ExEP Yield Modeling Tool and Validation Test Results

Dr. Rhonda Morgan[1]
rhonda.morgan@jpl.nasa.gov

Dr. Michael Turmon[1], Dr. Dmitry Savransky[2,3], Dr. Christian Delacroix[2,3], Daniel Garrett[2], Dr. Patrick Lowrance[4], Xiang Cate Liu[4], Dr. Paul Nunez[1]

[1]Jet Propulsion Laboratory/California Institute of Technology; [2]Space Imaging and Optical Systems Laboratory, Cornell University; [3]Carl Sagan Institute, Cornell University; [4]IPAC, California Institute of Technology

August 10, 2017

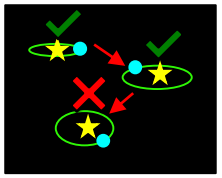
© 2017 All rights reserved. Government sponsorship acknowledged.

What is EXOSIMS?

https://github/dsavransky/EXOSIMS

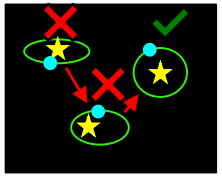
- EXOSIMS
  - Open source. Python. Parametric. Probabilistic. Modular.
  - Creates ensembles of DRMs which can be analyzed statistically.

Universe n



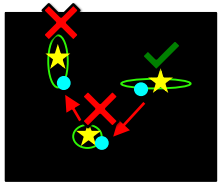
2 Detections

Universe n+1

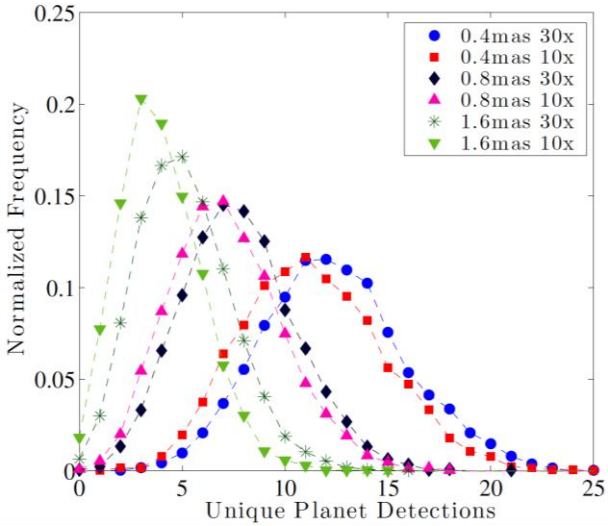


1 Detection

Universe n+2

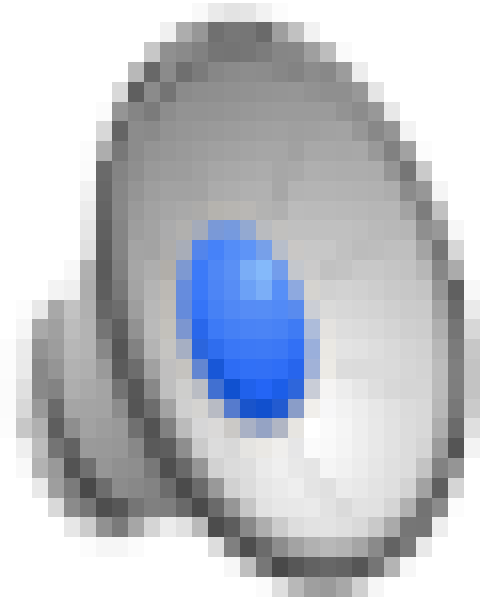


1 Detection



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

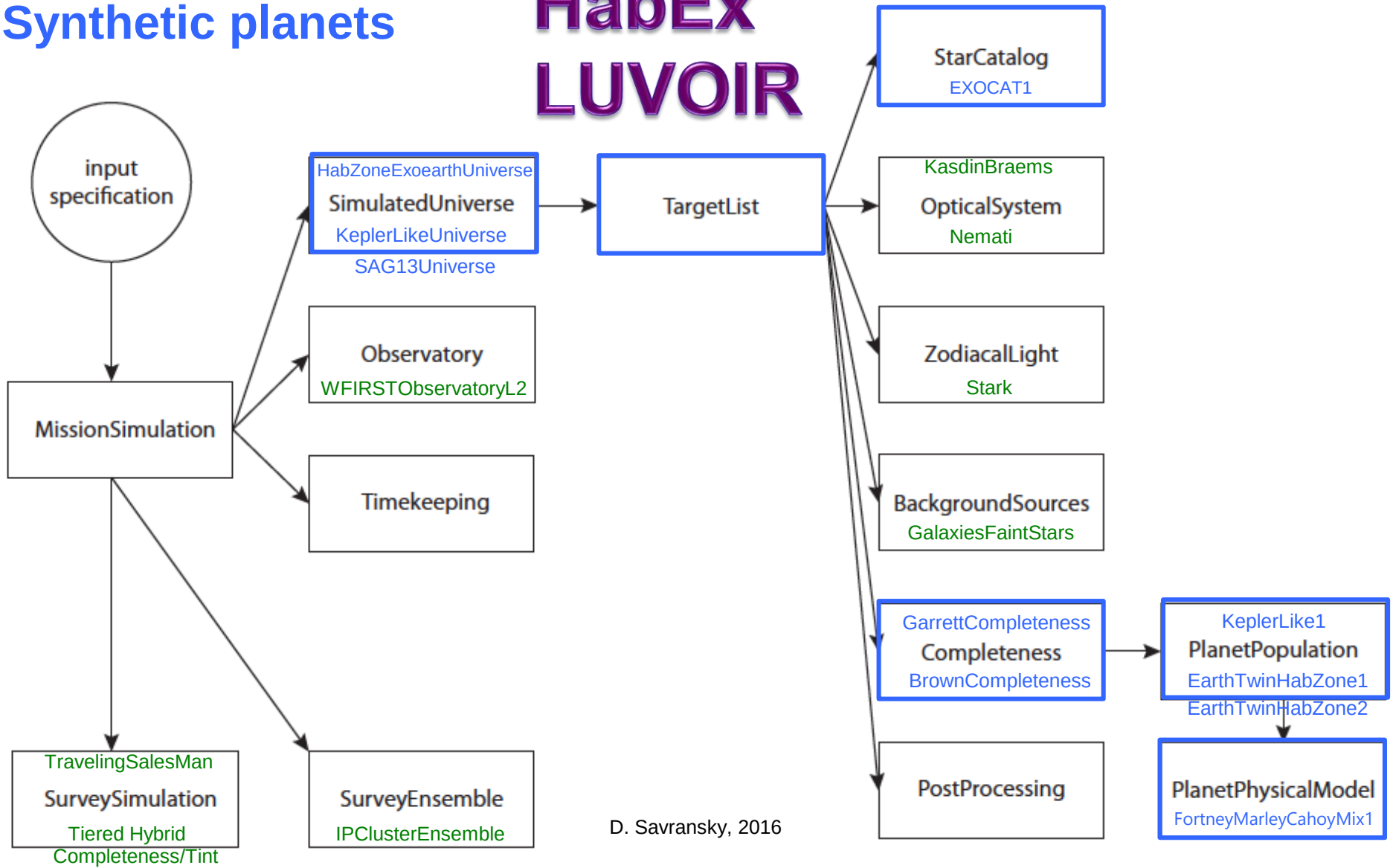
Sample DRM – Hybrid Starshade Coronagraph



