EXOSIMS Instantiantiaion Input Specs: Json specfile SurveyEnsemble **TimeKeeping** Observatory MissionSimulation import EXOSIMS.MissionSim as msim Instantiates Instantiates sim = msim.MissionSim(scriptfile) Instantiates SurveySimulation Instantiates Instantiate **S**imulated**U**niverse **TargetList** Instantiates - Instantiates Instantiates Instantiates Instantiates Instantiates BrownCompleteness Completeness as Completeness **StarCatalog** module? BackgroundSource **PostProcessing** Instantiate **OpticalSystem PlanetPopulation** EarthTwinHabZone 1 &2 **Brown Completeness** EXOCAT1 KasdinBraems GalaxiesFaintStars Instantiate Instantiate KeplerLike1 SIMBAD3000Catalog Nemati PlanetPhysicalModel ZodiacalLight Completeness array KnownRVPlanets exists in .comp file? SIMBADCatalog WFIRSTOpticalSystem FortneyMarleyCahoyMix1 Stark Start Monte Carlo Load previous MC with Simulation completeness array. Number of iterations = Nplanets/nplan: Nplanets = Input of number of planets to generate Simulation done on 2D grid nplan = min(1e6, Nplanets) of planet separation vs planet delta mag Filter target list based on: • Binaries Populate TargetList. • IWA, OWA • Find max int time for targets Takes vals from StarCat into • delta mag · obtain stellar parameters empirically TargetList attributes in catalog_atts • int time • completeness • Generate Planetary systems • Randomly selected stars are selected to host planets based Make all objects in Set inherited classes to top level on probability of number of **specs top level (e.g., self.X = TargetList.X) systems with planets. • generate orbital parameters • sys with planets = sInds