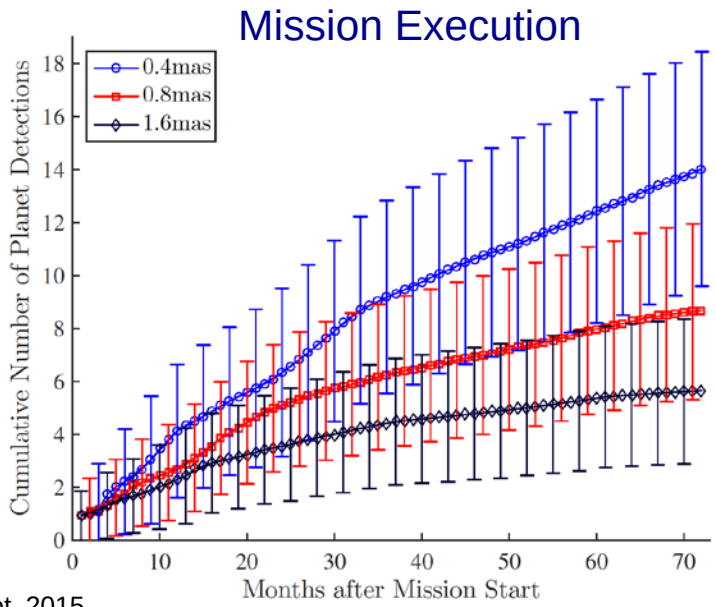
Savransky and Garret, 2015

The power of ensemble statistics









Savransky and Garret, 2015

ExEP Yield Input Parameters

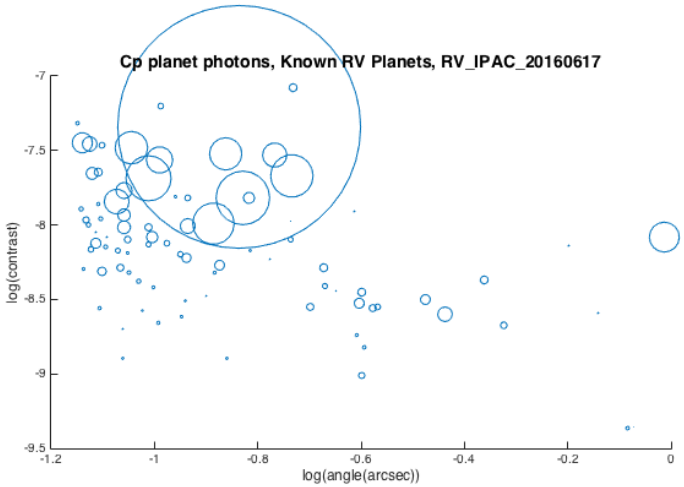
	Input Value (1st run)	Default	Notes
Astrophysics			
type of planets		EarthTwin	uniform random, keplerLike, KnownRV
eta_earth	0.1	0.16	
planet radius		1 earth radius	
planet mass		1 earth mass	
geometric albedo		0.2	Lambert phase function is used
semi-major axis		[0.75, 1.5] AU	uniform random distribution
eccentricity		[0,0.35]	uniform random distribution
cos(inclination)		[-1, 1]	uniform random distribution
argument of perigree		[0, 360] degrees	uniform random distribution
ascension of the ascending node		[0, 360] degrees	uniform random distribution
Zodical Light model		Stark	local zodi, interpolates for longitude
exozodi mean and variance	mean=3, var=0	22 mag/asec^2	no spatial variance. Variance is log normal star to star.
Telescope			
Diameter of pupil	study parm		
obscuration factor	0	0.2	due to spiders, secondary (let's assume off-axis primary)
throughput	0.6	0.57	Telescope only, not coronagraph
telescope keep out angle for internal coronagraph	60	45 degrees	sun avoidance angle
Detection lambda		550 nm	
detection bandwidth	0.15	0.2	delta_lambda/lambda
Spectra central lambda	1000	550 nm	
spectral BW		R 70	
SNR detection	7	5	Fixes false alarm probability (assuming Gaussian noise) - Dimitri
SNR characterization	7	10	Fixes false alarm probability (assuming Gaussian noise) - Dimitri
limiting deltamagnitude		26	no source fainter than m* + that limit will be detected
Starlight Suppression Systems			
Internal Coronagraph type	N/A	HLC	
IWA (only fixed by D, NOT by lambda)	103mas*(4m/D)	2.5 lambda/D	That is 2l/D at 1000nm and 3.64 lambda/D at 550nm
OWA (is a fixed number of lambda/D)	20 lambda/D	10 lambda/D	At all wavelengths lambda (fixed by nb of DM actuators)
Contrast	10^-10		Raw contrast, assumed constant btw IWA and OWA (what is raw c
Throughput for Coronagraph	0.4		# polarization channels? - Dimitri
Optical Overhead time	0.5 day	1 day	
Detectors			
dark current	0.0001 count/pix/s	0.0009 e-/s	defaults are for WFIRST EM-CCD modeling
readNoise	0.1 count rms/pix	3 e-	
exposure time per frame		1000 seconds	will set the number of reads (amd minimum exposure)
QE	0.8	0.88	
Post Processing			
post-processing contrast factor	0.1	1/10	input can be [0,1]. 1/10 and 1/30 are common
allowable false alarm probability		3.00E-05	Used in Kasdin-Braems method, from 2006 paper, eqn 33
allowable missed detction probability		1.00E-03	Used in Kasdin-Braems method, from 2006 paper, eqn 33
Mission			
mission lifetime		5 years	
fraction to exoplanets		0.5	
start date		January 1, 2036	