EXOSIMS Architecture

Synthetic planets

HabEx
LUVOIR

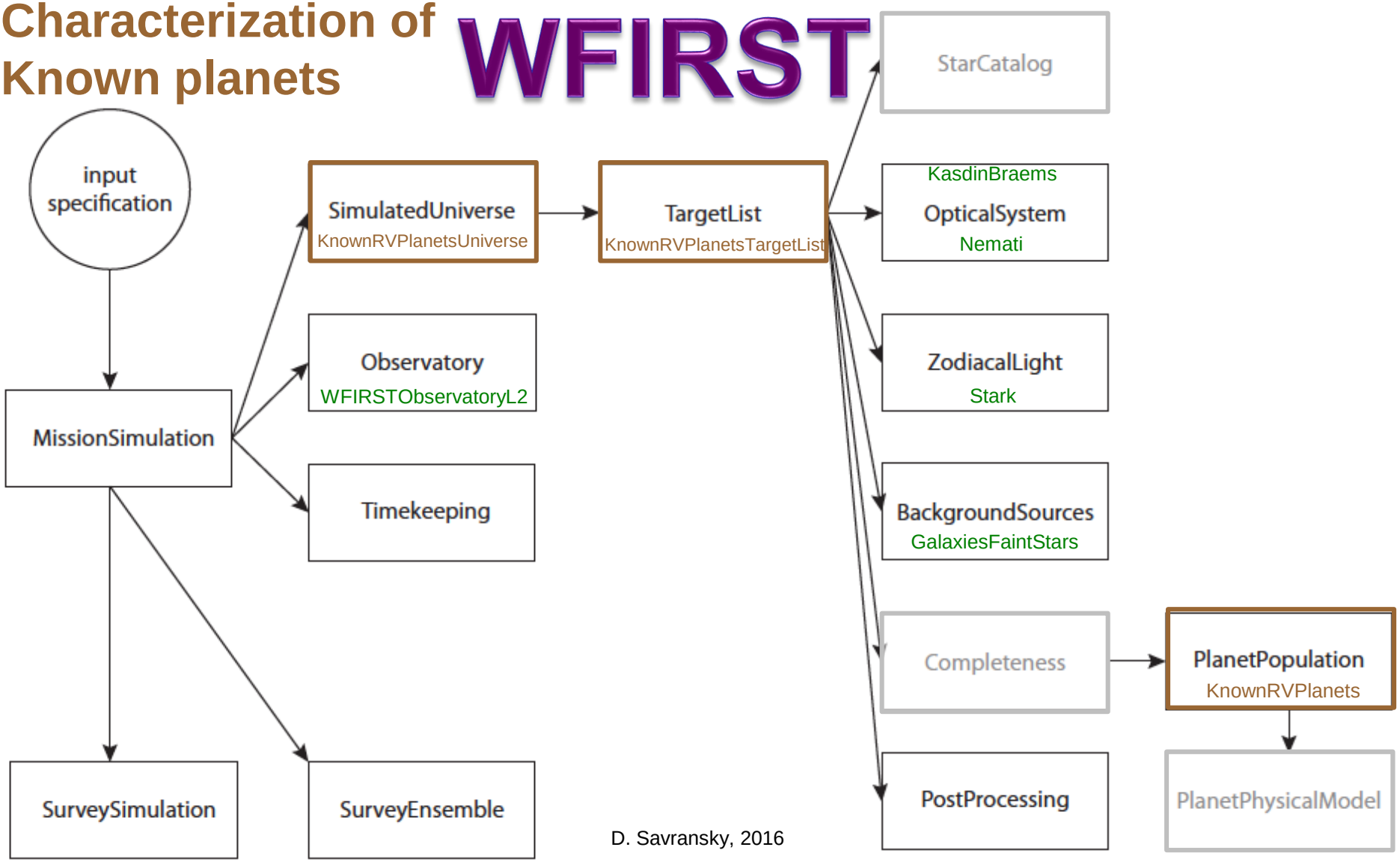


D. Savransky, 2016

EXOSIMS Architecture

Characterization of Known planets

WFIRST



D. Savransky, 2016

Inputs

- https://docs.google.com/spreadsheets/d/1-w1eybYV6nyX02kH7eoNWie8UinzlVnLM4F-64AagHw

ExEP Yield Input Parameters		
	Input	Notes
Astrophysics		
eta_earth	0.5	SAG-13 Integrated over box0.5< Rp 1.5 earth radius
planet radius	[0.5, 1.5] earth ra	log uniform random distribution
planet mass	1 earth mass	log uniform random distribution
geometric albedo	[0.2, 0.2]	Lambert phase function is used, uniform random distribution
semi-major axis	[0.75, 1.5] AU	log 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
exozodi mean and variance	23 mag/asec^2	constant, corrected for Mv

Telescope

Diameter of pupil	4m, 6x4m, 6m	
obscuration factor	0	off-axis primary
throughput	0.6333	Telescope only, not coronagraph
Detection lambda	550 nm	
detection bandwidth	0.2	delta_lambda/lambda
SNR detection	5	
limiting deltamagnitude	26	
minimum completeness	0.1	

Inputs cont.

Occulters

Internal Coronagraph type	HLC	
IWA (only fixed by D)	2 lambda/D	
OWA	20 lambda/D	
Contrast	1.00E-10	
core throughput	2.00E-02	
core mean intensity	4.00E-14	
Throughput for Coronagraph	0.4	occ_trans for zodi transmission

Detectors

imaging or spectrometer	imaging	
dark current	5.00E-04	defaults are for WFIRST EM-CCD
readNoise	0.007 e-	
exposure time per frame	1000 sec	
QE	0.91	
pixelpitch	5.00E-06	
focal length		FHWM sampled by 3.3 pixels
CIC	3.00E-03	
ENF	1	
EM CCd gain	1	

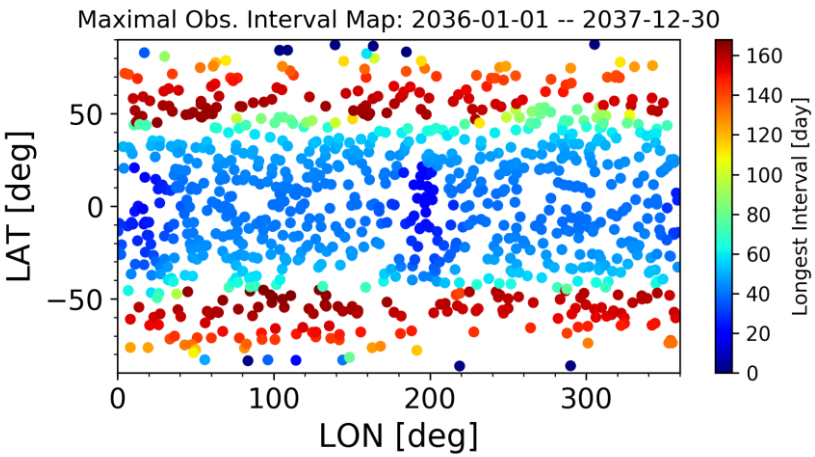
Post Processing

post-processing contrast factor	1/10	input can be [0,1]. 1/10 and 1/30 are common

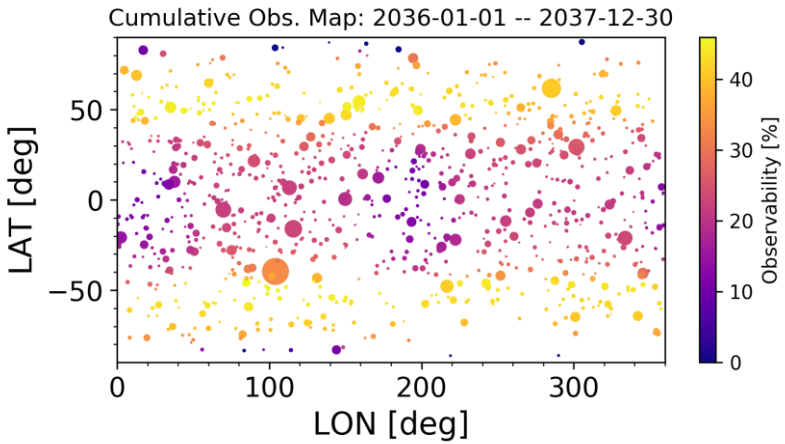
Mission

mission lifetime	3 years	
fraction to exoplanets	1	

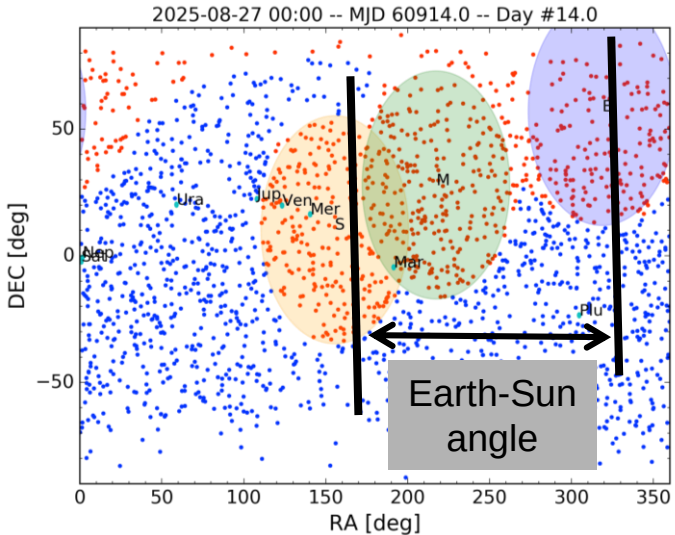
Starshade Sky Coverage Map



largest continuous interval of observability for each target (1-day granularity)



Cumulative fraction-of-time target is observable
Size of point is proportional to target Vmag.



- Two year mission: 2036-2037 inclusive
- dMagLim = 26.5
- Sun keepout: > 45˚ and < 83.4˚ is OK
- Earth keepout: >35˚ is OK
- Moon keepout: >15˚ is OK
- Other bodies: > 1˚ is OK

TESTING

Three Test Levels

EXOSIMS: 3850 lines of source code and 208 functions

Tests: 3450 lines of source code and 237 functions

1. Unit tests

- Test each important function within each module (75% coverage)
- Test code verifies operation of function across all its inputs
- Dozens of bugs have been identified and fixed

2. Integration testing

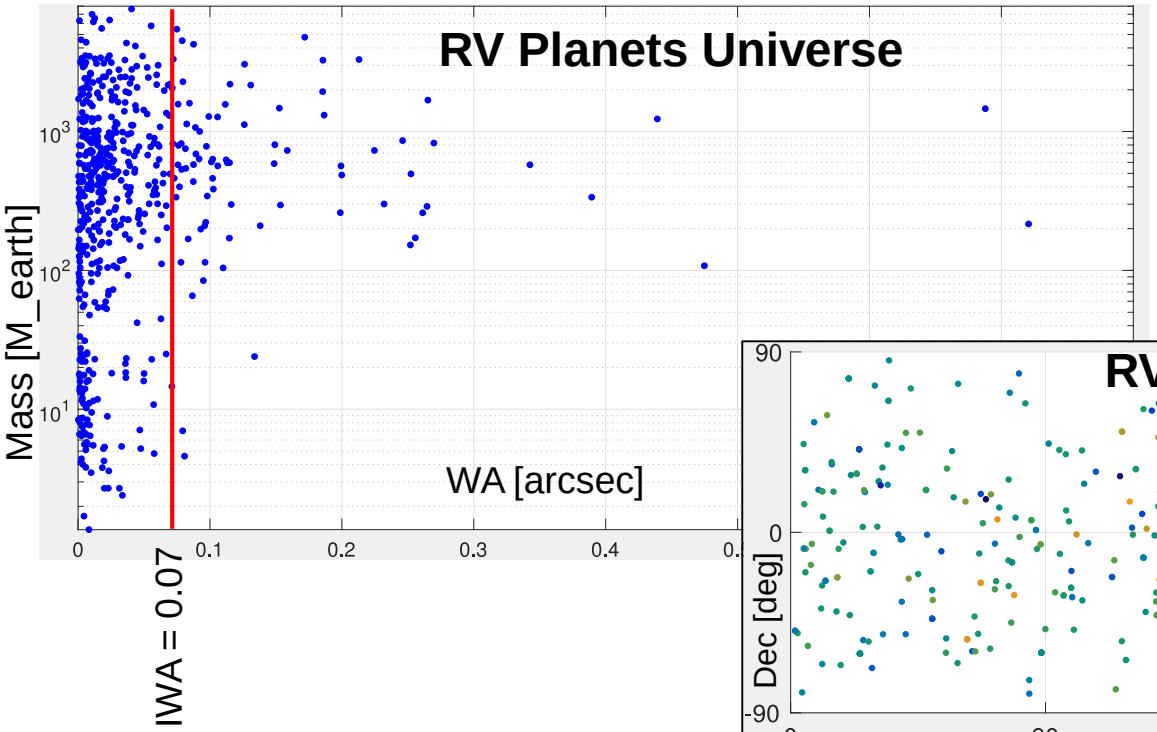
- Verify correctness of multiple functions and data sources
- Done for keepout, photometry, completeness
  - E.g., for keepout: spacecraft position, planetary ephemeris, orbit propagation simultaneously tested

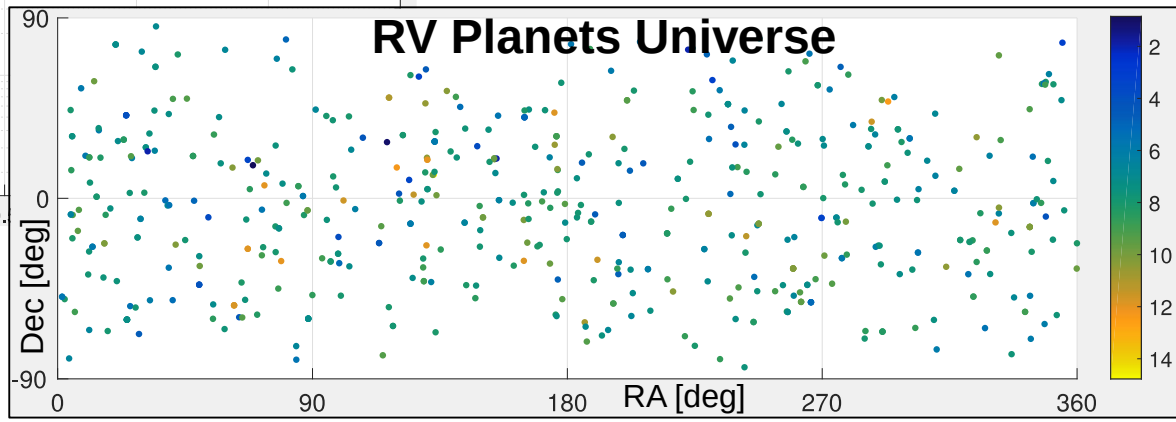
3. End-to-end validation

- Chris Stark's AYO code using standard inputs: ExoCat star list, SAG13 occurrences) for Decadal Mission Concept Studies

Case Study: Photometry

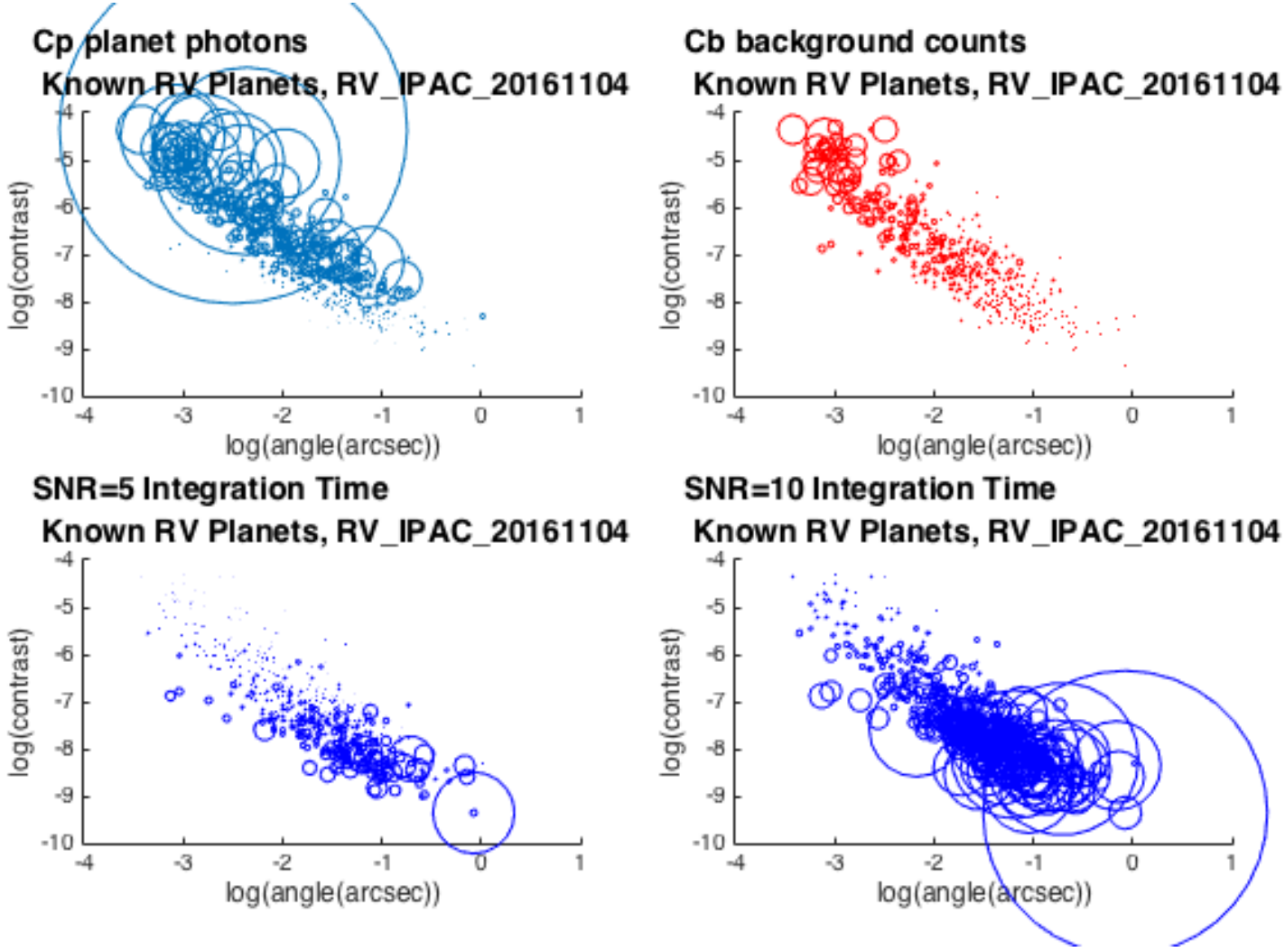
- For photometric integration tests, we used a separate Matlab code with a universe composed of Radial Velocity planets (IPAC catalog)
  - EXOSIMS and Matlab code both work from this catalog: positions, masses, orbits
- Current universe has 580 RV planets of which ~100 can be used in the test
  - Limits include working angle and delta mag
- A range of planet radii and masses are represented in the simulated universe