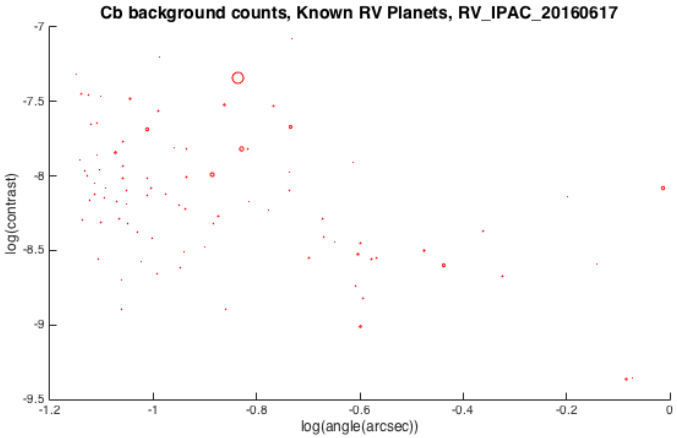
How do we know the code is right?

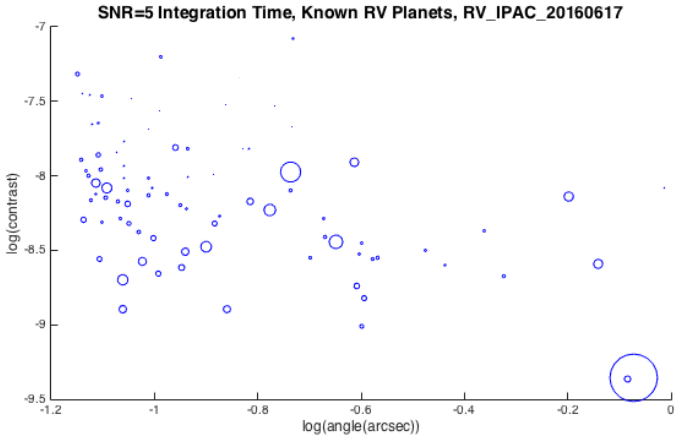
- Validate against other codes
  - Many options exist: Stark, Traub, Nemati
  - A good top level comparison of results at the finish line
  - Compares Monte Carlo outputs on complex codes
  - Many assumptions (~100) in yield code drive the results
- Point-check the physics
  - A good grassroots, bottom up approach
  - Done in smallest chunks of code sensible (unit tests)
  - Each test is tailored to that unit of code
    - Some test software things
    - Some test physics results against expected values

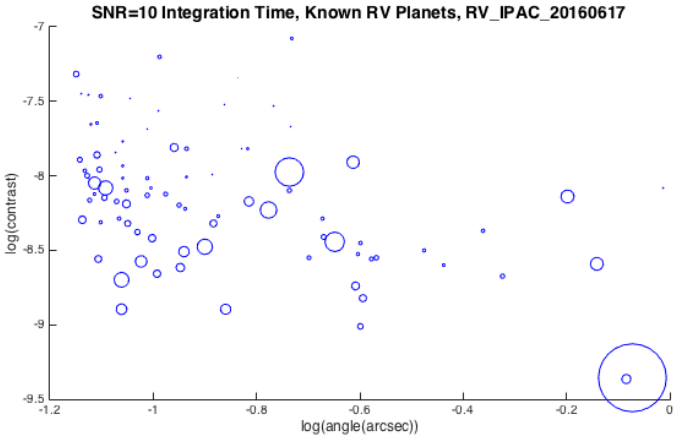
Test Case for C_p, C_b, T_{int}

- Use Known RV planets from IPAC portal
- Validated Matlab code computes photon counts from planet and background and detector noise sources and calculates integration time









Two (Three?) Test Levels

- **Validate the science calculation**
- **Validate software function**

EXOSIMS: 3850 lines of source code and 208 functions

Tests: 3450 lines of source code and 237 functions

1. **Unit Tests**
   - Each function within each module is tested
   - Test code verifies operation of function across all its inputs
   - Unit tests have 75% coverage of EXOSIMS source code
   - Dozens of bugs have been identified and fixed