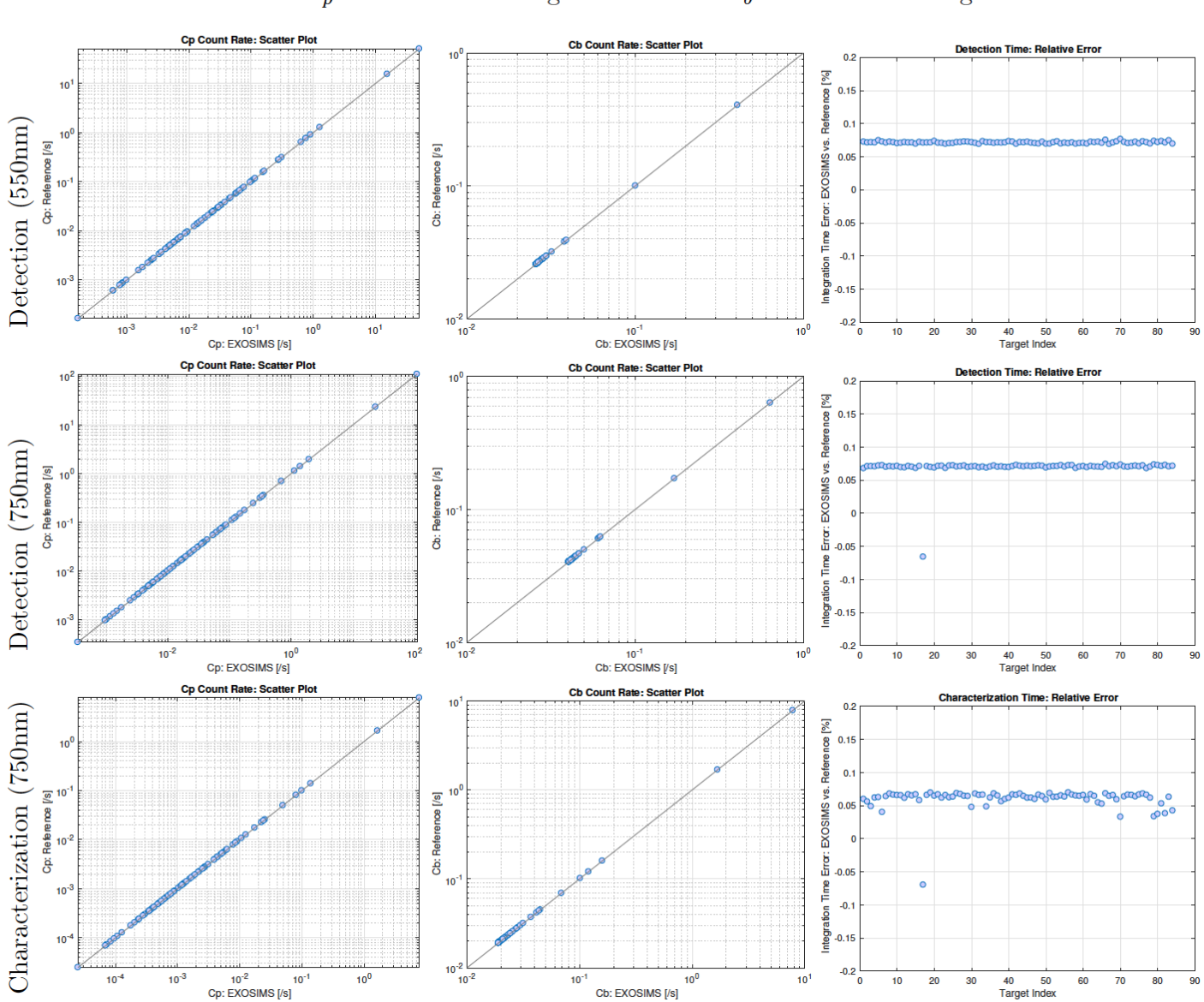


Photometric Results

- Universe of KnownRVPlanets from IPAC portal
- Validated vs. Matlab code: Photon counts (planet/background), integration time



Photometry agreement



- Verified wavelength dependent star flux correction
- Verified bandwidth difference between detection and characterization.

Relative errors are expressed in percent.

Unit Testing Orientation

- The core concept is in the name
  - “Unit” = smallest non-decomposable code segment
  - Typically a unit test applies to a single function
  - Expect several tests per function ⇒ more tests than code
- It’s common in software
  - Not common in science modeling
  - Is used in WFIRST integrated model pipeline and for the conversion of PROPER optical diffraction model
- Python has a unittest framework to provide support
  - Automatically discover tests, run one or all
  - Automatically compute coverage statistics

The Simplest Unit Test


```
def test_calc_Phi(self):
        r"""Test calc_Phi method for computation of exoplanet phase function.

        Approach: Check the computation of Phi at sample points based on
        the Lambert scattering function from, e.g., Traub et al., JATIS 2(1),
        2016, sec. 2.3."""

        ppmodel = self.fixture()
        r = np.linspace(0.0, 2*math.pi, 100) * u.rad

        result = ppmodel.calc_Phi(r)

        phi = lambda(x): (math.sin(x) + (math.pi - x) * math.cos(x))/math.pi
        expected = [phi(x.value) for x in r]

        np.testing.assert_allclose(result, expected, rtol=1e-07)
```

Initialize

EXOSIMS

Check

Compare

Example: Planet Population

Unit tests for Fortney-Marley-Cahoy planet physical model use tabulations of planet properties independently extracted from the three source *ApJ* papers (as corrected):

Fortney, Marley, and Barnes, 2007, *ApJ*, 659, pg. 1670.

Cahoy, K. L., Marley, M.S., & Fortney, 2010, *ApJ*, 724.1, pg. 189.

Fortney, Marley, and Barnes, 2007, *ApJ*, 668, pg. 1267 (erratum).

PLANETARY RADII ACROSS FIVE ORDERS OF MAGNITUDE IN MASS AND STELLAR INSOLATION: APPLICATION TO TRANSITS

J. J. Fortney,[1,2,3] M. S. Marley,[1] and J. W. Barnes[4]

Received 2006 November 3; accepted 2006 December 20

TABLE 4

Giant Planet Radii at 4.5 Gyr

		0.15	0.24	0.41	0.68	1.0	1.46	2.44	4.1	6.8	11.3
(AU)	Mass	17	28	46	77	129	215	318	464	774	1292	2154	3594
0.02	0	...	1.355	1.252	1.183	1.190	1.189	1.179	1.174	1.170	1.178	1.164	1.118
	10	0.726	0.934	1.019	1.072	1.123	1.148	1.153	1.157	1.160	1.172	1.160	1.116
	25	-	0.430	0.756	0.928	1.032	1.091	1.116	1.131	1.145	1.163	1.155	1.112
	50	-	-	-	0.695	0.891	1.004	1.056	1.091	1.121	1.148	1.144	1.106
	100	-	-	-	-	0.613	0.841	0.944	1.011	1.076	1.118	1.125	1.095
0.045	0	1.003	1.065	1.038	1.049	1.086	1.105	1.107	1.108	1.113	1.118	1.099	1.053
	10	0.599	0.775	0.878	0.964	1.029	1.069	1.083	1.092	1.104	1.112	1.095	1.050
	25	-	0.403	0.686	0.846	0.952	1.019	1.050	1.069	1.090	1.104	1.090	1.047
	50	-	-	-	0.648	0.831	0.942	0.996	1.033	1.068	1.090	1.081	1.042
	100	-	-	-	-	0.587	0.798	0.896	0.961	1.026	1.062	1.063	1.032
				1.005	1.024	1.062	1.085	1.090	1.092	1.099	1.104	1.084	1.038
				0.858	0.942	1.008	1.051	1.067	1.077	1.090	1.098	1.080	1.036
				0.675	0.829	0.934	1.002	1.034	1.054	1.077	1.090	1.075	1.033
				-	0.639	0.817	0.928	0.982	1.019	1.055	1.076	1.066	1.027
				-	-	0.582	0.788	0.884	0.949	1.014	1.049	1.049	1.018
				0.983	1.011	1.050	1.074	1.081	1.084	1.091	1.096	1.075	1.030
				0.845	0.931	0.997	1.041	1.058	1.068	1.082	1.090	1.072	1.028
				0.666	0.820	0.924	0.993	1.026	1.046	1.069	1.082	1.067	1.025
				-	0.633	0.810	0.920	0.974	1.011	1.048	1.068	1.058	1.020
				-	-	0.578	0.782	0.878	0.942	1.007	1.041	1.041	1.010
				0.866	0.913	0.957	0.994	1.019	1.037	1.056	1.062	1.055	1.023
				0.759	0.844	0.911	0.966	0.999	1.024	1.048	1.057	1.052	1.021
				0.611	0.750	0.849	0.926	0.972	1.037	1.035	1.050	1.047	1.018
				-	0.594	0.754	0.865	0.926	0.973	1.015	1.037	1.039	1.013
				-	-	0.558	0.746	0.842	0.911	0.976	1.012	1.023	1.004

ers are planet masses in M_J, while row 2 is in $M_\oplus$. The symbol "-" indicates that the planet mass is smaller than 4.3) the approximate T_{eq} value at each distance as 1960 K (0.02 AU), 1300 K (0.045 AU), 875 K (0.1 AU),

Python test values

```
##########################################################################
# Giant Planet Radii at 4.5 Gyr, processed from Table 4 of
# J. J. Fortney, M. S. Marley, and J. W. Barnes,
# The Astrophysical Journal, 659, pg. 1670, 2007.
# Tabular contents pasted into text-file from Acrobat Pro, then reformatted
# by short awk script.
# Each quad-tuple is one entry of the table, in units as stated:
#   (distance[AU], CoreMass[M_earth], PlanetMass[M_earth], Radius[R_jup])
# NaNs exist where CoreMass > PlanetMass, which is impossible.  The first
# NaN is given as "..." in the printed table, apparently because unrealistic.
giant_planet_radii = [
    ( 0.02,   0,  17,np.NaN), ( 0.02,   0,  28, 1.355), ( 0.02,   0,  46, 1.252),
    ( 0.02,   0,  77, 1.183), ( 0.02,   0, 129, 1.190), ( 0.02,   0, 215, 1.189),
    ( 0.02,   0, 318, 1.179), ( 0.02,   0, 464, 1.174), ( 0.02,   0, 774, 1.170),
    ( 0.02,   0,1292, 1.178), ( 0.02,   0,2154, 1.164), ( 0.02,   0,3594, 1.118),
    ( 0.02,  10,  17, 0.726), ( 0.02,  10,  28, 0.934), ( 0.02,  10,  46, 1.019),
    ( 0.02,  10,  77, 1.072), ( 0.02,  10, 129, 1.123), ( 0.02,  10, 215, 1.148)
```

Code and Unit Tests Correspond

EXOSIMS Code

- __init__.py
- BackgroundSources
- Completeness
- MissionSim.py
- Observatory
- OpticalSystem
- PlanetPhysicalModel
- PlanetPopulation
- PostProcessing
- Prototypes
- Scripts
- SimulatedUniverse
- StarCatalog
- SurveyEnsemble
- SurveySimulation
- TargetList
- TimeKeeping
- util
- ZodiacalLight

- __init__.py
- BackgroundSources.py
- Completeness.py
- Observatory.py
- OpticalSystem.py
- PlanetPhysicalModel.py
- PlanetPopulation.py
- PostProcessing.py
- SimulatedUniverse.py
- StarCatalog.py
- SurveyEnsemble.py
- SurveySimulation.py
- TargetList.py
- TimeKeeping.py
- ZodiacalLight.py

1708 code lines

EXOSIMS Unit Tests

- __init__.py
- BackgroundSources
- Completeness
- coverage_entry.py
- htmlcov
- Observatory
- OpticalSystem
- PlanetPhysicalModel
- PlanetPopulation
- PostProcessing
- Prototypes
- setup.sh
- SimulatedUniverse
- StarCatalog
- SurveyEnsemble
- SurveySimulation
- TargetList
- test_MissionSim.py
- TestModules
- TestSupport
- TimeKeeping
- util
- ZodiacalLight

- __init__.py
- test_Observatory.py
- test_OpticalSystem.py
- test_PlanetPhysicalModel.py
- test_PlanetPopulation.py
- test_PostProcessing_others.py
- test_PostProcessing.py
- test_SurveySimulation.py
- test_TargetList.py
- test_TimeKeeping.py
- test_ZodiacalLight.py

1449 code lines

Line counts measured in SLOC

Test Coverage

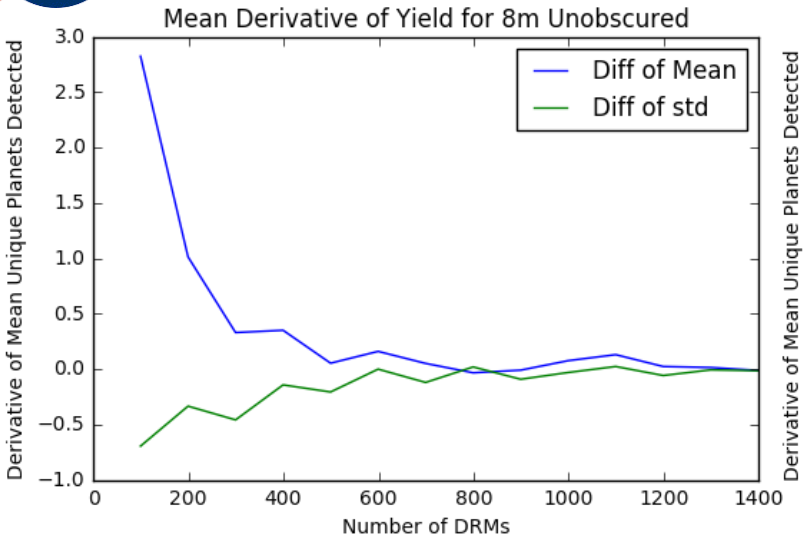
This comes from the standard `coverage` utility built on Python `unittest`

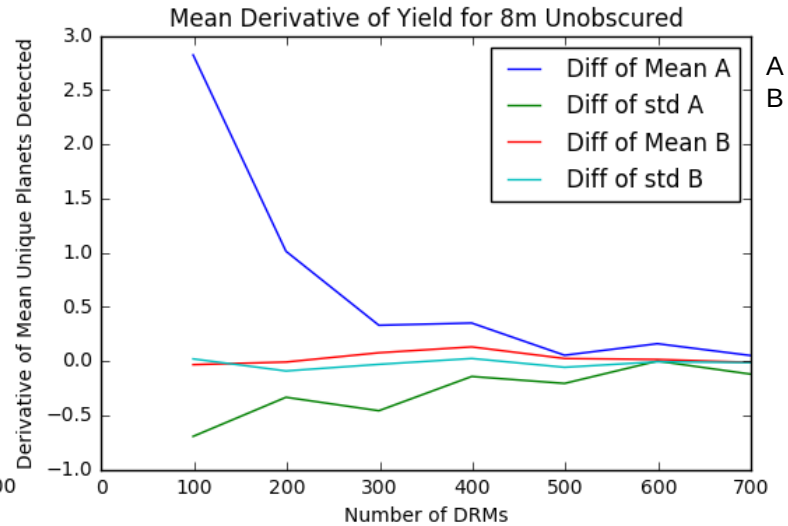
It is an interactive webpage with drill-down into code lines