1.5 Subsystem-level tests? – completeness, keepout

1. **End-to-End** Cross Validation
   1. Chris Stark's AYO code using standard inputs (ExoCat starlist, SAG13 occurrences) for Decadal Mission Concept Studies

Typical Testing Strategies

- Numerical checks with other implementations
  - Solar system ephemeris are checked vs. an independent Matlab implementation using JPL Spice kernel
  - Stellar proper motion is point-checked with an independent calculation
  - Random number distribution checks, where possible *verify*
- Numerical checks versus tabulated data
  - Fortney Marley Cahoy planet physical model checks against full tabulations of planet properties from the original ApJ papers. *specify*
  - Known RV Planets universe checks exhaustively against IPAC catalog of stars and planets (584 planets, about 35 attributes)
- Functionality checks
  - Object constructors must set all properties
    - OpticalSystem sets ~30 parameters, all of which are checked individually
  - Target filtering works across all properties (mass, dMag, ...)
  - Verify object type, units, size, range matches constraints

Specific Contributions

Perhaps not helpful for science audience in current form, could be extended and briefed to sponsors. Another strategy would be to show-by-example of fixing problems in keepout.

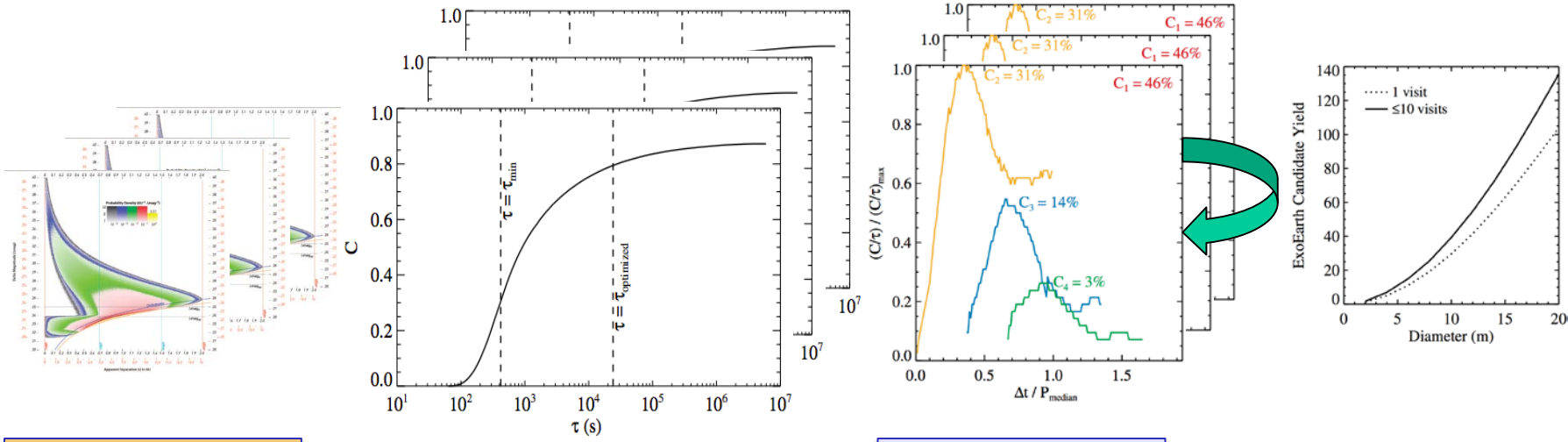
- Finding bugs
  - Solar system ephemeris
  - Keepout
  - L2 halo orbit
- Revealing issues (problems that are not outright bugs)
  - Cached completeness result could lead to mysterious bugs
  - More detailed reporting of simulation results
- Changing method of implementation
  - Ephemeris now uses JPL Spice kernels rather than Kepler elements
- Improving repeatability
  - Use of one overall random number seed allows end-to-end repeatability
  - DeterministicUniverse class

STATUS AND PLANS

- Going-forward strategy
  - Contributed much of the test code (1100 LOC / 33%) back into EXOSIMS
  - Rest in pipeline to be delivered
  - Test code will be posted to github EXOSIMS repository
  - Hope that Cornell team will integrate the tests into its deployment process
  - Through testing we think ~80% of functionality can be trusted to work, about 10% is error-handling, etc., and unimportant, and about 10% is scheduler which can be tested through end-to-end runs

Chris Stark's AYO Approach

Completeness(T_{int})



Monte Carlo Planets around stars → Instrument → Optimize Integration Time Alloc. → Yield

Planet Classes

Star list

AYO: Altruistic Yield Optimization