Coverage report: 79%

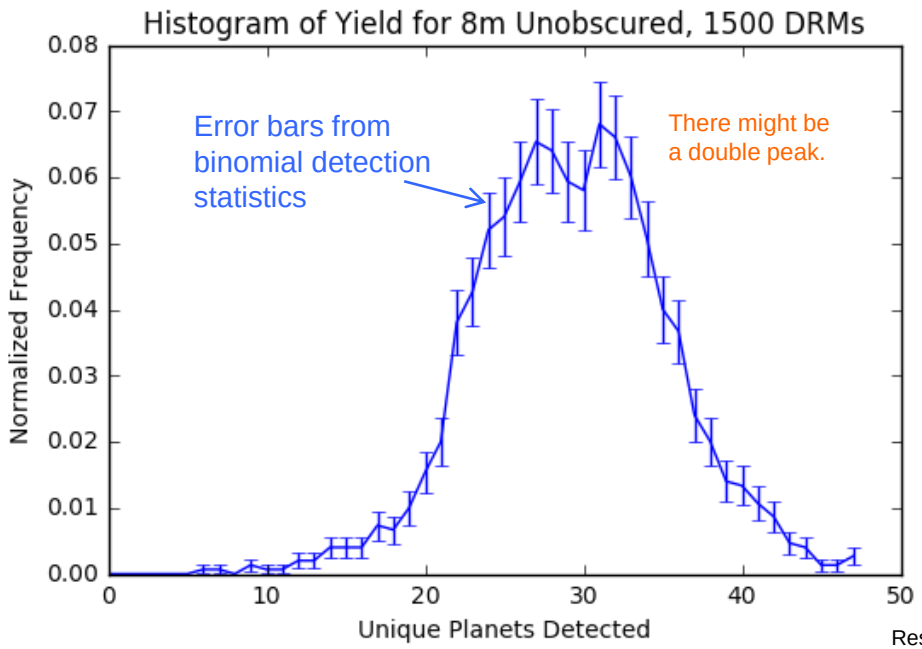
Module	statements	missing	excluded	coverage
EXOSIMS/BackgroundSources/__init__.py	0	0	0	100%
EXOSIMS/Completeness/BrownCompleteness.py	182	71	0	61%
EXOSIMS/Completeness/__init__.py	0	0	0	100%
EXOSIMS/MissionSim.py	124	15	0	88%
EXOSIMS/Observatory/WFIRSTObservatory.py	44	0	0	100%
EXOSIMS/Observatory/__init__.py	0	0	0	100%
EXOSIMS/OpticalSystem/KasdinBraems.py	32	1	0	97%
EXOSIMS/OpticalSystem/Nemati.py	46	36	0	22%
EXOSIMS/OpticalSystem/WFIRSTOpticalSystem.py	26	0	0	100%
EXOSIMS/OpticalSystem/__init__.py	0	0	0	100%
EXOSIMS/PlanetPhysicalModel/FortneyMarleyCahoyMix1.py	92	3	0	97%
EXOSIMS/PlanetPhysicalModel/__init__.py	0	0	0	100%
EXOSIMS/PlanetPopulation/EarthTwinHabZone1.py	18	0	0	100%
EXOSIMS/PlanetPopulation/EarthTwinHabZone2.py	14	0	0	100%
EXOSIMS/PlanetPopulation/KeplerLike1.py	60	0	0	100%
EXOSIMS/PlanetPopulation/KnownRVPlanets.py	80	1	0	99%
EXOSIMS/PlanetPopulation/__init__.py	0	0	0	100%
EXOSIMS/PostProcessing/__init__.py	0	0	0	100%
EXOSIMS/Prototypes/BackgroundSources.py	20	10	0	50%
EXOSIMS/Prototypes/Completeness.py	20	4	0	80%
EXOSIMS/Prototypes/Observatory.py	254	13	0	95%
EXOSIMS/Prototypes/OpticalSystem.py	203	12	0	94%
EXOSIMS/Prototypes/PlanetPhysicalModel.py	26	0	0	100%
EXOSIMS/Prototypes/PlanetPopulation.py	87	3	0	97%
EXOSIMS/Prototypes/PostProcessing.py	32	3	0	91%
EXOSIMS/Prototypes/SimulatedUniverse.py	102	22	0	78%
EXOSIMS/Prototypes/StarCatalog.py	32	3	0	91%
EXOSIMS/Prototypes/StarCatalog.py	32	3	0	91%
EXOSIMS/Prototypes/SurveyEnsemble.py	5	0	0	100%
EXOSIMS/Prototypes/SurveySimulation.py	327	240	0	27%
EXOSIMS/Prototypes/TargetList.py	157	24	0	85%
EXOSIMS/Prototypes/TimeKeeping.py	59	0	0	100%
EXOSIMS/Prototypes/ZodiacalLight.py	42	13	0	69%
EXOSIMS/Prototypes/__init__.py	1	0	0	100%
EXOSIMS/SimulatedUniverse/KeplerLikeUniverse.py	27	0	0	100%
EXOSIMS/SimulatedUniverse/KnownRVPlanetsUniverse.py	51	0	0	100%
EXOSIMS/SimulatedUniverse/__init__.py	0	0	0	100%
EXOSIMS/StarCatalog/EXOCAT1.py	40	1	0	98%
EXOSIMS/StarCatalog/__init__.py	0	0	0	100%
EXOSIMS/SurveyEnsemble/__init__.py	0	0	0	100%
EXOSIMS/SurveySimulation/__init__.py	0	0	0	100%
EXOSIMS/TargetList/KnownRVPlanetsTargetList.py	45	0	0	100%
EXOSIMS/TargetList/__init__.py	0	0	0	100%
EXOSIMS/TimeKeeping/__init__.py	0	0	0	100%
EXOSIMS/ZodiacalLight/Stark.py	46	37	0	20%
EXOSIMS/ZodiacalLight/__init__.py	0	0	0	100%
EXOSIMS/__init__.py	2	0	0	100%
EXOSIMS/util/__init__.py	0	0	0	100%
EXOSIMS/util/deltaMag.py	4	0	0	100%
EXOSIMS/util/eccanom.py	24	1	0	96%
EXOSIMS/util/get_module.py	92	9	0	90%
EXOSIMS/util/keplerSTM.py	84	4	0	95%
EXOSIMS/util/statsFun.py	38	8	0	79%
Total	**2538**	**534**	**0**	**79%**

How many DRMs should be in an ensemble?





A = DRMs 1:700
B = DRMs 800:1500

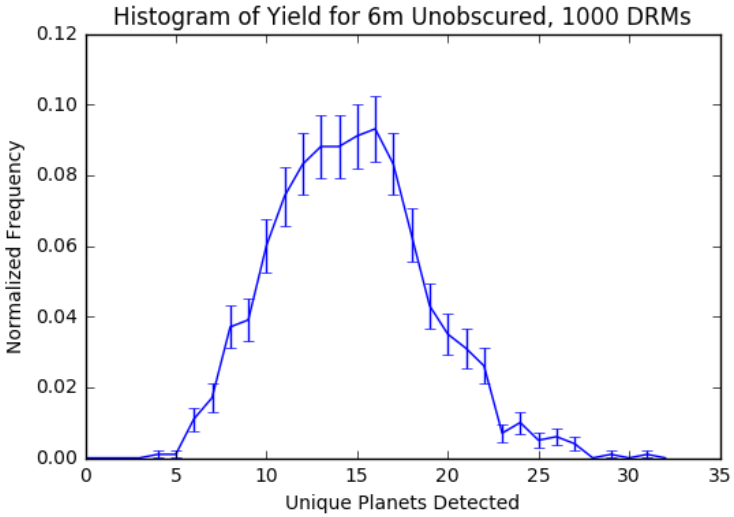


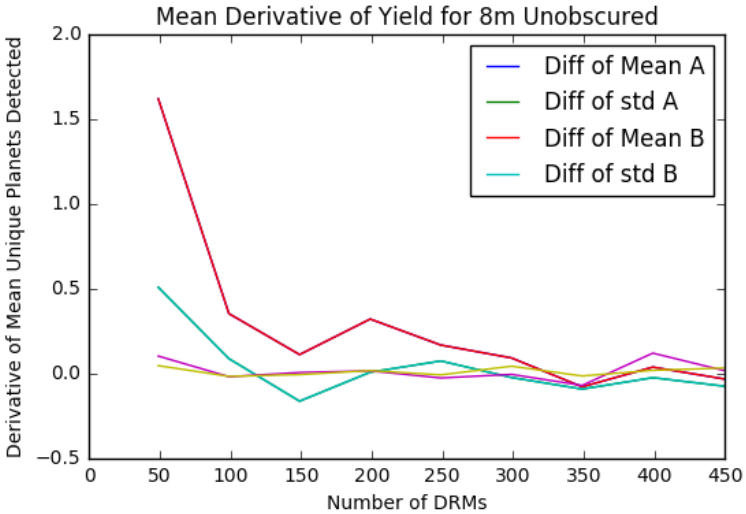
8m diameter unobscured
missionLife = 1
missionPortion = 1.0
Num DRMs = 1500
eta = 0.1

For an 8m Diameter and eta = 0.1, I would use 800 DRMs

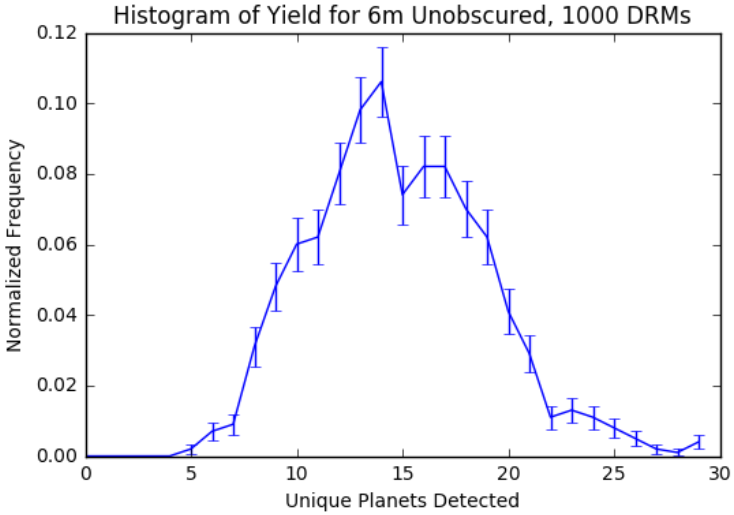
Results from numDRMs_8m_1500.ipnb using output HabExAYO_8m_20170127_cluster/20170301_2045

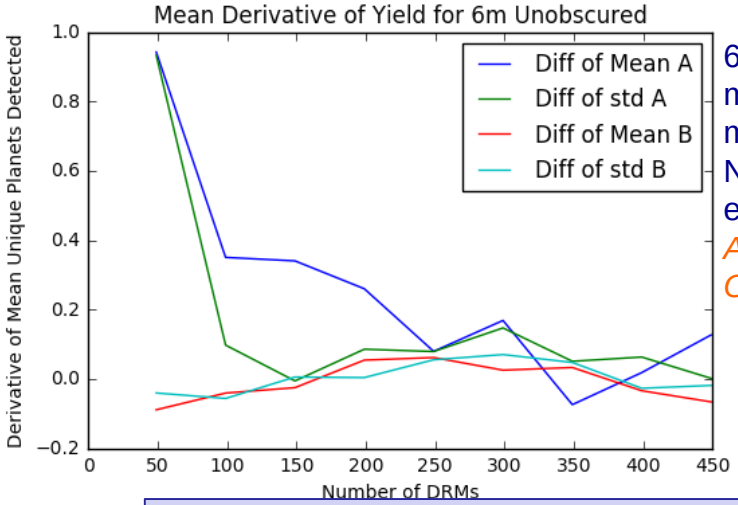
How many DRMs … 6m





6m diameter unobscured
missionLife = 1
missionPortion = 1.0
Num DRMs = 1000
eta = 0.1



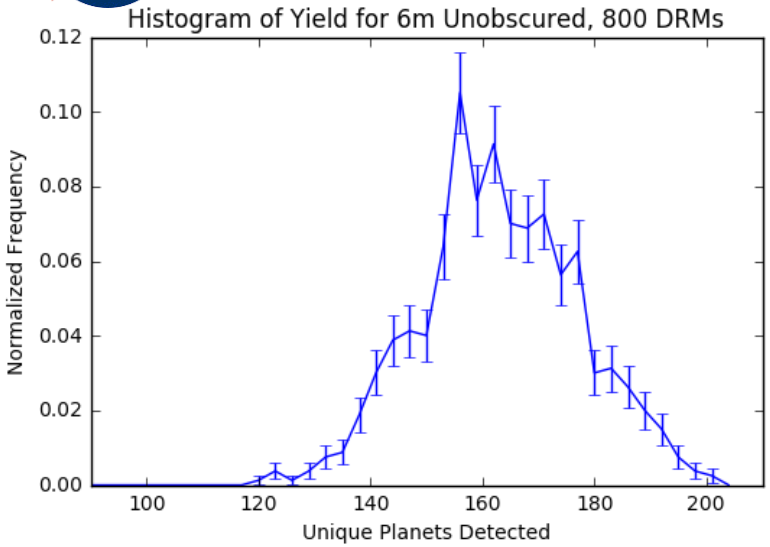


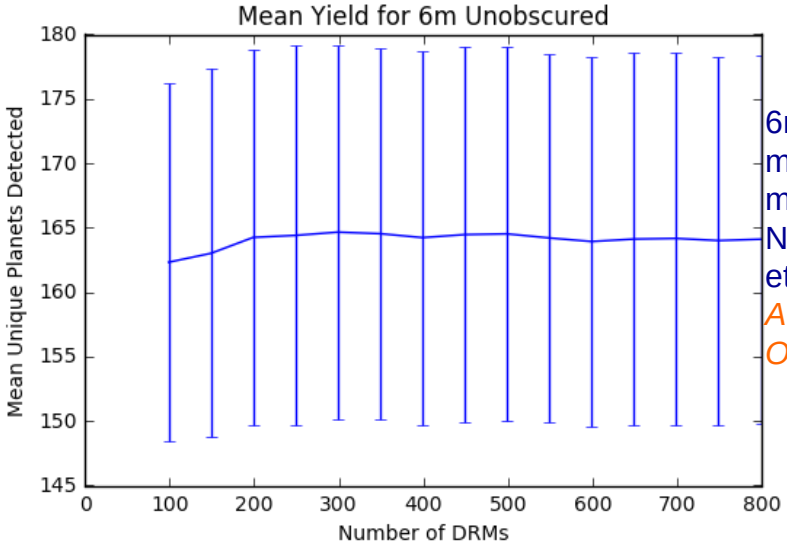
6m diameter unobscured
missionLife = 1
missionPortion = 1.0
Num DRMs = 1000
eta = 0.1
After github update to fix Observatory

For an 6m Diameter and eta = 0.1, I would use 600 DRMs

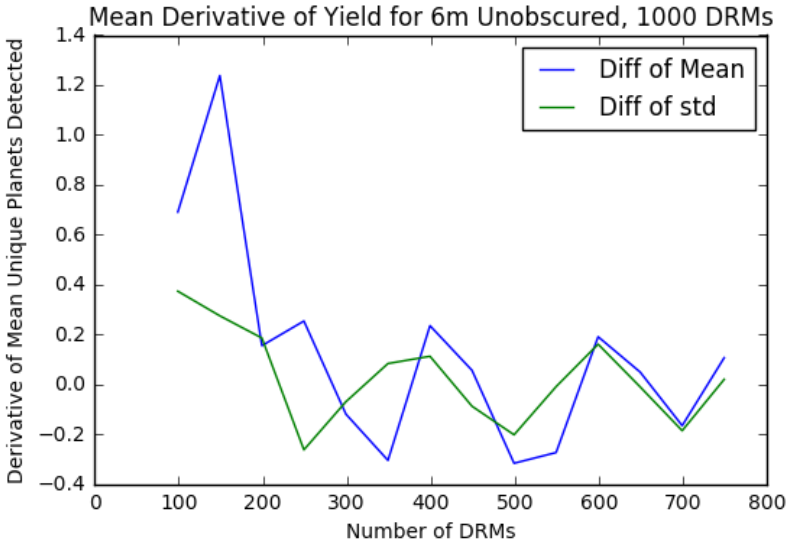
Top: Results from numDRMs_6m_1000.ipnb using output HabExAYO_6m_20170215_cluster/ 20170221_2327
Bottom: Results from numDRMs_6m_1000numDRMs_6m_1000_20170301_0029.ipnb using output HabExAYO_6m_20170215_cluster/20170301_0029

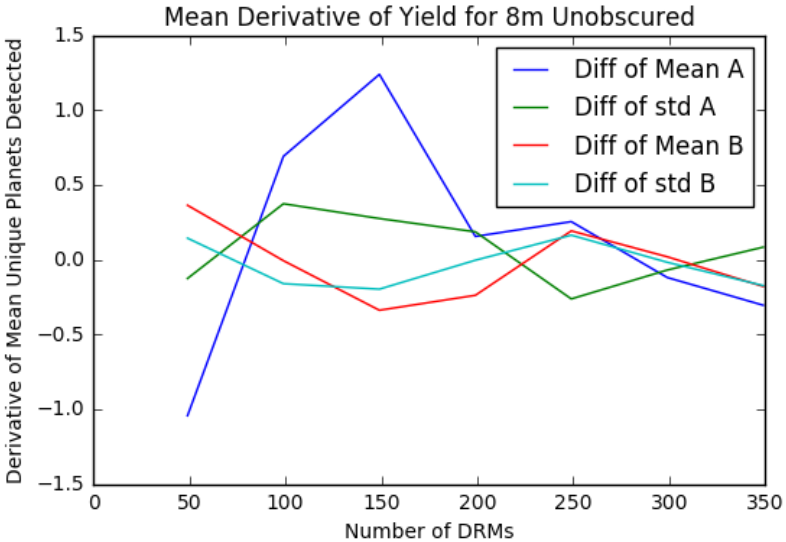
How many DRMs … eta = 1





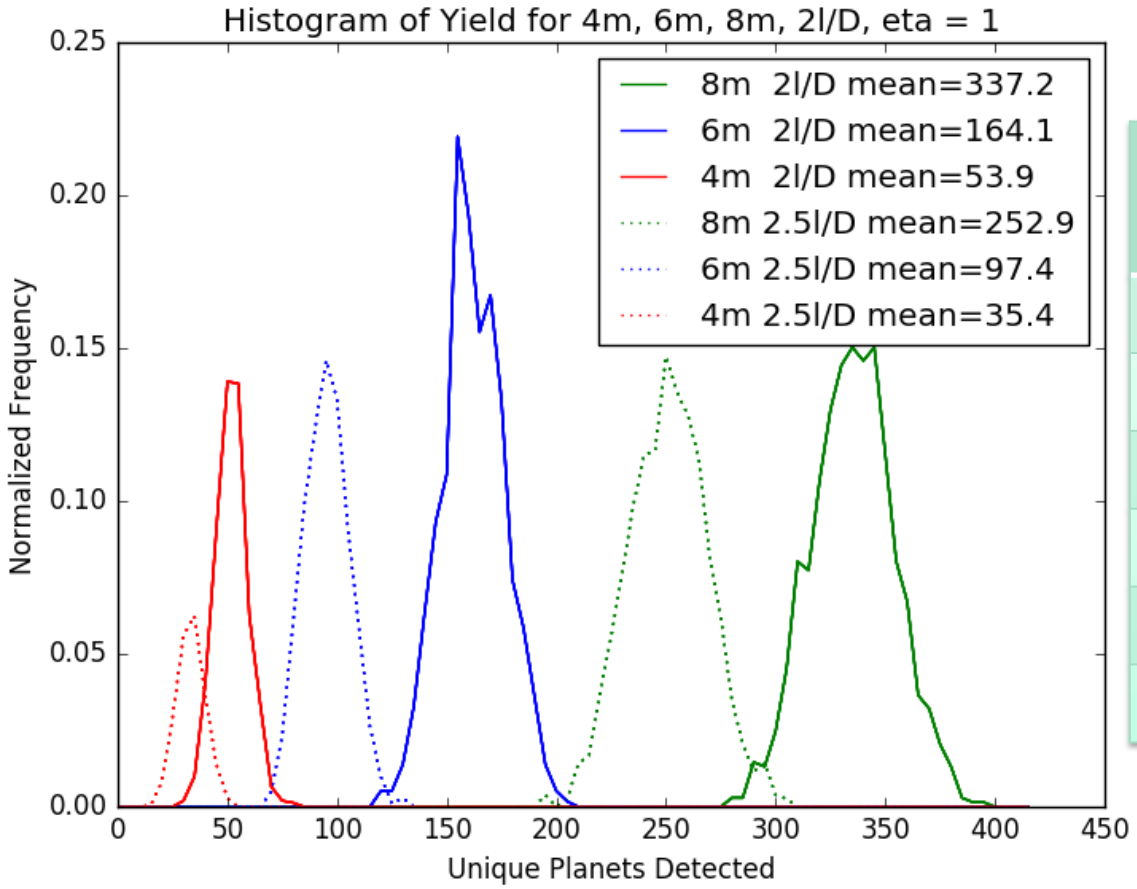
6m diameter unobscured
missionLife = 1
missionPortion = 1.0
Num DRMs = 800
eta = 1.0
After github update to fix Observatory





For an 6m Diameter and eta = 1.0, I would use 400 DRMs

EXOSIMS and AYO, very different approaches, produce similar yields



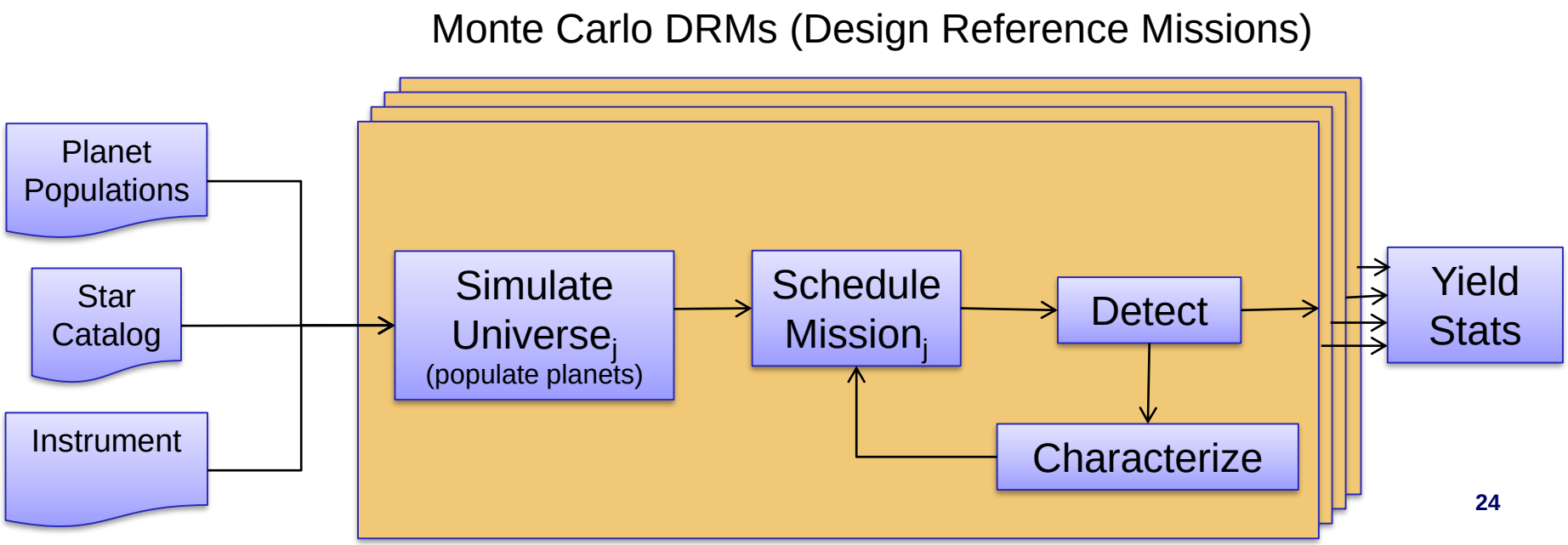
	AYO	EXOSIMS mean	EXOSIMS std
4m, 2l/D	6.6	5.4	1.5
4m, 2.5 l/D	5.6	3.5	0.7
6m, 2 l/D	14.0	16.4	1.4
6m, 2.5 l/D	12.2	9.7	1.0
8m, 2 l/D	23.2	33.7	1.9
8m, 2.5 l/D	20.3	25.3	1.8

Possible sources of variance:

- Overhead accounting
- Star selection
- Exozodi throughput

BACKUP

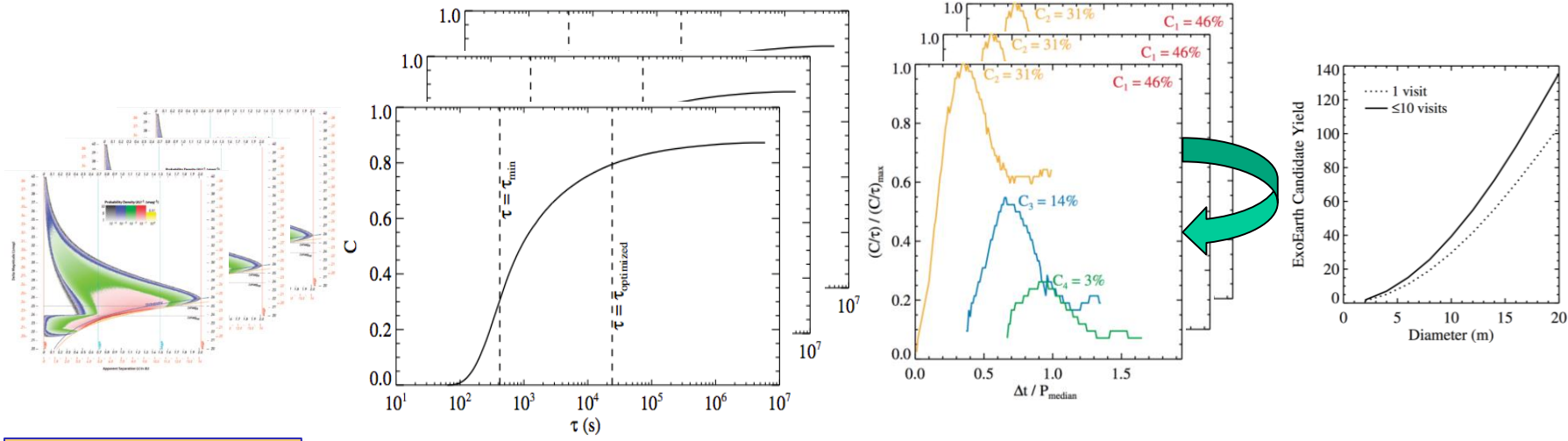
Simulated Mission Ensemble



- Simulate a universe by sampling planet population distributions
- Schedule a mission using spacecraft and mission observing constraints
  - Dynamically respond to detections
- Repeat with a new universe 1,000 times

Chris Stark's AYO Approach

Completeness(T_{int})



Monte Carlo Planets around stars → Instrument → Optimize Int. Time Alloc. → Cumulative Completeness

Planet Classes

Star list

AYO: Altruistic Yield Optimization

Produces Static Time Budget

Typical Unit Test Strategies

- Numerical checks with another implementation
  - Solar system ephemeris: checked vs. an independent implementation using JPL Spice kernel
  - Stellar proper motion is point-checked with an independent calculation
  - Random number distribution checks, in some planet population cases
- Numerical checks versus tabulated data
  - Fortney-Marley-Cahoy planet physical model checks against full tabulations of planet properties from three *ApJ* papers (2007, 2010)
  - Known RV Planets universe checks exhaustively against IPAC catalog of stars and planets (584 planets, about 35 attributes)
  - Zodiacal light (Stark) checks vs. an independent tabulation by P. Lowrance covering ~40 sources at two wavelengths
- Functionality checks
  - Object constructors must set all properties (as many as 30)
  - Verify object type, units, size, range matches constraints

Testing Has Improved EXOSIMS

- Finding bugs
  - Solar system ephemeris
  - Keepout
  - L2 halo orbit
  - IWA for constant contrast
- Revealing issues (problems that are not bugs)
  - Cached completeness result could lead to incorrect results
  - More detailed reporting of simulation results
- Changing method of implementation
  - Solar system ephemeris now uses JPL Spice kernels rather than Kepler elements
- Improving repeatability
  - Unified random number generation allows end-to-end repeatability
  - DeterministicUniverse class
  - Logging